BSc Thesis

Valdemar H. Lorenzen

*Melih Kandemir

IMADA, SDU

Abstract

TODO

Write the abstract

1 Introduction

Many real-world applications present an inherent challenge that current reinforcement learning (RL) methods struggle to address effectively: the problem of delayed and sparse rewards [1], [2].

> **Delayed and Sparse Rewards:** Learning scenarios where meaningful feedback signals (rewards) are provided only far after a long sequence of actions, and where most actions yield no immediate feedback.
>
> *Example: In drug discovery, the effectiveness of a designed molecule can only be evaluated after its complete synthesis, with no intermediate feedback during the design process.*

Consider, for instance, the process of drug design, where a reinforcement learning agent must make a series of molecular modifications to create an effective compound. The value of these decisions — the drug's efficacy — can only be assessed once the entire molecule is complete. Similarly, in robotics tasks like assembly or navigation, success often depends on precise sequences of actions where feedback is only available upon having completed the entire task.

Traditional reinforcement learning algorithms face two critical limitations in such environments:

1. **Credit Assignment:** When rewards are delayed, the algorithm struggles to correctly attribute success or failure to specific actions in a long sequence [3]. This is analogous to trying to improve a chess strategy when only knowing the game's outcome, without understanding which moves were actually decisive.
2. **Exploration Efficiency:** With sparse rewards, random exploration becomes highly inefficient [4], [5]. An agent might need to execute precisely the right sequence of actions to receive any feedback at all, making random exploration about as effective as searching for a needle in a haystack.

This thesis investigates a novel approach to addressing these challenges through the comparison of two promising methodologies: **Generative Flow Networks** (GFlowNets) as proposed by [6], and **Bayesian Exploration Networks** (BEN) as proposed by [7]. These approaches represent different perspectives on handling uncertainty and exploration in reinforcement learning.

1. *GFlowNets* frame the learning process as a flow network, potentially offering more robust learning in situations with multiple viable solutions.
2. *BENs* leverages Bayesian uncertainty estimation to guide exploration more efficiently, potentially making better use of limited feedback.

By comparing these approaches, we aim to understand their relative strengths and limitations in environments with delayed and sparse rewards. Our investigation focuses specifically on examining these methods in carefully designed environments that capture the essential characteristics of delayed and sparse reward scenarios while remaining tractable for systematic analysis.

1.1 Research Objectives and Contributions

This thesis aims to advance our understanding of efficient learning in sparse reward environments through three primary objectives:

1. **Comparative Analysis:** Conduct a rigorous empirical comparison between GFlowNets and Bayesian Exploration Networks in standardized environments with delayed rewards.
2. **Hypothesis Testing:** Investigate whether BEN's Bayesian exploration strategy leads to more efficient learning compared to GFlowNets in highly delayed reward scenarios, particularly during early training stages.
3. **Algorithmic Understanding:** Analyze the underlying mechanisms that drive performance differences between these approaches, focusing on their handling of uncertainty and exploration.

The contributions of this work include:

- A comprehensive empirical evaluation using the n-chain environment with varying degrees of reward delay.
- Insights into the relative strengths and limitations of Bayesian and flow-based approaches to exploration.
- Implementation and analysis of both algorithms with comparisons.

1.2 Thesis and Structure

The remainder of this thesis is structured as follows:

Section 2: Preliminaries provides the theoretical foundations of reinforcement learning and explores existing approaches to handling sparse rewards. This chapter establishes the mathematical framework and notation used throughout the thesis.

Section 3: Theoretical Framework presents our hypothesis and analytical approach. We develop the mathematical foundations for comparing GFlowNets and BEN.

Section 4: Experimental Design details our testing methodology, including environment specifications, evaluation metrics, and implementation details.

Section 5: Results and Analysis presents our findings, including both quantitative performance metrics and qualitative analysis of learning behaviors. We examine how each algorithm handles the exploration-exploitation trade-off and adapts to varying levels of reward sparsity.

Section 6: Conclusion summarizes our findings, discusses their implications for the field, and suggests directions for future research.

2 Preliminaries

2.1 Flow Networks

GFlowNets rely on the concept of flow networks. The flow network is represented as a directed acyclic graph $G = (\mathcal{S}, \mathcal{A})$, where $\mathcal{S}$ represents the state space and $\mathcal{A}$ represents the action space.

> **Flow Network:** A directed acyclic graph with a single source node (initial state) and one or more sink nodes (terminal states), where flow is conserved at each intermediate node [6], [8].
>
> *Example: In molecular design, states represent partial molecules and actions represent adding molecular fragments.*

2.1.1 States and Trajectories

We distinguish severel types of states:

- An initial state $s_0 \in \mathcal{S}$ (the source);
- Terminal states $x \in \mathcal{X} \subset \mathcal{S}$ (sinks);
- Intermediate states that form the pathways from source to sinks.

A trajectory τ represents a complete path through the network, starting at s_0 and ending at some terminal state x. Formally, we write a trajectory as an ordered sequence $\tau = (s_0 \to s_1 \to \ldots \to s_n = x)$, where each transition $(s_t \to s_{t+1})$ corresponds to an action in $\mathcal{A}$.

2.1.2 Flow Function and Conservation

The *trajectory flow function* $F : \mathcal{T} \to \mathbb{R}_{\geq 0}$ assigns a non-negative value to each possible trajectory [8]. From this flow function, two important quantities are derived:

1. **State flow**: For any state s, its flow is the sum of flows through all trajectories passing through it:

$$F(s) = \sum_{s \in \tau} F(\tau). \tag{1}$$

2. **Edge flow**: For any action (edge) $s \to s'$, its flow is the sum of flows through all trajectories using that edge:

$$F(s \to s') = \sum_{\tau = (\ldots \to s \to s' \to \ldots)} F(\tau). \tag{2}$$

These flows must satisfy a conservation principle known as the *flow matching constraint*:

> **Flow Matching:** For any non-terminal state s, the total incoming flow must equal the total outgoing flow:
>
> $$F(s) = \sum_{(s'' \to s) \in \mathcal{A}} F(s'' \to s) = \sum_{(s \to s') \in \mathcal{A}} F(s \to s'). \tag{3}$$

2.1.3 Markovian Flow

The flow function induces a probability distribution over trajectories. Given a flow function F, we define $P(\tau) = \frac{1}{Z}F(\tau)$ [8], where $Z = F(s_0) = \sum_{\tau \in \mathcal{T}} F(\tau)$ is the *partition function* — i.e., the total flow through the network.

> **Markovian Flow:** A flow is *Markovian* when it can be factored into local decisions at each state. This occurs when the following criteria are met [8]:
>
> 1. Forward policies $P_F(-|s)$ over children of each non-terminal state s.t.
>
> $$P(\tau = (s_0 \to \ldots \to s_n)) = \prod_{t=1}^{n} P_F(s_t|s_{t-1}). \tag{4}$$
>
> 2. Backward policies $P_B(-|s)$ over parents of each non-initial state s.t.
>
> $$P(\tau = (s_0 \to \ldots \to s_n)|s_n = x) = \prod_{t=1}^{n} P_B(s_{t-1}|s_t). \tag{5}$$

The Markovian property allows us to decompose complex trajectory distributions into simple local decisions, making learning tractable while maintaining the global flow constraints.

2.2 GFlowNets

GFlowNets are an approach to learning policies that sample from desired probability distributions [6]. They frame the learning process as discovering a flow function that makes the probability of generating any particular object proportional to its reward.

Given a reward function $R : \mathcal{X} \to \mathbb{R}_{\geq 0}$ defined over the set of terminal states $\mathcal{X}$, GFlowNets aim to approximate a Markovian flow F on the graph G s.t. $F(x) = R(x)$ for all $x \in \mathcal{X}$. We will make use of the following definition of a GFlowNet.

> **GFlowNet:** [8] defines a GFlowNet as any learning algorithm that discovers flow functions matching terminal state rewards, consisting of:
>
> 1. A model that outputs:
>   - Initial state flow $Z = F(s_0)$;
>   - Forward action distributions $P_F(-|s)$ for non-terminal states.
> 2. An objective function that, when globally minimized, guarantees $F(x) = R(x)$ for all terminal states.
>
> *Example: In molecular design, this ensures that high-reward molecules are genered more frequently, while maintaining diversity through exploration of multiple pathways.*

The power of GFlowNets lies in their ability to handle situations where multiple action sequences can lead to the same terminal state — a common scenario in real-world applications like molecular design or image synthesis. Unlike traditional RL methods that focus on finding a single optimal path, GFlowNets learn a distribution over all possible paths directly proportional to their rewards.

2.2.1 Learning Process

The learning process of GFlowNets involves iteratively improving both flow estimates and the policies. The forward policy of a GFlowNet can sample trajectories from the learned Markovian flow F by sequentially selecting actions according to $P_F(-|s)$. When the training converges to a global minimum of the objective function, this sampling process guarantees that $P(x) \propto R(x)$. That is, the probability of generating any terminal state x is proportional to its reward $R(x)$. This property makes GFlowNets particularly well-suited for:

1. **Diverse Candidate Generation:** Rather than converging to a single solution, GFlowNets maintain a distribution over solutions weighted by their rewards.
2. **Multi-Modal Exploration:** The flow-based approach naturally handles problems with multiple distinct solutions of similar quality.
3. **Compositional-Structure Learning:** By learning flows over sequences of actions, GFlowNets can capture and generalize compositional patterns in the solution space.

To achieve this, GFlowNets employ various training objectives, with *trajectory balance* [8] being one such particularly effective objective.

2.2.2 Trajectory Balance

Trajectory balance focuses on ensuring consistency across entire trajectories, instead of matching flows at every state (which can be computationally expensive).

> **Trajectory Balance:** A principle that ensures the probability of generating a trajectory matches its reward by maintaining consistency between forward generation and backward reconstruction probabilities.

Consider a Markovian flow F that induces a distribution P over trajectories according to $P(\tau) = \frac{1}{Z} F(\tau)$. The forward policy P_F and backward policy P_B must satisfy the following *trajectory balance constraint* [8]

$$Z \prod_{t=1}^{n} P_F(s_t|s_{t-1}) = F(x) \prod_{t=1}^{n} P_B(s_{t-1}|s_t). \tag{6}$$

That is to say, the probability of constructing a trajectory forward should match the probability of reconstructing it backward, scaled by the appropriate rewards.

2.2.3 Trajectory Balance as an Objective

To convert the trajectory balance function into a training objective, we introduce a parametrized model with parameters θ that outputs:

1. A forward policy $P_F(-|s; \theta)$;

2. A backward policy $P_B(-|s;\theta)$;
3. A scalar estimate Z_θ of the partition function.

For any complete trajectory $\tau = (s_0 \to \ldots \to s_n = x)$, we define the *trajectory balance loss* as

$$\mathcal{L}_{\text{TB}}(\tau) = \left(\log \frac{Z_\theta \prod_{t=1}^n P_F(s_t|s_{t-1};\theta)}{R(x) \prod_{t=1}^n P_B(s_{t-1}|s_t;\theta)}\right)^2. \tag{7}$$

This loss captures how well our model satisfies the trajectory balance constraint. When the loss approaches zero, our model has learned to generate samples proportional to their rewards. In practice, we compute this loss in the log domain to avoid numerical stability, as suggested by [8]:

$$\mathcal{L}_{\text{TB}}(\tau) = \left(\log Z_\theta + \log \sum_{t=1}^n P_F(s_t|s_{t-1};\theta) - \log R(x) - \log \sum_{t=1}^n P_B(s_{t-1}|s_t;\theta)\right)^2. \tag{8}$$

[8] also remarks that a simplificatoin of Equation 7 occurs in tree-structured state spaces (when G is a directed tree), where each state has exactly one parent. In such cases, the backward policy becomes deterministic ($P_B = 1$), reducing the loss function to

$$\mathcal{L}_{\text{TB}}(\tau) = \left(\log \frac{Z_\theta \prod_{t=1}^n P_F(s_t|s_{t-1};\theta)}{R(x)}\right)^2, \tag{9}$$

which can be exploited for the n-chain environment.

The model is trained by sampling trajectories from a training policy π_θ — typically a tempered version of $P_F(-|-;\theta)$ to encourage exploration — and updating parameters using stochastic gradient descent: $\theta \leftarrow \theta - \alpha \mathbb{E}_{\tau \sim \pi_\theta} \nabla_\theta \mathcal{L}_{\text{TB}}(\tau)$.

2.3 Markov Decision Processes

The concept of the Markov Decision Process (MDP) [9] is fundamental in reinforcement learning and provides a model for sequential decision-making under uncertainty.

> **Markov Decision Process [9]:** A tuple $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, P_0, P_S, P_R, \gamma \rangle$ where:
>
> - $\mathcal{S}$ is the set of states;
> - $\mathcal{A}$ is the set of actions;
> - $P_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution;
> - $P_S : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the state transition distribution;
> - $P_R : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathbb{R})$ is the reward distribution;
> - $\gamma \in [0, 1]$ is the discount factor.

At each timestep t, an agent observes its current state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$ according to some policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$. The environment then transitions to a new state s_{t+1} according to the transition distribution $P_S(s_t, a_t)$ and provides a reward r_t sampled from $P_R(s_t, a_t)$. The agent's objective is to find a policy π that maximizes the expected sum of discounted future rewards

$$J^\pi := \mathbb{E}_{\tau \sim P^\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \tag{10}$$

where $\tau = (s_0, a_0, r_0, s_1, ...)$ represents a trajectory through the environment and P^π is the distribution over trajectories induced by following policy π. An optimal policy $\pi^* \in \Pi^* := \arg\max_\pi J^\pi$ can be found through the optimal value function $V^* : \mathcal{S} \to \mathbb{R}$ or the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which satisfy the Bellman optimality equations [9], [10]

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a), \tag{11}$$

$$Q^*(s, a) = \mathbb{E}_{r, s' \sim P_{R,S}(s,a)}\left[r + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a')\right]. \tag{12}$$

This framework serves as the building block for more sophisticated models like Contextual MDPs and Bayesian approaches to reinforcement learning, as described in the following sections.

2.4 Contextual Reinforcement Learning

In contextual RL, we use the concept of a Contextual MDP.

> **Contextual MDP:** A Markov Decision Process augmented with a context variable that determines the specific dynamics of the environment [11]. This allows us to model uncertainty about the true environment through uncertainty about the context.

In a Contextual Markov Decision Process (CMDP), we work in an infinite-horizon, discounted setting where a context variable $\varphi \in \Phi \subseteq \mathbb{R}^d$ indexes specific MDPs. Formally, we describe this as

$$\mathcal{M}(\varphi) := \langle \mathcal{S}, \mathcal{A}, P_0, P_S(s, a, \varphi), P_R(s, a, \varphi), \gamma \rangle. \tag{13}$$

where the context φ parametrizes both:

- A transition distribution $P_S(s, a, \varphi)$ determining how states evolve;
- A reward distribution $P_R(s, a, \varphi)$ determining the rewards received.

The agent has complete knowledge of the following aspects of the environment:

- The state space $\mathcal{S} \subset \mathbb{R}^n$;
- The action space $\mathcal{A}$;
- The initial state distribution P_0;
- The discount factor γ.

However, the agent does not know the true context φ^* that determines the actual dynamics and rewards.

2.4.1 Policies and Histories

In contextual RL, an agent follows a *context-conditioned policy* $\pi : \mathcal{S} \times \Phi \to \mathcal{P}(\mathcal{A})$, selecting actions according to $a_t \sim \pi(s_t, \varphi)$. As the agent interacts with the environment, it accumulates a history of experiences $h_t := \{s_0, a_0, r_0, s_1, a_1, r_1, ..., a_{t-1}, r_{t-1}, s_t\}$. This history belongs to a state-action-reward product space $\mathcal{H}_t$ and follows a context-conditioned distribution $P_t^\pi(\varphi)$ with density

$$p_t^\pi(h_t|\varphi) = p_0(s_0) \prod_{i=0}^{t} \pi(a_i|s_i, \varphi) p(r_i, s_{i+1}|s_i, a_i, \varphi). \tag{14}$$

2.4.2 Optimization Objective

The agent's goal in a CMDP is to find a policy that optimizes the expected discounted return

$$J^\pi(\varphi) = \mathbb{E}_{\tau_\infty \sim P_\infty^\pi(\varphi)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{15}$$

An optimal policy $\pi^*(\cdot, \varphi)$ belongs to the set $\Pi_\Phi^*(\varphi) := \arg\max_{\pi \in \Pi_\Phi} J^\pi(\varphi)$. With this, we define the optimal Q-function $Q^*(h_t, a_t, \varphi)$.

> **Optimal Q-Function:** For an optimal policy π^*, *the optimal Q-function* $Q^* : \mathcal{S} \times \mathcal{A} \times \Phi \to \mathbb{R}$ satisfies the Bellman equation
>
> $$\mathcal{B}^*[Q^*](s_t, a_t, \varphi) = Q^*(s_t, a_t, \varphi), \tag{16}$$
>
> where $\mathcal{B}^*$ is the optimal Bellman operator defined as
>
> $$\mathcal{B}^*[Q^*](s_t, a_t, \varphi) := \mathbb{E}_{r_t, s_{t+1} \sim P_{R,S}(s_t, a_t, \varphi)} \left[r_t + \max_{a' \in \mathcal{A}} Q^*(s_{t+1}, a', \varphi) \right]. \tag{17}$$

2.4.3 The Learning Challenge

When an agent has access to the true MDP $\mathcal{M}(\varphi^*)$, finding an optimal policy becomes a *planning problem*. However, in real-world scenarios, agents typically lack access to the true transition dynamics and reward functions. This transforms the task into a *learning problem*, where the agent must balance:

1. *Exploration*: learning about the environment's dynamics through interaction;
2. *Exploitation*: using current knowledge to maximize rewards.

This tension — known as the exploration/exploitation dilemma — remains one of the core challenges in reinforcement learning. As we'll see in the next section, Bayesian approaches offer a principled framework for addressing this challenge.

2.5 Bayesian Reinforcement Learning

In the Bayesian approach to RL, rather than viewing uncertainty as a problem to be eliminated, it becomes an integral part of the decision-making process — something to be reasoned about systematically [12].

> **Bayesian Epistemology:** A framework that characterizes uncertainty through probability distributions over possible worlds. In reinforcement learning, this means maintaining distributions over possible MDPs, updated as new evidence arrives [12].

2.5.1 From Prior to Posterior

The Bayesian learning process starts with a *prior distribution* P_Φ representing our initial beliefs about the true context φ^* before any observations. As the agent interacts with the environment, it accumulates a history of experiences h_t and updates these beliefs through Bayesian inference, forming a *posterior distribution* $P_\Phi(h_t)$.

This history-dependent posterior in Bayesian RL differentiates it from traditional RL approaches.

> **History-Conditioned Policies:** Unlike traditional RL policies that map states to actions, Bayesian policies operate on entire histories, defining a set of history-conditioned policies $\Pi_{\mathcal{H}} := \{\pi : \mathcal{H} \to \mathcal{P}(\mathcal{A})\}$, where $\mathcal{H} := \{\mathcal{H}_t | t \geq 0\}$ denotes the set of all histories [12].

Where the prior P_Φ represents our initial uncertainty (the special case where $h_t = \emptyset$), the posterior $P_\Phi(h_t)$ captures our refined beliefs after observing interactions with the environment. This allows us to reason about future outcomes by marginalizing across all possible MDPs according to our current uncertainty.

2.5.2 The Bayesian Perspective on Transitions

The power of the Bayesian approach stems from how it handles state transitions. Instead of committing to a single model of the environment, it maintains a distribution over possible transitions through the *Bayesian state-reward transition distribution*

$$P_{R,S}(h_t, a_t) := \mathbb{E}_{\varphi \sim P_\Phi(h_t)}\left[P_{R,S}(s_t, a_t, \varphi)\right]. \tag{18}$$

This distribution lets us reason about future trajectories using the *prior predictive distribution* P_t^π with density

$$p_t^\pi(h_t) = p_0(s_0) \prod_{i=0}^{t} \pi(a_i | h_i) p(r_i, s_{i+1} | h_i, a_i). \tag{19}$$

The belief transition distribution $P_{\mathcal{H}}(h_t, a_t)$ captures how our beliefs evolve with new observations, with density

$$p_{\mathcal{H}}(h_{t+1} | h_t, a_t) = p(s_{t+1}, r_t | h_t, a_t). \tag{20}$$

This formulation leads to the definition of the Bayes-adaptive MDP (BAMDP) [13]

$$\mathcal{M}_{\text{BAMDP}} := \langle \mathcal{H}, \mathcal{A}, P_0, P_{\mathcal{H}}(h, a), \gamma \rangle. \tag{21}$$

2.5.3 Natural Resolution of the Exploration Dilemma

An interesting aspects of the Bayesian framework is how it naturally resolves the exploration-exploitation dilemma [12], [13]. Rather than treating exploration as a separate mechanism, it emerges naturally from the optimization of expected returns under uncertainty

$$J^{\pi}_{\text{Bayes}} := \mathbb{E}_{h_\infty \sim P^{\pi}_{\infty}} \left[\sum_{i=0}^{\infty} \gamma^i r_i \right]. \tag{22}$$

A Bayes-optimal policy achieves perfect balance between exploration and exploitation because:

1. It accounts for uncertainty through the posterior at each timestep;
2. It considers how this uncertainty will evolve in the future;
3. It weights future information gain by the discount factor γ.

2.5.4 The Optimal Bayesian Q-Function

For a Bayes-optimal policy π^*, we can define the optimal Bayesian Q-function as $Q^*(h_t, a_t) := Q^{\pi^*_{\text{Bayes}}}(h_t, a_t)$. This Q-function satisfies the optimal Bayesian Bellman equation

$$Q^*(h_t, a_t) = \mathcal{B}^*[Q^*](h_t, a_t), \tag{23}$$

where $\mathcal{B}^*[Q^*]$ is the optimal Bayesian Bellman operator

$$\mathcal{B}^*[Q^*](h_t, a_t) := \mathbb{E}_{h_{t+1} \sim P_{\mathcal{H}}(h_t, a_t)} \left[r_t + \gamma \max_{a'} Q^*(h_{t+1}, a') \right]. \tag{24}$$

2.6 Bayesian Exploration Networks

Model-free reinforcement learning takes a different approach to learning optimal behaviors compared to model-based methods. Rather than explicitly modeling the environment's dynamics, model-free approaches attempt to learn optimal policies from experience. Bayesian Exploration Networks (BENs) extend this idea into the Bayesian realm by characterizing uncertainty in the Bellman operator itself, instead of in the environment's transition dynamics [7].

> **Model-Free vs Model-Based:** While model-based approaches maintain explicit probabilistic models of the environment's dynamics, model-free methods like BEN directly learn mappings from states to values or actions [14]. This can be more computationally efficient but requires careful handling of uncertainty.

2.6.1 The Bootstrapping Perspective

Instead of modeling the full complexity of state transitions, we can use bootstrapping to estimate the optimal Bayesian Bellman operator directly. Given samples from the true reward-state distribution $r_t, s_{t+1} \sim P^*_{R,S}(s_t, a_t)$, we estimate $b_t = \beta_\omega(h_{t+1}) \coloneqq r_t + \gamma \max_{a'} Q_\omega(h_{t+1}, a')$.

This bootstrapping process can be viewed as a transformation of variables — mapping from the space of rewards and next states to a single scalar value. This significantly reduces the dimensionality of the problem while preserving the essential information needed for learning optimal policies [7].

> **Bootstrapped Distribution:** The samples b_t follow what we call the Bellman distribution $P^*_B(h_t, a_t; \omega)$, which captures the distribution of possible Q-value updates [7]. This distribution encapsulates both the environment's inherent randomness and our uncertainty about its true nature.

2.6.2 Sources of Uncertainty

When predicting future Q-values, BEN distinguishes between two types of uncertainty.

1. **Aleatoric Uncertainty**: The inherent randomness in the environment's dynamics that persists even with perfect knowledge.

   Example: Rolling a fair die — this uncertainty cannot be reduced with more data.

2. **Epistemic Uncertainty**: Our uncertainty about the true Bellman distribution itself. This represents our lack of knowledge about the environment and can be reduced through exploration and learning.

   Example: Determining whether a die is fair — this uncertainty is can be reduced with more data.

This separation of uncertainties allows BEN to distinguish between what is fundamentally unpredictable (aleatoric) and what can be learned through exploration (epistemic), leading to more efficient learning strategies [7].

2.6.3 Network Architecture

BEN implements this uncertainty handling through three neural networks [7]:

1. **Recurrent Q-Network**:

   At its core, BEN uses a recurrent neural network (RNN) to approximate the optimal Bayesian Q-function. The Q-network processes the entire history of interactions. We denote the output at timestep t as $q_t = Q_\omega(h_t, a_t) = Q_\omega\left(\hat{h}_{t-1}, o_t\right)$, where h_t represents the history up to time t, a_t is the action, $\hat{h}_{t-1}$ is the recurrent encoding of previous history, and o_t contains the current observation tuple $\{r_{t-1}, s_t, a_t\}$. By conditioning on history rather than just current state, BENs can capture how uncertainty evolves over time, making it capable of learning Bayes-optimal policies [7].

2. **Aleatoric Network**:

The aleatoric network models the inherent randomness in the environment. It uses normalizing flows to transform a simple base distribution (such as a standard Gaussian) into a more complex distribution $P_B(h_t, a_t, \varphi; \omega)$ over possible next-state Q-values, representing the aleatoric uncertainty in the Bellman operator, by applying the transformation $b_t = B(z_{\text{al}}, q_t, \varphi)$ [7], where

- $z_{\text{al}} \in \mathbb{R} \sim P_{\text{al}}$ is a base variable with a zero-mean, unit variance Gaussian P_{al};
- q_t is the Q-value from the recurrent network;
- and φ and ω represent the network parameters.

We optimize the parameters ω of the recurrent Q-network and the aleatoric network using the Mean Squared Bayesian Bellman Error (MSBBE), which satisfies the optimal Bayesian Bellman equation for our Q-function approximator.

3. **Epistemic Network**:

The epistemic network captures our uncertainty about the environment itself. The network maintains a dataset of bootstrapped samples $\mathcal{D}_\omega(h_t) := \{(b_i, h_i, a_i)\}_{i=0}^{t-1}$ collected from interactions with the environment. Each tuple in this dataset consists of

- a bootstrapped value estimate b_i;
- the history at that timestep h_i;
- the action taken a_i.

Given this dataset, we would ideally compute the posterior distribution $P_\Phi(\mathcal{D}_\omega(h_t))$ representing our refined beliefs about the environment after observing these samples. However, computing this posterior directly is typically intractable for complex environments [7]. Instead, BEN employs normalizing flows for variational inference.

The epistemic network learns a tractable approximation P_ψ parametrized by $\psi \in \Psi$ that aims to capture the essential characteristics of the true posterior. We optimize this approximation by minimizing the KL-divergence between our approximation and the true posterior:

$$\mathrm{KL}\big(P_\psi \parallel P_\Phi(\mathcal{D}_\omega(h_t))\big). \tag{25}$$

This optimization is performed indirectly by maximizing the Evidence Lower Bound (ELBO) $\mathrm{ELBO}(\psi; h, \omega)$, which is equivalent as proved by [7].

2.6.4 Training Process

The network is trained through a dual optimization process:

1. **MSBBE Optimization:** The Mean Squared Bayesian Bellman Error (MSBBE) is computed as the difference between the predictive optimal Bellman operator $B^+[Q_\omega]$ and Q_ω [7]:

$$\mathrm{MSBBE}(\omega; h_t, \psi) := \left\| B^+[Q_\omega](h_t, a_t) - Q_\omega(h_t, a_t) \right\|_\rho^2, \tag{26}$$

which is minimized to learn the parametrisation ω^*, satisfying the optimal Bayesian Bellman equation for our Q-function approximator, with ρ being an arbitrary sampling distribution with support over $\mathcal{A}$.

The predictive optimal Bellman operator can be obtained by taking expectations over variable b_t using the predictive optimal Bellman distribution $P_B(h_t, a_t; \omega)$:

$$B^+[Q_\omega](h_t, a_t) := \mathbb{E}_{b_t \sim P_B(h_t, a_t; \omega)}[b_t], \tag{27}$$

where $P_B(h_t, a_t; \omega) = \mathbb{E}_{\varphi \sim P_\Phi(\mathcal{D}_\omega(h_t))}[P_B(h_t, a_t, \varphi; \omega)]$.

This gives rise to a nested optimisation problem, as is common in model-free RL [7], which can be solved using two-timescale stochastic approximation [15]. In the case of BEN, we update the epistemic network parameters ψ using gradient descent on an asymptotically faster timescale than the function approximator parameters ω to ensure convergence to a fixed point, as proposed by [7].

2. **ELBO Optimization:** The Evidence Lower BOund (ELBO) serves as the optimization objective for training BEN's epistemic network. While minimizing the KL-divergence $\mathrm{KL}\left(P_\psi \parallel P_\Phi(\mathcal{D}_\omega(h_t))\right)$ directly would give us the most accurate approximation of the true posterior, computing this divergence is typically intractable. Instead, we can derive and optimize the ELBO, which provides a tractable lower bound on the model evidence [7].

By applying Baye's rule on this KL-divergence, [7] derives

$$\begin{aligned} &\mathrm{ELBO}(\psi; h_t, \omega) \\ &:= \mathbb{E}_{z_{\mathrm{ep}} \sim P_{\mathrm{ep}}}\left[\sum_{i=0}^{t-1}\left(B^{-1}(b_i, q_i, \varphi)^2 - \log\left|\ \partial_b B^{-1}(b_i, q_i, \varphi)\ \right|\right) - \log p_\Phi(\varphi)\right], \end{aligned} \tag{28}$$

where $\varphi = t_\psi(z_{\mathrm{ep}})$ and:

- z_{ep} is drawn from the base distribution P_{ep} (a standard Gaussian $\mathcal{N}\left(0, I^d\right)$);
- B^{-1} is the inverse of the aleatoric network's transformation;
- $\partial_b B^{-1}$ is the Jacobian of this inverse transformation;
- t_ψ represents the epistemic network's transformation.

> **Jacobian Term:** The term $\partial_b B^{-1}$ accounts for how the epistemic network's transformation changes the volume of probability space. This is important for maintaining proper probability distributions when using normalizing flows [16].

The ELBO objective breaks down into three key components:

1. A reconstruction term $B^{-1}(b_i, q_i, \varphi)^2$ that measures how well our model can explain the observed Q-values;
2. A volume correction term $\log\left|\partial_b B^{-1}(b_i, q_i, \varphi)\right|$ that accounts for the change in probability space;
3. A prior regularization term $\log p_\Phi(\varphi)$ that encourages the approximated posterior to stay close to our prior beliefs.

By minimizing the ELBO, we obtain an approximate posterior that balances accuracy with computational tractability, allowing BEN to maintain and update its uncertainty estimates efficiently during learning [7].

> **Training Dynamics:** The two optimization processes occur at different timescales, with epistemic updates happening more frequently than the Q-network updates. This separation ensures stable convergence while maintaining the ability to adapt to new information [15].

With this architecture, BEN can learn truly Bayes-optimal policies while maintaining the computational efficiency of model-free methods [7]. This makes it particularly well-suited for environments with sparse, delayed rewards where efficient exploration is important.

3 Theoretical Framework

In this section, we develop a theoretical framework for comparing GFlowNets and Bayesian Exploration Networks (BENs) in environments with delayed and sparse rewards. Our goal is to establish precise criteria for evaluating these different approaches to exploration and uncertainty handling.

3.1 Problem Formulation

Consider an environment with delayed rewards characterized by

- **Reward Delay:** The temporal gap T_{reward} between an action and its corresponding reward signal . We formally define this as

$$T_{\text{reward}} := \min\{t \mid s_t \in \mathcal{X}, r_t \neq 0\}. \tag{29}$$

  In our n-chain environment, T_{reward} corresponds to the chain length.

- **Reward Sparsity:** The proportion ρ of state-action pairs that yield non-zero rewards:

$$\rho := \frac{|\{(s,a) \in \mathcal{S} \times \mathcal{A} : \mathbb{E}[R(s,a)] \neq 0\}|}{|\mathcal{S} \times \mathcal{A}|}, \tag{30}$$

  where $R(\cdot)$ is some reward distribution.

These characteristics create distinct challenges, as discussed, for reinforcement learning algorithms.

1. The *temporal credit assignment problem* becomes more severe with increasing T_{reward}.
2. The *exploration efficiency* becomes critical as ρ decreases.
3. The *signal-to-noise ratio* in value estimation deteriorates with both T_{reward} and ρ.

3.1.1 Value Propagation Mechanisms

The algorithms differ in how they handle value propagation. GFlowNet value propagation maintains consistency between forward and backward flows through the trajectory balance constraint $Z \prod_{t=1}^{n} P_F(s_t|s_{t-1}) = F(x) \prod_{t=1}^{n} P_B(s_{t-1}|s_t)$.

BEN value propagation directly models the distribution of bootstrapped values through the estimated Bellman operator $b_t = r_t + \gamma \max_{a'} Q_\omega(h_{t+1}, a')$.

3.1.2 Uncertainty Representation

Both approaches maintain uncertainty estimates but through different mechanisms:

- GFlowNets implicitly capture uncertainty through the learned flow distribution;
- BENs explicitly separate aleatoric and epistemic uncertainty.

This leads to our central hypothesis.

> **Hypothesis:** In environments with highly delayed rewards (large T_{reward}), BEN's explicit uncertainty decomposition leads to more efficient learning compared to GFlowNets, particularly in early training stages.

However, this advantage diminishes as T_{reward} decreases. This hypothesis is supported by the following three observations.

1. BEN's direct modeling of the Bellman operator allows for faster value propagation.
2. The explicit separation of uncertainty types enables more targeted exploration.
3. GFlowNets must learn complete trajectories before gaining signal about reward structure.

3.1.3 Analytical Framework

To evaluate our hypothesis about the relative performance of GFlowNets and BENs in delayed reward environments, we establish three metrics that capture different aspects of learning and exploration efficiency.

1. **Sample Efficiency:** Measures how quickly each algorithm converges to optimal behavior through their respective loss functions.

   For GFlowNets, we track the trajectory balance loss $\mathcal{L}_{\text{TB}}(\tau) = \left(\log Z_\theta + \log \sum_{t=1}^{n} P_F(s_t | s_{t-1}; \theta) - \log R(x) - \log \sum_{t=1}^{n} P_B(s_{t-1} | s_t; \theta)\right)^2$.

   While for BEN, we monitor the Mean Squared Bayesian Bellman Error $\text{MSBBE}(\omega; h_t, \psi) = \| B^+[Q_\omega](h_t, a_t) - Q_\omega(h_t, a_t) \|_\rho^2$.

   These metrics allow us to quantify learning progress and compare convergence rates between algorithms as a function of reward delay T_{reward}.

2. **Distribution Matching:** Evaluates how well the learned policy matches the true underlying reward structure.

   In our n-chain environment, where terminal states are guaranteed to be reached, we measure the KL-divergence between the true terminal state distribution P (determined by rewards) and the empirical distribution Q generated by each algorithm $\text{KL}(P \parallel Q)$. This metric is particularly relevant for GFlowNets, as they explicitly aim to learn a sampling distribution proportional to the reward function.

3. **Exploration Efficiency:** Captures how effectively each algorithm explores the state space before converging to optimal behavior.

   We introduce two complementary metrics.

   - **State Coverage Ratio:** Measures the proportion of the state space explored over time as
   $$E(t) := \frac{|S_{\text{visited}}(t)|}{|S|}, \tag{31}$$
   where $S_{\text{visited}}(t)$ represents the set of unique states visited up to time t.
   - **Time-to-First-Success:** Quantifies initial exploration effectiveness as
   $$T_{\text{success}} := \min\{t \mid s_t \in \mathcal{X}, r_t > 0\}. \tag{32}$$

   This metric becomes increasingly important as reward delay T_{reward} grows, as it indicates how quickly each algorithm can discover successful trajectories in sparse reward settings.

Combining these metrics, we construct a evaluation framework that addresses three important aspects of performance:

1. Learning efficiency through loss convergence analysis;
2. Policy quality through distribution matching;
3. Exploration effectiveness through coverage and discovery time.

This framework allows us to investigate how the advantage of BEN's explicit uncertainty decomposition versus GFlowNet's flow-based approach vary with reward delay T_{reward} and sparsity ρ. In particular, we can test our hypothesis that BEN's advantages become more pronounced as T_{reward} increases by examining the correlation between reward delay and relative performance across the mentioned metrics.

4 Experimental Design

We implement a modified n-chain environment that serves as a testbed for studying delayed rewards. This environment presents properties that make it particularly suitable for our analysis.

> **N-Chain Environment:** A sequential decision-making environment with a branching structure, where rewards are only received at terminal states.

Parameters of the n-chain environment include a chain length n, controlling reward delay T_{reward}, a branching factor b, affecting exploration complexity, and terminal state rewards, determining optimal distributions. Adjusting the chain length n and branching factor b also allow us to control the reward sparsity ρ by proxy, as a longer chain involves more states, increasing sparsity. Similarly, increasing the branching factor b introduces more branches, again inreasing the number of states and, thus, the sparsity ρ as well.

The environment consists of three main components:

1. **State Space:** A chain of length n with a branching point at the middle ($\lfloor \frac{n}{2} \rfloor$), creating multiple possible trajectories. This results in a total of $\lfloor \frac{n}{2} \rfloor + b(n - \lfloor \frac{n}{2} \rfloor)$ possible states, and exactly b terminal states.
2. **Action Space:** At each state, an agent can move forward with the $\mathrm{FORWARD}$ action, or stay in terminal states with the $\mathrm{TERMINAL\_STAY}$ action. At the split point, the agent must choose a branch using the BRANCH_i action, where $i \in \{1, ..., b\}$. This results in a total of $2 + b$ actions.
3. **Reward Structure:** Generally, a reward function $R : \mathcal{X} \to \mathbb{R}$ is defined over terminal states $x \in \mathcal{X}$, creating a natural target distribution for sampling and clear optimal policies. For GFlowNets, the reward function is further constrained to the domain $\mathbb{R}_{>0}$, yielding a reward function $R : \mathcal{X} \to \mathbb{R}_{>0}$.

This design creates a sparse reward landscape — agents must execute sequences of $\lfloor \frac{n}{2} \rfloor - 1$ actions before reaching the branch point, where the chosen branch determines the final reward, followed by another $\lfloor \frac{n}{2} \rfloor$ actions to reach any terminal state. This structure allows us to precisely control both reward delay T_{reward} and sparsity ρ.

4.1 Evaluation Protocol

We evaluate each algorithm through the following experiment.

- **Base Configuration:**
  - Three terminal states with fixed rewards;
  - GFlowNet:
    - Rewards: $\{10, 20, 70\}$;
    - Exploration factor $\varepsilon = 0.1$;
  - BEN:
    - Rewards: $\{-500, 10, 200\}$;
    - Discount factor $\gamma = 0.9$.
- **Delay Variation Studies:**
  - Chain lengths $n \in \{3, 5, 7, 9, 11\}$;
  - Keeping terminal rewards fixed;
  - Measuring performance vs. delay T_{reward}.

For each configuration, we conduct 10 independent trials with different random seeds to minimize the impact of statistical variance. We then apply the framework discussed in Section 3.1.3 on the results of these three configurations for analysis.

4.2 Implementation Details

4.2.1 Environment Setup

Our implementation of the n-chain environment creates a decision space that enables precise control over reward delay and sparsity. The environment is implemented as a deterministic MDP, where each state is encoded as a composite tensor of shape `[n*3 + 4]`, consisting of a one-hot position encoding of length 3n to account for all three branches, as well as a one-hot branch encoding of length 4 (pre-split + 3 possible branches).

The state space is managed through a `NChainState` class that tracks three attributes:

1. *Position:* An integer in [0, n-1] indicating location in the chain;
2. *Branch:* An integer flag (-1 for pre-split, {0,1,2} for branch selection);
3. *Chain Length:* The parameter n that determines the delay between actions and rewards.

The action space is managed through a `NChainAction` enum and consists of the following five distinct actions:

- `TERMINAL_STAY`: Available only in terminal states;
- `FORWARD`: For progression along the chosen path;
- `BRANCH_0`, `BRANCH_1`, `BRANCH_2`: Available only at the split point.

The environment enforces strict action masking through a get_valid_actions method that returns only legitimate actions for each state. This creates three distinct decision phases:

1. Pre-split: Only `FORWARD` actions are valid;
2. Split-point: Only `BRANCH_i` actions are valid, for $i \in \{0, 1, 2\}$;
3. Post-split: `FORWARD` until terminal, then `TERMINAL_STAY`.

For interaction with the environment, the implementation provides methods like `step`, `reset`, as well as utility methods like state-to-tensor conversions. This approach allows for systematic variation of both reward delay through the chain length n, and reward sparsity through the ratio of rewarding states to total states.

4.2.2 Network Architectures

The GFlowNet implementation consists of three primary components working in concert to learn flow-matching policies:

1. *State Encoder:* A multi-layer perceptron that processes state tensors of shape `[batch_size, state_dim]` into a learned encoding of shape `[batch_size, hidden_dim]`. This encoding captures the essential features of each state necessary for policy decisions.

2. *Forward Policy Network:* A single dense layer that transforms the state encoding into a forward policy distribution over actions, outputting tensors of shape `[batch_size, num_actions]`. This network determines the probabilities of taking each possible action from the current state.

3. *Backward Policy Network:* For the n-chain environment, this component is simplified due to the tree structure of the state space. Since each non-initial state has exactly one parent, the backward policy becomes deterministic, requiring only a lightweight network layer to maintain architectural symmetry.

Additionally, we maintain a scalar parameter representing $\log Z$ (the partition function), which is important for our flow matching approach and is learned alongside the network parameters.

The BEN implementation is heavily based on the implementation provided by [7], but is adapted for the n-chain environment. It comprises three interacting networks:

1. *Q-Bayes Network:* A recurrent neural network that processes observation tuples (state, action, reward) to generate Q-values and maintain a history encoding. The network accepts inputs of shape `[batch_size, state_dim + action_dim + reward_dim]` and outputs both Q-values `[batch_size, num_actions]` and a RNN hidden state `[batch_size, rnn_hidden_dim]` for further processing.
2. *Aleatoric Network:* Implements a normalizing flow to model the inherent randomness in Bellman updates through:
   - A conditioning network (ConditionerMLP) [7] that generates parameters for the flow based on Q-values and RNN state;
   - An inverse autoregressive flow that transforms a base distribution into the desired Bellman distribution.
3. *Epistemic Network:* Another normalizing flow that captures uncertainty about the environment itself, transforming a base variable $z_{\text{ep}} \in \mathbb{R}^d$ into the parameter space that defines our beliefs about the environment.

4.2.3 Training Procedures

Training of the GFlowNet follows a tempered exploration strategy where:

1. The forward policy is "softened" during training using an ε-greedy approach with $\varepsilon = 0.1$, allowing for off-policy exploration while maintaining flow-matching properties:

   ```
   (1 - self.epsilon) * policy + self.epsilon * random_policy.
   ```

2. The trajectory balance loss is minimized using stochastic gradient descent with the Adam optimizer [17]:

   ```
   L_TB(tau) = (log_Z + sum_log_pf - log_R - sum_log_pb).pow(2),
   ```

   where `tau` represents a trajectory, `sum_log_pf` represents the sum of log forward probabilities over `tau`, and `sum_log_pb` similarly represents the sum of log backward probabilities over `tau`.

BEN employs a two-timescale optimization process:

1. *Fast Timescale:* Updates to the epistemic network parameters ψ through ELBO minimization:

   ```
   ELBO(psi; h, omega) = -log_p - torch.mean(log_q) - 1 / (time_period + 1) * prior,
   ```

   where `log_p` represents the base variable z_{al} obtained from $B^{-1}(b_i, q_i, \varphi)^2$, `torch.mean(log_q)` represents the log Jacobian of this base variable $\log|\partial_b B^{-1}(b_i, q_i, \varphi)|$, and `1 / (time_period + 1) * prior` represents the log prior $\log p_\Phi(\varphi)$.
2. *Slow Timescale:* Updates to the Q-network parameters ω through MSBBE minimization:

   ```
   MSBBE(omega; h_t, psi) = torch.abs((b1 - q) * (b2 - q)),
   ```

   where `b1` and `b2` represent two samples from the predictive Bellman operator $B^+[Q_\omega](h_t, a_t)$, and `q` represents the Q-value obtained from $Q_\omega(h_t, a_t)$. This operation is similar to a simple squared error, with the only difference being the use of two different samples, minimizing statistical variance.

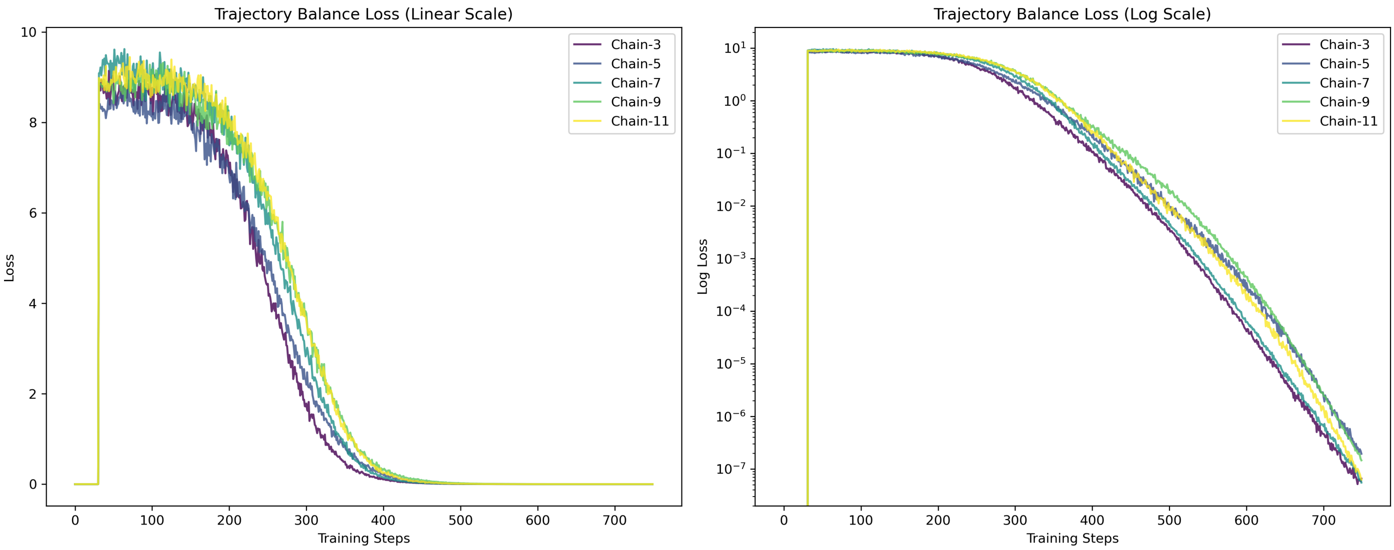

Figure 1: Mean GFlowNet trajectory balance loss across chain lengths.

This separation of timescales ensures stable convergence while maintaining the ability to adapt to new information, controlled by the discount factor γ. For details about hyperparameter selection, we refer to Appendix A.

5 Results and Analysis

5.1 GFlowNet

TODO

FACT CHECK!

- Run the statistical significance tests and calculate difference in μ's of terminal rewards
- Find out what Cohen's d is (effect sizes)

5.1.1 Training Stability and Loss Dynamics

The trajectory balance loss follows a consistent pattern of reduction across all chain lengths, with three distinct phases:

- *Early Stage:* High loss values ($\mu \approx 6.8$-7.4) with substantial variance;
- *Mid Stage:* Significant reduction ($\mu \approx 1.2$-1.7) with moderate variance;
- *Late Stage:* Near-zero loss ($\mu < 0.003$) with minimal variance.

This progression is statistically significant across all transitions ($p < 0.001$), with particularly large effect sizes (Cohen's $d > 2.0$) between early and late stages, indicating robust convergence regardless of chain length, as seen in Figure 1.

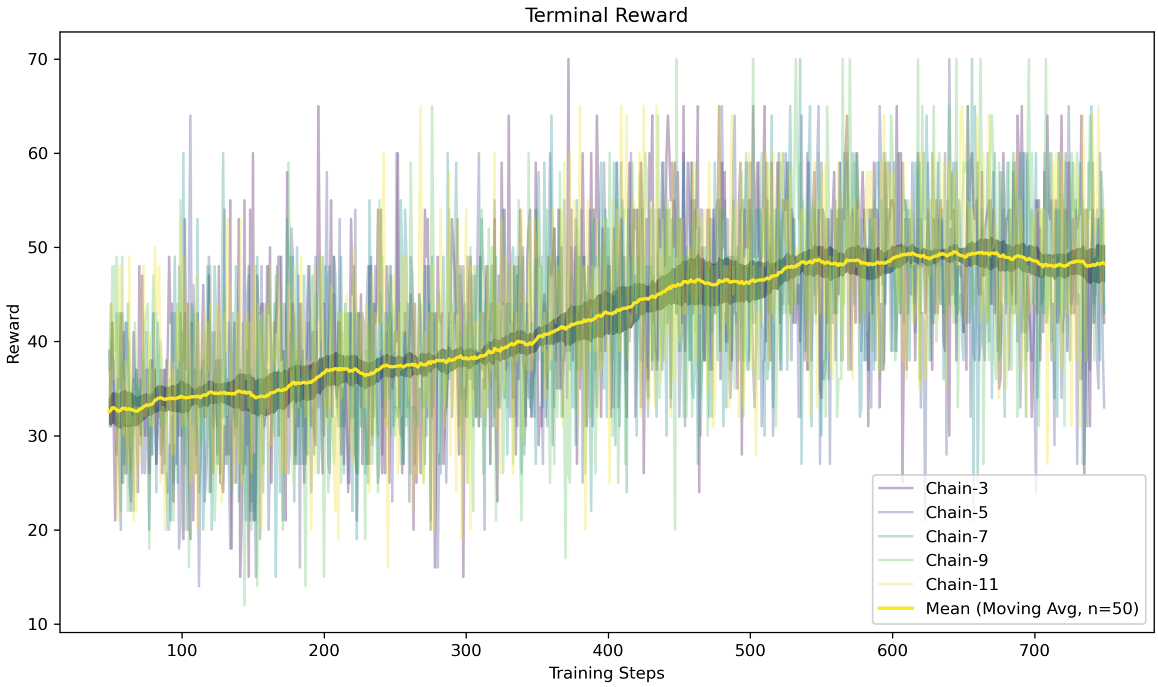

Figure 2: Mean GFlowNet terminal rewards across chain lengths with a moving average (window size: 50).

5.1.2 Solution Quality Evolution

Terminal rewards show a monotonic improvement pattern, as seen in Figure 2. The improvement in terminal rewards shows a consistent "two-step" enhancement, with moderate-to-large effect sizes (0.75-1.0) between consecutive stages and large effect sizes (1.4-1.9) between early and late stages.

Notably, longer chains ($n \geq 9$) demonstrate larger improvements in terminal rewards, with $n = 9$ showing the most substantial gain ($\Delta\mu \approx 16$ between early and late stages). This suggests that the algorithm becomes especially effective at optimizing longer sequences as training progresses.

The time-to-first-success is trivial for the n-chain environment: since the agent must always move forward, it is simply the chain length n.

Policy Quality Assessment

As GFlowNets aim to learn policies proportional to their reward distributions, we use the KL-divergence between the true reward distribution R_i and the learned policy P_i to assess the quality of the learned policy for $i \in \{3, 5, 7, 9, 11\}$ corresponding to the different chain lengths. The KL-divergence measurements look as follows:

- *Chain length 3:* $\mathrm{KL}(P_3 \mid R_3) = 0.00401$;
- *Chain length 5:* $\mathrm{KL}(P_5 \mid R_5) = 0.02062$;
- *Chain length 7:* $\mathrm{KL}(P_7 \mid R_7) = 0.01696$;
- *Chain length 9:* $\mathrm{KL}(P_9 \mid R_9) = 0.00385$;

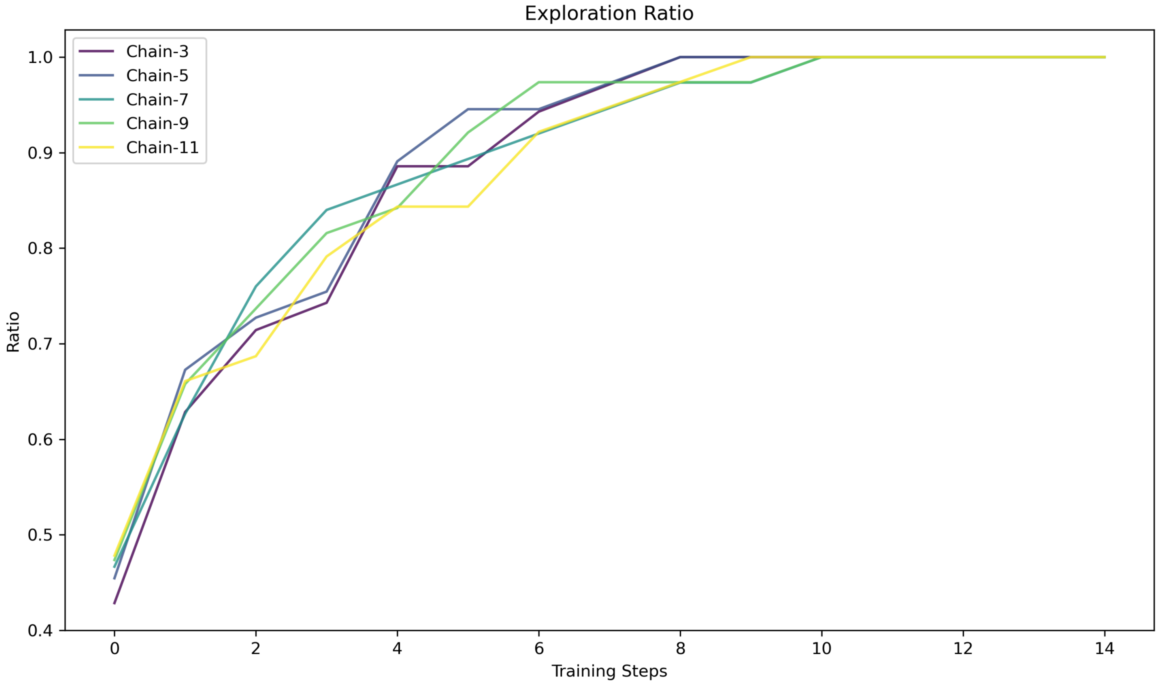

Figure 3: Mean GFlowNet forward and backward entrody across chain lengths.

- *Chain length 11:* $\mathrm{KL}(P_{11} \mid R_{11}) = 0.00701$;
- *Mean chain length:* $\overline{\mathrm{KL}(P \mid R)} = 0.01049$.

Looking at these measurements, the relationship between chain length and distributional accuracy does not seem follow a simple monotonic pattern. We conjecture that this is a consequence of the small sample size of 10 training runs per chain length, and as such, the KL values should converge given enough runs.

Looking at the average KL-divergence across all chain lengths, we conclude that the learned policy is very close to the true reward distribution.

5.1.3 Exploration-Exploitation Balance

The state coverage ratio exhibits rapid convergence to optimal exploration (1.0) by mid-stage of only 15 iterations, as seen in Figure 3, as well as:

- Statistically significant transition ($p < 0.005$) from early to mid stage;
- Perfect maintenance of exploration in late stage ($\sigma = 0$).

This indicates that GFlowNet quickly learns to fully explore the solution space and maintains this behavior throughout training. This also suggests that the state space is too small to pose a serious challenge for the model, which could be accomodated by increasing the number of critical decision points (by increasing the number branches in the environment).

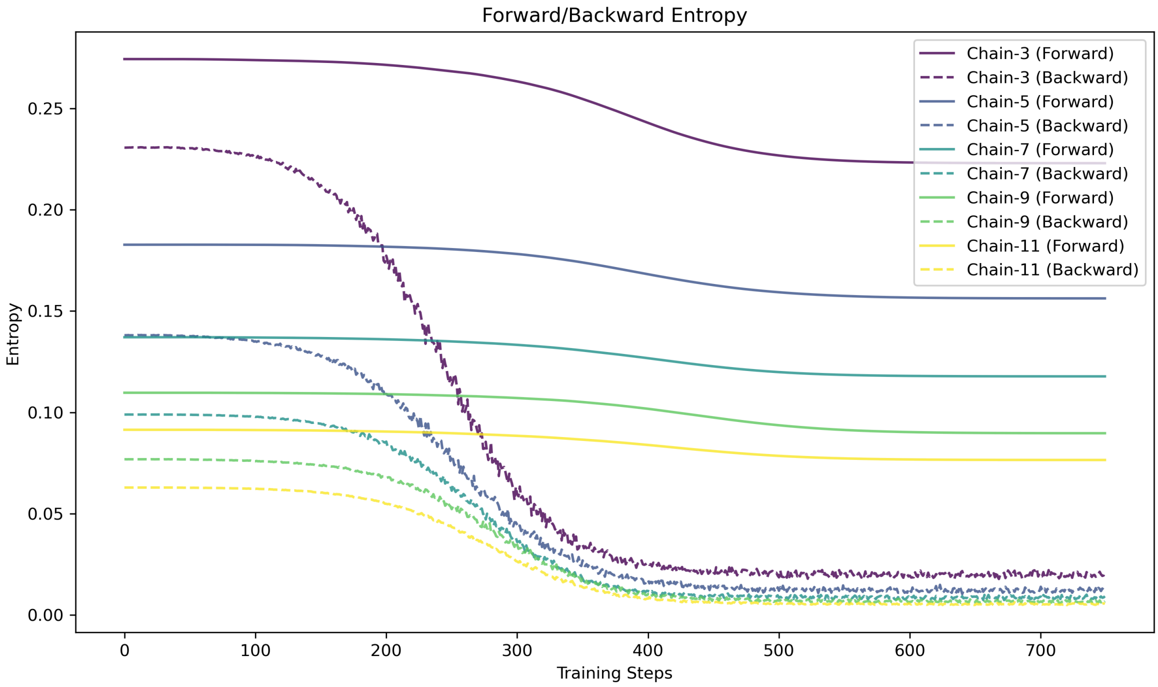

Figure 4: Mean GFlowNet forward and backward entropy across chain lengths.

5.1.4 Information Theoretic Measures

NOTE

Maybe move this to the appendix as we haven't calculated entropies for BEN.

Both forward and backward entropy demonstrate reduction across training, as seen in Figure 4.

- *Forward Entropy:* Shows consistent, statistically significant decreases ($p < 10^{-82}$) across all stages, with effect sizes growing with chain length;
- *Backward Entropy:* Exhibits the most dramatic reductions among all metrics, with effect sizes ranging from 8.5 to 16.3 between early and late stages.

The relative magnitude of entropy reduction remains notably consistent across chain lengths, suggesting a scale-invariant learning process.

5.1.5 Chain Length Sensitivity

The analysis reveals some chain length-dependent effects:

- Longer chains ($n \geq 9$) show higher terminal rewards in late stages;
- Convergence stability (measured by loss variance) remains consistent across chain lengths;
- Information theoretic measures scale proportionally with chain length while maintaining similar convergence patterns.

This suggests that the algorithm's learning dynamics remain robust across different problem scales.

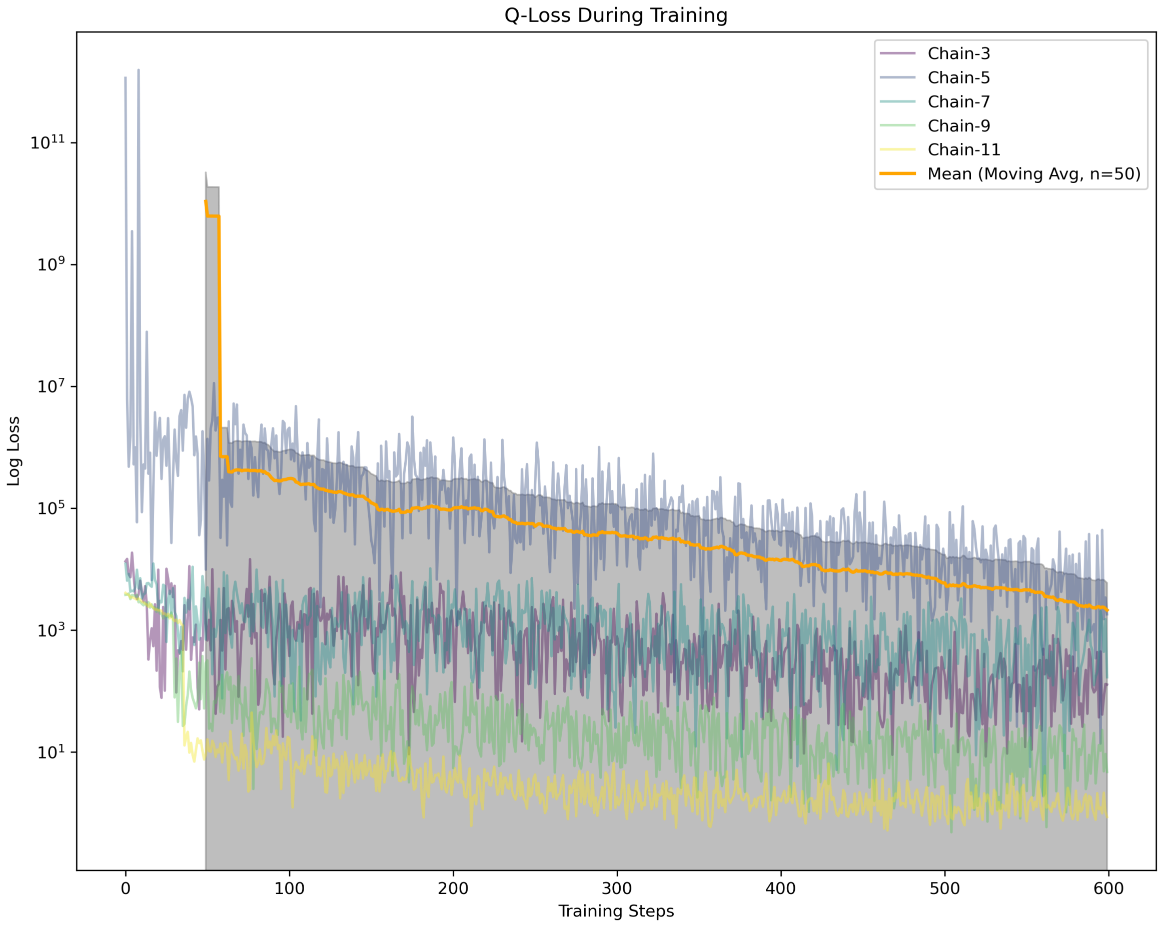

Figure 5: Mean BEN Q-learning (MSBBE) loss across chain lengths.

In short, the statistical analysis reveals structured convergence patterns that combine rapid initial improvement with stable late-stage optimization. The consistent statistical significance ($p < 0.001$) across multiple metrics and stages suggests that GFlowNet's learning process is both reliable and scalable across different chain lengths.

5.2 BEN

5.2.1 Training Stability and Loss Dynamics

- **Q-Learning Loss**: The Q-Learning loss demonstrates a consistent and statistically significant reduction across all chain lengths, as seen in Figure 5, with three notable characteristics.
  - *Early Stage Volatility:* High variance in early training (σ ranging from 610 to 2948);
  - *Mid Stage Stabilization:* Sharp reduction in both mean and variance;
  - *Late Stage Refinement:* Convergence to stable, low values.

  Particularly noteworthy is the scale-dependent convergence rate — longer chains ($n \geq 7$) show more gradual convergence, while shorter chains achieve stability more quickly.

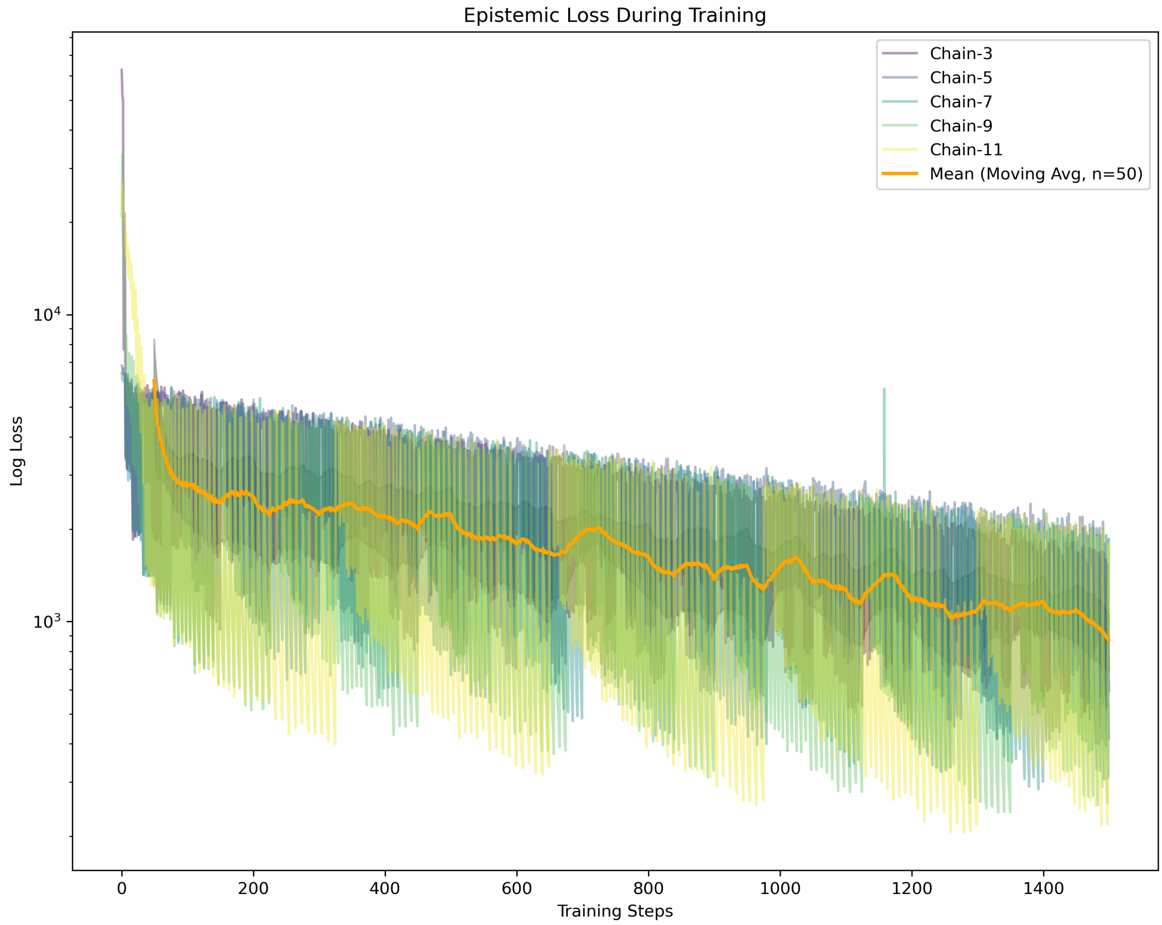

Figure 6: Mean BEN epistemic (ELBO) loss across chain lengths.

The magnitude of improvement (effect size) between early and late stages increases with chain length, suggesting that longer chains require more substantial transformations in the learning process.

- **Epistemic Loss**: The epistemic uncertainty shows a more complex pattern than the Q-learning loss, as seen in Figure 6.
  - For $n = 3$: Monotonic decrease (-2257.74 mean difference, $p < 0.001$);
  - For $n = 5$: Sharp initial drop followed by gradual decline;
  - For $n \in \{7, 9, 11\}$: Non-monotonic behavior with occasional increases.

This suggests that uncertainty management becomes more challenging with longer chains, possibly due to the expanded state space.

5.2.2 Reward Dynamics

The reward patterns exhibit rather unpromising behavior as seen in Figure 7. The statistical tests yield the following results:

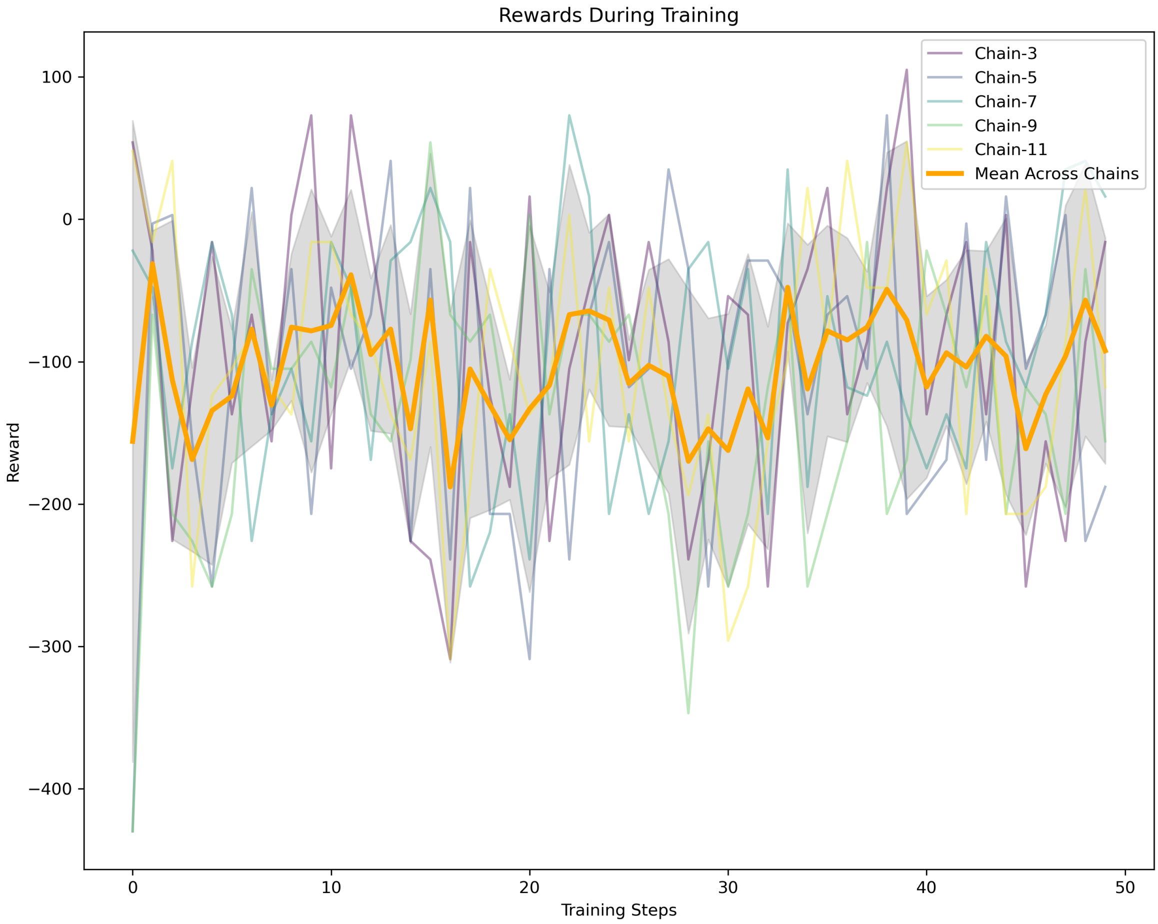

Figure 7: Mean BEN terminal rewards across chain lengths.

- *Short Chains ($n \in \{3, 5\}$):* No statistically significant changes between stages;
- *Medium Chains ($n \in \{7, 9\}$):* Temporary dip in mid-stage performance;
- *Long Chains ($n = 11$):* Significant improvement in late stage ($p < 0.05$).

The statistical significance of stage-wise improvements increases with chain length, suggesting that longer chains benefit more from extended training. These statistical results, however, do not seem to capture the apparent "trend"of the rewards (which is seemingly random) — the analysis would benefit from further investigation on this part.

5.2.3 Cumulative Returns

The cumulative returns, probably the most definitive metric with regards to the quality of the learned policy, can be seen in Figure 8. They show:

- Monotonic decrease across all chain lengths;
- Larger effect sizes (-7.13 to −8.59) for longer chains;
- *Statistical Significance:* $p < 10^{-10}$ for all chain lengths.

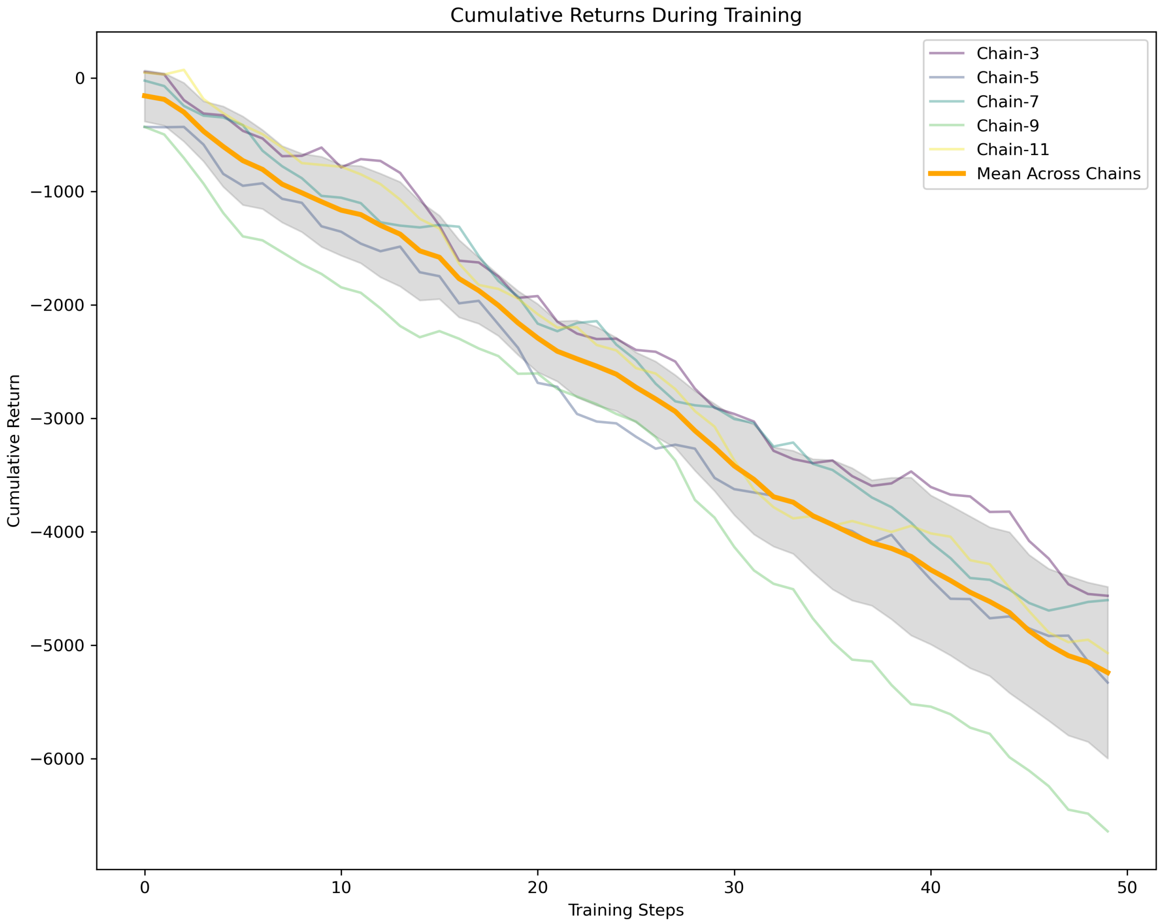

Figure 8: Mean BEN cumulative returns across chain lengths.

The relative magnitude of stage-wise change remains consistent across chain lengths. *This is critical as it indicates that the model is not learning optimal policies.*

5.2.4 Exploration-Exploitation Balance

Due to unfortunate events and time constraints, we only have access to the final state coverage ratio across all chain lengths for BEN. For all chain lengths, these final state coverage ratios were all 1, by which we can only conclude that BEN has effectively explored the entire state space. Increasing the state space beyond what the models (both GFlowNet and BEN) are capable of exploring to completion before finishing their training would be more interesting, as this would better showcase the exploration-exploitation capabilities of the models and the trade-offs they have to make.

5.3 Comparison Between GFlowNet and BEN

5.3.1 Learning Stability and Convergence

GFlowNet demonstrates stable convergence properties across all chain lengths, with its trajectory balance loss consistently decreasing through well defined phases. In contrast, BEN's learning trajectory shows more volatile behavior, especially in its epistemic loss patterns. This difference becomes more pronounced with increasing chain length, indicating that GFlowNet maintains consistent learning dynamics across problem scales, while BEN's stability deteriorates with increasing chain length. The most striking difference appears in their convergence behaviors.

- GFlowNet achieves near-zero loss values (< 0.003) in its late stages across all chain lengths;
- BEN's Q-learning loss shows persistent fluctuations, particularly in longer chains;
- BEN's epistemic loss exhibits non-monotonic behavior for $n \geq 7$, indicating challenges in uncertainty estimation at larger scales.

5.3.2 Reward Optimization

The contrast in reward optimization capabilities is noteworthy:

- GFlowNet shows consistent improvement in terminal rewards, with larger gains in longer chains ($n \geq 9$);
- BEN displays concerning patterns in both terminal rewards and cumulative returns, with monotonic decreases across all chain lengths.

The statistical significance of these differences ($p < 10^{-10}$ for BEN's declining returns) strongly suggests that GFlowNet's flow-based approach is better suited to this environment's reward structure.

5.3.3 Exploration Efficiency

Due to the unfortunately missing exploration data for BEN, a fair comparison between the two cannot be made, except for the fact they both managed to completely explore the state space.

5.3.4 Scale Sensitivity

The algorithms show markedly different responses to increasing chain length. GFlowNet maintains consistent convergence patterns, shows improved reward optimization with longer chains, and exhibits stable entropy reduction across scales. BEN shows degrading stability with longer chains, demonstrates increasingly volatile epistemic uncertainty, and experiences more severe performance degradation at larger scales.

5.4 Hypothesis Evaluation

Revisiting our original hypothesis that "BEN's explicit uncertainty decomposition leads to more efficient learning compared to GFlowNets, particularly in early training stages," our results suggest we must reject this hypothesis for the n-chain environment. The data instead supports a contrary conclusion:

> **Our Finding:** GFlowNet's flow-based approach appears fundamentally better suited to environments with deterministic dynamics and delayed rewards, while BEN's explicit uncertainty decomposition may be introducing unnecessary complexity for this particular class of problems.

This finding hints on an interesting nuance: while BEN's sophisticated uncertainty handling might be valuable in stochastic environments, it may be disadvantageous in deterministic scenarios where simpler flow-matching approaches suffice.

5.4.1 Architectural Implications

GFlowNet's success suggests that for environments with deterministic transitions and delayed rewards, explicitly modeling flow distributions might be more effective than maintaining separate aleatoric and epistemic uncertainty estimates. This insight could inform algorithm design, especially for scenarios where reward delay is a primary challenge. This comparative analysis suggests that the choice between these approaches should be guided by the specific characteristics of the target environment, with GFlowNet showing clear advantages in deterministic, delayed-reward scenarios.

6 Conclusion

NOTE

- Summary of contributions
- Key insights
- Future work directions

6.1 Future Research

NOTE

Everything I think I could have done better essentially.

- Stochastic Analysis:
  - Introducing random rewards drawn from distributions;
  - Terminal states $x \in \mathcal{X}$ rewards: $R_x \sim \mathcal{N}(\mu_x, \sigma^2)$;
  - Testing robustness to uncertainty.
- Experiment with impact of changing:
  - exploration factor ε;
  - discount factor γ.
- Mention more compute needed than what was available in these experiments for more interesting results.
  - This project could be considered a preliminary experiment to more serious experiments requiring more compute and/or time.

Bibliography

[1] R. S. Sutton and Barto, *Reinforcement learning : an introduction /*, Second edition. in Adaptive computation and machine learning. Cambridge, Massachusetts ;: The MIT Press, 2020.

[2] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. D. Turck, and P. Abbeel, "VIME: Variational Information Maximizing Exploration." [Online]. Available: https://arxiv.org/abs/1605.09674

[3] A. Harutyunyan *et al.*, "Hindsight Credit Assignment." [Online]. Available: https://arxiv.org/abs/1912.02503

[4] I. Osband, C. Blundell, A. Pritzel, and B. V. Roy, "Deep Exploration via Bootstrapped DQN." [Online]. Available: https://arxiv.org/abs/1602.04621

[5] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven Exploration by Self-supervised Prediction." [Online]. Available: https://arxiv.org/abs/1705.05363

[6] E. Bengio, M. Jain, M. Korablyov, D. Precup, and Y. Bengio, "Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation." [Online]. Available: https://arxiv.org/abs/2106.04399

[7] M. Fellows, B. Kaplowitz, C. S. de Witt, and S. Whiteson, "Bayesian Exploration Networks." [Online]. Available: https://arxiv.org/abs/2308.13049

[8] N. Malkin, M. Jain, E. Bengio, C. Sun, and Y. Bengio, "Trajectory balance: Improved credit assignment in GFlowNets." [Online]. Available: https://arxiv.org/abs/2201.13259

[9] M. L. Puterman, *Markov decision processes : Discrete stochastic dynamic programming*. in Wiley series in probability and mathematical statistics. Applied probability and statistics section. New York: Wiley, 1994.

[10] R. Bellman, *Dynamic programming /*. in A Rand corporation research study. Princeton,N.J: Princeton univer.press, 1962.

[11] A. Hallak, D. D. Castro, and S. Mannor, "Contextual Markov Decision Processes." [Online]. Available: https://arxiv.org/abs/1502.02259

[12] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, "Convex Optimization: Algorithms and Complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 5–6, pp. 359–483, 2015, doi: 10.1561/2200000049.

[13] M. O. Duff and A. G. Barto, "Optimal learning: computational procedures for bayes-adaptive markov decision processes," 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:118819159

[14] P. Dayan and Y. Niv, "Reinforcement learning: The Good, The Bad and The Ugly," *Current opinion in neurobiology*, vol. 18, no. 2, pp. 185–196, 2008.

[15] V. S. Borkar, *Stochastic Approximation : A Dynamical Systems Viewpoint*, 1st ed. 2008. in Texts and Readings in Mathematics ; 48. Gurgaon: Hindustan Book Agency, 2008.

[16] D. Rezende and S. Mohamed, “Variational Inference with Normalizing Flows,” in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., in Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 2015, pp. 1530–1538. [Online]. Available: https://proceedings.mlr.press/v37/rezende15.html

[17] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” [Online]. Available: https://arxiv.org/abs/1412.6980

Appendix

A Hyperparameter selection

GFlowNet Hyperparameters:

- Hidden dimension: 64;
- Learning rate: 1e-4;
- Exploration ε: 0.1;
- Batch size: 32.

BEN Hyperparameters:

- RNN hidden dimension: 64;
- Learning rate (Q-network): 1e-4;
- Learning rate (Epistemic network): 1e-4;
- Base dimension (z_{ep}): 8;
- Discount factor γ: 0.9;
- Batch size: 32.

B Policy Quality Assessment Results

B.1 GFlowNet

```
Chain length 3, KL divergence: 0.00401
Chain length 5, KL divergence: 0.02062
Chain length 7, KL divergence: 0.01696
Chain length 9, KL divergence: 0.00385
Chain length 11, KL divergence: 0.00701
Mean KL divergence across all files: 0.01049
```

Listing 1: Quality assessment by KL divergence between true and observed reward distributions.

C Statistical Significance Tests

C.1 GFlowNet

```
==================================================
Analysis for trajectory_balance_loss
==================================================

Summary Statistics:

Early stage:
  Mean   : 6.82581
  Std    : 2.81166
  Median : 8.15058

Mid stage:
  Mean   : 0.931558
  Std    : 1.26559
  Median : 0.262708

Late stage:
  Mean   : 0.000422112
  Std    : 0.000892446
  Median : 1.66286e-05

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 3.9726e-47
  Significant  : Yes
  Effect size  : -2.70345
  Mean diff    : -5.89425
  Median diff  : -7.88788

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.59106e-84
  Significant  : Yes
  Effect size  : -1.04048
  Mean diff    : -0.931136
  Median diff  : -0.262691

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 9.29597e-48
  Significant  : Yes
  Effect size  : -3.43304
  Mean diff    : -6.82538
  Median diff  : -8.15057

==================================================
Analysis for terminal_reward
==================================================
```

```
Summary Statistics:

Early stage:
  Mean   : 35.3644
  Std    : 9.19597
  Median : 36

Mid stage:
  Mean   : 44.1653
  Std    : 9.85363
  Median : 43

Late stage:
  Mean   : 49.2667
  Std    : 7.42465
  Median : 49

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 4.4844e-21
  Significant  : Yes
  Effect size  : 0.923453
  Mean diff    : 8.80095
  Median diff  : 7

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 4.06779e-10
  Significant  : Yes
  Effect size  : 0.584742
  Mean diff    : 5.10134
  Median diff  : 6

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 6.93379e-48
  Significant  : Yes
  Effect size  : 1.66348
  Mean diff    : 13.9023
  Median diff  : 13

==================================================
Analysis for exploration_ratio
==================================================

Summary Statistics:
```

```
Early stage:
  Mean   : 0.992713
  Std    : 0.0505792
  Median : 1

Mid stage:
  Mean   : 1
  Std    : 0
  Median : 1

Late stage:
  Mean   : 1
  Std    : 0
  Median : 1

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00433248
  Significant  : Yes
  Effect size  : 0.20376
  Mean diff    : 0.00728745
  Median diff  : 0

MID-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 1
  Significant  : No
  Effect size  : 0
  Mean diff    : 0
  Median diff  : 0

EARLY-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00382047
  Significant  : Yes
  Effect size  : 0.20376
  Mean diff    : 0.00728745
  Median diff  : 0

==================================================
Analysis for forward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.272931
  Std    : 0.00158218
```

```
  Median : 0.27355

Mid stage:
  Mean   : 0.248951
  Std    : 0.0136987
  Median : 0.24999

Late stage:
  Mean   : 0.223683
  Std    : 0.00105377
  Median : 0.223144

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.39375e-82
  Significant  : Yes
  Effect size  : -2.45926
  Mean diff    : -0.0239799
  Median diff  : -0.0235602

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.50099e-84
  Significant  : Yes
  Effect size  : -2.60084
  Mean diff    : -0.0252673
  Median diff  : -0.0268458

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.05232e-83
  Significant  : Yes
  Effect size  : -36.6358
  Mean diff    : -0.0492472
  Median diff  : -0.0504061

==================================================
Analysis for backward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.20616
  Std    : 0.0309031
  Median : 0.221136

Mid stage:
```

```
  Mean   : 0.0424959
  Std    : 0.0271554
  Median : 0.0293473

Late stage:
  Mean   : 0.0199545
  Std    : 0.00105893
  Median : 0.0197963

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.46542e-82
  Significant  : Yes
  Effect size  : -5.62619
  Mean diff    : -0.163664
  Median diff  : -0.191789

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.65605e-67
  Significant  : Yes
  Effect size  : -1.17304
  Mean diff    : -0.0225415
  Median diff  : -0.00955104

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.10488e-83
  Significant  : Yes
  Effect size  : -8.5163
  Mean diff    : -0.186205
  Median diff  : -0.20134
```

Listing 2: GFlowNet, Chain Length $n = 3$.

```
==================================================
Analysis for trajectory_balance_loss
==================================================

Summary Statistics:

Early stage:
  Mean   : 6.69499
  Std    : 2.70338
  Median : 7.97144

Mid stage:
  Mean   : 1.19563
```

```
  Std    : 1.44005
  Median : 0.473378

Late stage:
  Mean   : 0.00140177
  Std    : 0.00252725
  Median : 0.000115058

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 3.9726e-47
  Significant  : Yes
  Effect size  : -2.5391
  Mean diff    : -5.49935
  Median diff  : -7.49807

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.68221e-84
  Significant  : Yes
  Effect size  : -1.1728
  Mean diff    : -1.19423
  Median diff  : -0.473263

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 9.29597e-48
  Significant  : Yes
  Effect size  : -3.5016
  Mean diff    : -6.69358
  Median diff  : -7.97133

==================================================
Analysis for terminal_reward
==================================================

Summary Statistics:

Early stage:
  Mean   : 33.2955
  Std    : 8.32213
  Median : 33

Mid stage:
  Mean   : 41.8629
  Std    : 8.77181
  Median : 42
```

```
Late stage:
  Mean   : 46.7922
  Std    : 8.57252
  Median : 48

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 6.02021e-25
  Significant  : Yes
  Effect size  : 1.00204
  Mean diff    : 8.56736
  Median diff  : 9

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 4.86562e-10
  Significant  : Yes
  Effect size  : 0.568362
  Mean diff    : 4.92925
  Median diff  : 6

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 2.30758e-47
  Significant  : Yes
  Effect size  : 1.59756
  Mean diff    : 13.4966
  Median diff  : 15

==================================================
Analysis for exploration_ratio
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.993375
  Std    : 0.0471553
  Median : 1

Mid stage:
  Mean   : 1
  Std    : 0
  Median : 1

Late stage:
  Mean   : 1
  Std    : 0
```

```
  Median : 1

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00433248
  Significant  : Yes
  Effect size  : 0.198686
  Mean diff    : 0.00662495
  Median diff  : 0

MID-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 1
  Significant  : No
  Effect size  : 0
  Mean diff    : 0
  Median diff  : 0

EARLY-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00382047
  Significant  : Yes
  Effect size  : 0.198686
  Mean diff    : 0.00662495
  Median diff  : 0

==================================================
Analysis for forward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.182292
  Std    : 0.000582602
  Median : 0.182608

Mid stage:
  Mean   : 0.17099
  Std    : 0.00687163
  Median : 0.171686

Late stage:
  Mean   : 0.156909
  Std    : 0.000865732
  Median : 0.156479

Statistical Tests:
```

```
EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.39375e-82
  Significant  : Yes
  Effect size  : -2.3177
  Mean diff    : -0.0113021
  Median diff  : -0.010922

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.50099e-84
  Significant  : Yes
  Effect size  : -2.87511
  Mean diff    : -0.0140806
  Median diff  : -0.0152075

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.05232e-83
  Significant  : Yes
  Effect size  : -34.3965
  Mean diff    : -0.0253826
  Median diff  : -0.0261295

==================================================
Analysis for backward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.124858
  Std    : 0.0162292
  Median : 0.13268

Mid stage:
  Mean   : 0.0295478
  Std    : 0.0190841
  Median : 0.0201874

Late stage:
  Mean   : 0.0123233
  Std    : 0.000816268
  Median : 0.0122581

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
```

```
  P-value      : 1.53827e-82
  Significant  : Yes
  Effect size  : -5.38039
  Mean diff    : -0.0953099
  Median diff  : -0.112493

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 6.52618e-68
  Significant  : Yes
  Effect size  : -1.27524
  Mean diff    : -0.0172245
  Median diff  : -0.00792933

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.10488e-83
  Significant  : Yes
  Effect size  : -9.79385
  Mean diff    : -0.112534
  Median diff  : -0.120422
```

Listing 3: GFlowNet, Chain Length $n = 5$.

```
==================================================
Analysis for trajectory_balance_loss
==================================================

Summary Statistics:

Early stage:
  Mean   : 7.40181
  Std    : 2.92816
  Median : 8.63355

Mid stage:
  Mean   : 1.42542
  Std    : 1.80647
  Median : 0.42669

Late stage:
  Mean   : 0.000556402
  Std    : 0.00114077
  Median : 2.25787e-05

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 4.00893e-47
```

```
  Significant  : Yes
  Effect size  : -2.45655
  Mean diff    : -5.9764
  Median diff  : -8.20686

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.68221e-84
  Significant  : Yes
  Effect size  : -1.11547
  Mean diff    : -1.42486
  Median diff  : -0.426668

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 9.29597e-48
  Significant  : Yes
  Effect size  : -3.57458
  Mean diff    : -7.40126
  Median diff  : -8.63353

==================================================
Analysis for terminal_reward
==================================================

Summary Statistics:

Early stage:
  Mean   : 36.0729
  Std    : 8.40941
  Median : 37

Mid stage:
  Mean   : 42.4153
  Std    : 8.48486
  Median : 43

Late stage:
  Mean   : 47.9686
  Std    : 8.64655
  Median : 48

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 2.84163e-14
  Significant  : Yes
  Effect size  : 0.750833
  Mean diff    : 6.34245
```

```
  Median diff  : 6

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 6.5447e-12
  Significant  : Yes
  Effect size  : 0.64829
  Mean diff    : 5.5533
  Median diff  : 5

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 2.47514e-39
  Significant  : Yes
  Effect size  : 1.39477
  Mean diff    : 11.8958
  Median diff  : 11

==================================================
Analysis for exploration_ratio
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.992982
  Std    : 0.0466208
  Median : 1

Mid stage:
  Mean   : 1
  Std    : 0
  Median : 1

Late stage:
  Mean   : 1
  Std    : 0
  Median : 1

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.0013914
  Significant  : Yes
  Effect size  : 0.212873
  Mean diff    : 0.00701754
  Median diff  : 0

MID-LATE:
```

```
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 1
  Significant  : No
  Effect size  : 0
  Mean diff    : 0
  Median diff  : 0

EARLY-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00119174
  Significant  : Yes
  Effect size  : 0.212873
  Mean diff    : 0.00701754
  Median diff  : 0

==================================================
Analysis for forward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.136599
  Std    : 0.000559334
  Median : 0.136826

Mid stage:
  Mean   : 0.128432
  Std    : 0.00478987
  Median : 0.129055

Late stage:
  Mean   : 0.118183
  Std    : 0.000587664
  Median : 0.117886

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.39375e-82
  Significant  : Yes
  Effect size  : -2.39501
  Mean diff    : -0.00816694
  Median diff  : -0.00777107

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.50099e-84
  Significant  : Yes
```

```
  Effect size  : -3.00352
  Mean diff    : -0.0102491
  Median diff  : -0.0111686

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.05232e-83
  Significant  : Yes
  Effect size  : -32.097
  Mean diff    : -0.018416
  Median diff  : -0.0189397

==================================================
Analysis for backward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.0920691
  Std    : 0.0091185
  Median : 0.0966924

Mid stage:
  Mean   : 0.0228588
  Std    : 0.0165806
  Median : 0.0144034

Late stage:
  Mean   : 0.0086946
  Std    : 0.000577371
  Median : 0.00869202

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.5014e-82
  Significant  : Yes
  Effect size  : -5.17256
  Mean diff    : -0.0692103
  Median diff  : -0.082289

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 4.52767e-70
  Significant  : Yes
  Effect size  : -1.20738
  Mean diff    : -0.0141642
  Median diff  : -0.00571134
```

```
EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.10488e-83
  Significant  : Yes
  Effect size  : -12.9049
  Mean diff    : -0.0833745
  Median diff  : -0.0880003
```

Listing 4: GFlowNet, Chain Length $n = 7$.

```
==================================================
Analysis for trajectory_balance_loss
==================================================

Summary Statistics:

Early stage:
  Mean   : 7.27744
  Std    : 2.84408
  Median : 8.48092

Mid stage:
  Mean   : 1.74531
  Std    : 1.98533
  Median : 0.719583

Late stage:
  Mean   : 0.00267701
  Std    : 0.00513376
  Median : 0.000157041

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 4.59541e-47
  Significant  : Yes
  Effect size  : -2.25563
  Mean diff    : -5.53212
  Median diff  : -7.76134

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 8.15457e-84
  Significant  : Yes
  Effect size  : -1.24133
  Mean diff    : -1.74264
  Median diff  : -0.719426
```

```
EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 9.29597e-48
  Significant  : Yes
  Effect size  : -3.61736
  Mean diff    : -7.27476
  Median diff  : -8.48077

==================================================
Analysis for terminal_reward
==================================================

Summary Statistics:

Early stage:
  Mean   : 34.081
  Std    : 8.16854
  Median : 33

Mid stage:
  Mean   : 41.0726
  Std    : 9.20453
  Median : 42

Late stage:
  Mean   : 50.0353
  Std    : 8.54646
  Median : 49

Statistical Tests:

EARLY-MID:
  Test used    : t-test
  P-value      : 8.03004e-18
  Significant  : Yes
  Effect size  : 0.803451
  Mean diff    : 6.99161
  Median diff  : 9

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 2.01281e-24
  Significant  : Yes
  Effect size  : 1.00913
  Mean diff    : 8.96271
  Median diff  : 7

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.2424e-57
```

```
  Significant  : Yes
  Effect size  : 1.90849
  Mean diff    : 15.9543
  Median diff  : 16

==================================================
Analysis for exploration_ratio
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.993288
  Std    : 0.0459779
  Median : 1

Mid stage:
  Mean   : 1
  Std    : 0
  Median : 1

Late stage:
  Mean   : 1
  Std    : 0
  Median : 1

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00139134
  Significant  : Yes
  Effect size  : 0.206455
  Mean diff    : 0.00671212
  Median diff  : 0

MID-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 1
  Significant  : No
  Effect size  : 0
  Mean diff    : 0
  Median diff  : 0

EARLY-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.0011917
  Significant  : Yes
  Effect size  : 0.206455
  Mean diff    : 0.00671212
```

```
  Median diff  : 0

==================================================
Analysis for forward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.109361
  Std    : 0.000378315
  Median : 0.109516

Mid stage:
  Mean   : 0.102774
  Std    : 0.00446085
  Median : 0.103877

Late stage:
  Mean   : 0.0905926
  Std    : 0.00113378
  Median : 0.0900322

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.39375e-82
  Significant  : Yes
  Effect size  : -2.08054
  Mean diff    : -0.00658623
  Median diff  : -0.00563877

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.50099e-84
  Significant  : Yes
  Effect size  : -3.74293
  Mean diff    : -0.0121817
  Median diff  : -0.0138448

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.05232e-83
  Significant  : Yes
  Effect size  : -22.2049
  Mean diff    : -0.018768
  Median diff  : -0.0194836

==================================================
```

```
Analysis for backward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.072528
  Std    : 0.00566449
  Median : 0.0752685

Mid stage:
  Mean   : 0.0205243
  Std    : 0.0146301
  Median : 0.0136152

Late stage:
  Mean   : 0.00679753
  Std    : 0.000457729
  Median : 0.00675402

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.53827e-82
  Significant  : Yes
  Effect size  : -4.6878
  Mean diff    : -0.0520037
  Median diff  : -0.0616532

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 2.35375e-77
  Significant  : Yes
  Effect size  : -1.32624
  Mean diff    : -0.0137268
  Median diff  : -0.00686123

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.10488e-83
  Significant  : Yes
  Effect size  : -16.3571
  Mean diff    : -0.0657304
  Median diff  : -0.0685144
```

Listing 5: GFlowNet, Chain Length $n = 9$.

```
==================================================
Analysis for trajectory_balance_loss
```

```
==================================================

Summary Statistics:

Early stage:
  Mean   : 7.43026
  Std    : 2.89578
  Median : 8.67855

Mid stage:
  Mean   : 1.72514
  Std    : 2.01813
  Median : 0.67939

Late stage:
  Mean   : 0.00124271
  Std    : 0.002363
  Median : 8.49505e-05

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 4.11994e-47
  Significant  : Yes
  Effect size  : -2.28586
  Mean diff    : -5.70512
  Median diff  : -7.99916

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.77444e-84
  Significant  : Yes
  Effect size  : -1.20803
  Mean diff    : -1.7239
  Median diff  : -0.679305

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 9.29597e-48
  Significant  : Yes
  Effect size  : -3.62812
  Mean diff    : -7.42902
  Median diff  : -8.67846

==================================================
Analysis for terminal_reward
==================================================

Summary Statistics:
```

```
Early stage:
  Mean   : 35.9231
  Std    : 8.40188
  Median : 36

Mid stage:
  Mean   : 43.5121
  Std    : 9.622
  Median : 43

Late stage:
  Mean   : 49.098
  Std    : 7.93491
  Median : 49

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 5.72986e-18
  Significant  : Yes
  Effect size  : 0.840184
  Mean diff    : 7.58902
  Median diff  : 7

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 8.90974e-12
  Significant  : Yes
  Effect size  : 0.633407
  Mean diff    : 5.58594
  Median diff  : 6

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.10968e-46
  Significant  : Yes
  Effect size  : 1.61226
  Mean diff    : 13.175
  Median diff  : 13

==================================================
Analysis for exploration_ratio
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.992501
```

```
  Std    : 0.0482797
  Median : 1

Mid stage:
  Mean   : 1
  Std    : 0
  Median : 1

Late stage:
  Mean   : 1
  Std    : 0
  Median : 1

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00245216
  Significant  : Yes
  Effect size  : 0.219652
  Mean diff    : 0.00749868
  Median diff  : 0

MID-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 1
  Significant  : No
  Effect size  : 0
  Mean diff    : 0
  Median diff  : 0

EARLY-LATE:
  Test used    : Mann-Whitney U (due to uniform data)
  P-value      : 0.00213111
  Significant  : Yes
  Effect size  : 0.219652
  Mean diff    : 0.00749868
  Median diff  : 0

==================================================
Analysis for forward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.0910028
  Std    : 0.000460817
  Median : 0.0911968
```

```
Mid stage:
  Mean   : 0.0850012
  Std    : 0.00345056
  Median : 0.0855755

Late stage:
  Mean   : 0.0769377
  Std    : 0.000574968
  Median : 0.0766502

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.39375e-82
  Significant  : Yes
  Effect size  : -2.4381
  Mean diff    : -0.00600162
  Median diff  : -0.00562132

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 7.50099e-84
  Significant  : Yes
  Effect size  : -3.25985
  Mean diff    : -0.00806349
  Median diff  : -0.00892528

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.05232e-83
  Significant  : Yes
  Effect size  : -26.9899
  Mean diff    : -0.0140651
  Median diff  : -0.0145466

==================================================
Analysis for backward_entropy
==================================================

Summary Statistics:

Early stage:
  Mean   : 0.0592377
  Std    : 0.00487484
  Median : 0.0616291

Mid stage:
  Mean   : 0.0163421
  Std    : 0.0117586
```

```
  Median : 0.0109835

Late stage:
  Mean   : 0.00542834
  Std    : 0.000253375
  Median : 0.00539537

Statistical Tests:

EARLY-MID:
  Test used    : Mann-Whitney U
  P-value      : 1.46542e-82
  Significant  : Yes
  Effect size  : -4.76574
  Mean diff    : -0.0428955
  Median diff  : -0.0506456

MID-LATE:
  Test used    : Mann-Whitney U
  P-value      : 3.26299e-78
  Significant  : Yes
  Effect size  : -1.3123
  Mean diff    : -0.0109138
  Median diff  : -0.00558811

EARLY-LATE:
  Test used    : Mann-Whitney U
  P-value      : 1.10488e-83
  Significant  : Yes
  Effect size  : -15.5892
  Mean diff    : -0.0538093
  Median diff  : -0.0562337
```

Listing 6: GFlowNet, Chain Length $n = 11$.

C.2 BEN

```
==================================================
Analysis for Q-Learning Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 2418.4
  Std      : 2948.42
  Median   : 1382.44

Mid stage:
  Mean     : 850.307
```

```
  Std      : 865.315
  Median   : 548.499

Late stage:
  Mean     : 249.975
  Std      : 267.485
  Median   : 164.514

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 2.75096e-12
  Significant   : Yes
  Effect size   : -0.721698
  Mean diff     : -1568.09
  Median diff   : -833.938

MID-LATE:
  Test used     : t-test
  P-value       : 3.35826e-19
  Significant   : Yes
  Effect size   : -0.937378
  Mean diff     : -600.332
  Median diff   : -383.985

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 2.16325e-39
  Significant   : Yes
  Effect size   : -1.03583
  Mean diff     : -2168.42
  Median diff   : -1217.92

==================================================
Analysis for Epistemic Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 3738.12
  Std      : 4340.37
  Median   : 3595.12

Mid stage:
  Mean     : 2224.3
  Std      : 926.447
  Median   : 1906.22
```

```
Late stage:
  Mean     : 1480.38
  Std      : 614.314
  Median   : 1450.01

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 1.87536e-39
  Significant   : Yes
  Effect size   : -0.482378
  Mean diff     : -1513.82
  Median diff   : -1688.89

MID-LATE:
  Test used     : t-test
  P-value       : 3.41227e-46
  Significant   : Yes
  Effect size   : -0.946429
  Mean diff     : -743.922
  Median diff   : -456.211

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 1.97483e-98
  Significant   : Yes
  Effect size   : -0.728375
  Mean diff     : -2257.74
  Median diff   : -2145.11

==================================================
Analysis for Rewards
==================================================

Summary Statistics:

Early stage:
  Mean     : -81.25
  Std      : 103.5
  Median   : -86

Mid stage:
  Mean     : -116.765
  Std      : 95.9329
  Median   : -99

Late stage:
  Mean     : -75.1765
  Std      : 90.9287
```

```
  Median   : -73

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 0.329132
  Significant   : No
  Effect size   : -0.355901
  Mean diff     : -35.5147
  Median diff   : -13

MID-LATE:
  Test used     : t-test
  P-value       : 0.217299
  Significant   : No
  Effect size   : 0.444964
  Mean diff     : 41.5882
  Median diff   : 26

EARLY-LATE:
  Test used     : t-test
  P-value       : 0.863135
  Significant   : No
  Effect size   : 0.0623456
  Mean diff     : 6.07353
  Median diff   : 13

==================================================
Analysis for Cumulative Returns
==================================================

Summary Statistics:

Early stage:
  Mean     : -572.312
  Std      : 353.307
  Median   : -648.5

Mid stage:
  Mean     : -2356.35
  Std      : 487.332
  Median   : -2300

Late stage:
  Mean     : -3809.12
  Std      : 401.997
  Median   : -3671

Statistical Tests:
```

```
EARLY-MID:
  Test used     : t-test
  P-value       : 8.10375e-13
  Significant   : Yes
  Effect size   : -4.19155
  Mean diff     : -1784.04
  Median diff   : -1651.5

MID-LATE:
  Test used     : t-test
  P-value       : 1.68029e-10
  Significant   : Yes
  Effect size   : -3.25217
  Mean diff     : -1452.76
  Median diff   : -1371

EARLY-LATE:
  Test used     : t-test
  P-value       : 1.87071e-21
  Significant   : Yes
  Effect size   : -8.55312
  Mean diff     : -3236.81
  Median diff   : -3022.5
```

Listing 7: BEN, Chain Length $n = 3$.

```
==================================================
Analysis for Q-Learning Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 9.0621e+09
  Std      : 1.11129e+11
  Median   : 359028

Mid stage:
  Mean     : 51210.2
  Std      : 82246.9
  Median   : 24168.2

Late stage:
  Mean     : 3529.9
  Std      : 5630.64
  Median   : 1565.3

Statistical Tests:
```

```
EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 1.61872e-55
  Significant   : Yes
  Effect size   : -0.115322
  Mean diff     : -9.06205e+09
  Median diff   : -334860

MID-LATE:
  Test used     : Mann-Whitney U
  P-value       : 1.27667e-59
  Significant   : Yes
  Effect size   : -0.817936
  Mean diff     : -47680.3
  Median diff   : -22602.9

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 1.30966e-94
  Significant   : Yes
  Effect size   : -0.115323
  Mean diff     : -9.0621e+09
  Median diff   : -357463

==================================================
Analysis for Epistemic Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 2214.01
  Std      : 1351.49
  Median   : 1759.53

Mid stage:
  Mean     : 915.499
  Std      : 561.958
  Median   : 724.881

Late stage:
  Mean     : 605.353
  Std      : 3316.38
  Median   : 288.135

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 1.72322e-180
```

```
  Significant   : Yes
  Effect size   : -1.25464
  Mean diff     : -1298.51
  Median diff   : -1034.65

MID-LATE:
  Test used     : Mann-Whitney U
  P-value       : 2.25913e-191
  Significant   : Yes
  Effect size   : -0.130398
  Mean diff     : -310.147
  Median diff   : -436.746

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.635261
  Mean diff     : -1608.66
  Median diff   : -1471.4

==================================================
Analysis for Rewards
==================================================

Summary Statistics:

Early stage:
  Mean     : -109.125
  Std      : 120.27
  Median   : -86

Mid stage:
  Mean     : -113.824
  Std      : 104.901
  Median   : -99

Late stage:
  Mean     : -96.8824
  Std      : 85.1993
  Median   : -105

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 0.908325
  Significant   : No
  Effect size   : -0.0416361
  Mean diff     : -4.69853
```

```
  Median diff   : -13

MID-LATE:
  Test used     : t-test
  P-value       : 0.619497
  Significant   : No
  Effect size   : 0.177284
  Mean diff     : 16.9412
  Median diff   : -6

EARLY-LATE:
  Test used     : t-test
  P-value       : 0.744724
  Significant   : No
  Effect size   : 0.117469
  Mean diff     : 12.2426
  Median diff   : -19

==================================================
Analysis for Cumulative Returns
==================================================

Summary Statistics:

Early stage:
  Mean     : -1084.31
  Std      : 438.654
  Median   : -1081.5

Mid stage:
  Mean     : -2960.94
  Std      : 548.241
  Median   : -3043

Late stage:
  Mean     : -4479.41
  Std      : 467.605
  Median   : -4589

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 1.02639e-11
  Significant   : Yes
  Effect size   : -3.77986
  Mean diff     : -1876.63
  Median diff   : -1961.5

MID-LATE:
```

```
  Test used     : t-test
  P-value       : 1.24336e-09
  Significant   : Yes
  Effect size   : -2.98019
  Mean diff     : -1518.47
  Median diff   : -1546

EARLY-LATE:
  Test used     : t-test
  P-value       : 8.83529e-20
  Significant   : Yes
  Effect size   : -7.48874
  Mean diff     : -3395.1
  Median diff   : -3507.5
```

Listing 8: BEN, Chain Length $n = 5$.

```
==================================================
Analysis for Q-Learning Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 2085.95
  Std      : 2162.4
  Median   : 1373.73

Mid stage:
  Mean     : 576.122
  Std      : 649.869
  Median   : 338.179

Late stage:
  Mean     : 156.021
  Std      : 191.834
  Median   : 67.8597

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 1.61151e-36
  Significant   : Yes
  Effect size   : -0.94565
  Mean diff     : -1509.83
  Median diff   : -1035.55

MID-LATE:
  Test used     : t-test
```

```
  P-value       : 7.51887e-33
  Significant   : Yes
  Effect size   : -0.876801
  Mean diff     : -420.101
  Median diff   : -270.319

EARLY-LATE:
  Test used     : t-test
  P-value       : 1.0056e-60
  Significant   : Yes
  Effect size   : -1.25724
  Mean diff     : -1929.93
  Median diff   : -1305.87

==================================================
Analysis for Epistemic Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 1376.98
  Std      : 1241.08
  Median   : 988.406

Mid stage:
  Mean     : 1258.87
  Std      : 31145.6
  Median   : 161.888

Late stage:
  Mean     : 2677.52
  Std      : 60170.7
  Median   : 5.31501

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.00535891
  Mean diff     : -118.114
  Median diff   : -826.518

MID-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : 0.0296113
```

```
  Mean diff     : 1418.65
  Median diff   : -156.573

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : 0.0305605
  Mean diff     : 1300.54
  Median diff   : -983.091

==================================================
Analysis for Rewards
==================================================

Summary Statistics:

Early stage:
  Mean     : -80.875
  Std      : 70.1827
  Median   : -57.5

Mid stage:
  Mean     : -114.882
  Std      : 97.6795
  Median   : -137

Late stage:
  Mean     : -79.6471
  Std      : 75.2853
  Median   : -86

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 0.276686
  Significant   : No
  Effect size   : -0.399853
  Mean diff     : -34.0074
  Median diff   : -79.5

MID-LATE:
  Test used     : t-test
  P-value       : 0.261584
  Significant   : No
  Effect size   : 0.404055
  Mean diff     : 35.2353
  Median diff   : 51
```

```
EARLY-LATE:
  Test used     : t-test
  P-value       : 0.962895
  Significant   : No
  Effect size   : 0.0168723
  Mean diff     : 1.22794
  Median diff   : -28.5

==================================================
Analysis for Cumulative Returns
==================================================

Summary Statistics:

Early stage:
  Mean     : -756.438
  Std      : 448.425
  Median   : -829.5

Mid stage:
  Mean     : -2396.41
  Std      : 540.392
  Median   : -2349

Late stage:
  Mean     : -4111.24
  Std      : 491.61
  Median   : -4231

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 2.4656e-10
  Significant   : Yes
  Effect size   : -3.30279
  Mean diff     : -1639.97
  Median diff   : -1519.5

MID-LATE:
  Test used     : t-test
  P-value       : 1.03534e-10
  Significant   : Yes
  Effect size   : -3.31959
  Mean diff     : -1714.82
  Median diff   : -1882

EARLY-LATE:
  Test used     : t-test
  P-value       : 3.69339e-19
```

```
  Significant   : Yes
  Effect size   : -7.13008
  Mean diff     : -3354.8
  Median diff   : -3401.5
```
Listing 9: BEN, Chain Length $n = 7$.

```
==================================================
Analysis for Q-Learning Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 201.363
  Std      : 646.288
  Median   : 28.6683

Mid stage:
  Mean     : 7.60222
  Std      : 6.14727
  Median   : 6.00191

Late stage:
  Mean     : 3.53114
  Std      : 2.27404
  Median   : 2.88227

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 3.91253e-87
  Significant   : Yes
  Effect size   : -0.423969
  Mean diff     : -193.76
  Median diff   : -22.6664

MID-LATE:
  Test used     : t-test
  P-value       : 9.57251e-41
  Significant   : Yes
  Effect size   : -0.878398
  Mean diff     : -4.07108
  Median diff   : -3.11963

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 7.90234e-134
  Significant   : Yes
```

```
  Effect size   : -0.432894
  Mean diff     : -197.831
  Median diff   : -25.7861

==================================================
Analysis for Epistemic Loss
==================================================

Summary Statistics:

Early stage:
  Mean     : 854.156
  Std      : 1244.76
  Median   : 476.503

Mid stage:
  Mean     : 2051.78
  Std      : 109092
  Median   : 9.81774

Late stage:
  Mean     : -21.3259
  Std      : 417.914
  Median   : -29.5177

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : 0.0155244
  Mean diff     : 1197.62
  Median diff   : -466.685

MID-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.0268745
  Mean diff     : -2073.1
  Median diff   : -39.3355

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.942936
  Mean diff     : -875.482
  Median diff   : -506.021
```

```
==================================================
Analysis for Rewards
==================================================

Summary Statistics:

Early stage:
  Mean     : -139.375
  Std      : 107.425
  Median   : -111.5

Mid stage:
  Mean     : -131
  Std      : 83.4245
  Median   : -118

Late stage:
  Mean     : -128.353
  Std      : 74.1294
  Median   : -137

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 0.809435
  Significant   : No
  Effect size   : 0.0870797
  Mean diff     : 8.375
  Median diff   : -6.5

MID-LATE:
  Test used     : t-test
  P-value       : 0.925005
  Significant   : No
  Effect size   : 0.0335436
  Mean diff     : 2.64706
  Median diff   : -19

EARLY-LATE:
  Test used     : t-test
  P-value       : 0.740539
  Significant   : No
  Effect size   : 0.119427
  Mean diff     : 11.0221
  Median diff   : -25.5

==================================================
Analysis for Cumulative Returns
```

```
==================================================

Summary Statistics:

Early stage:
  Mean     : -1496.19
  Std      : 584.326
  Median   : -1587.5

Mid stage:
  Mean     : -3164.82
  Std      : 677.043
  Median   : -2960

Late stage:
  Mean     : -5642.65
  Std      : 604.554
  Median   : -5607

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 3.02444e-08
  Significant   : Yes
  Effect size   : -2.63863
  Mean diff     : -1668.64
  Median diff   : -1372.5

MID-LATE:
  Test used     : t-test
  P-value       : 2.52487e-12
  Significant   : Yes
  Effect size   : -3.8606
  Mean diff     : -2477.82
  Median diff   : -2647

EARLY-LATE:
  Test used     : t-test
  P-value       : 6.79593e-19
  Significant   : Yes
  Effect size   : -6.97439
  Mean diff     : -4146.46
  Median diff   : -4019.5
```

Listing 10: BEN, Chain Length $n = 9$.

```
==================================================
Analysis for Q-Learning Loss
==================================================
```

```
Summary Statistics:

Early stage:
  Mean     : 150.151
  Std      : 610.171
  Median   : 2.32373

Mid stage:
  Mean     : 1.27351
  Std      : 0.58875
  Median   : 1.13778

Late stage:
  Mean     : 1.09848
  Std      : 0.48181
  Median   : 1.00344

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 7.15432e-75
  Significant   : Yes
  Effect size   : -0.345059
  Mean diff     : -148.878
  Median diff   : -1.18596

MID-LATE:
  Test used     : t-test
  P-value       : 1.97005e-08
  Significant   : Yes
  Effect size   : -0.325374
  Mean diff     : -0.175033
  Median diff   : -0.134332

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 4.34471e-101
  Significant   : Yes
  Effect size   : -0.345464
  Mean diff     : -149.053
  Median diff   : -1.32029

==================================================
Analysis for Epistemic Loss
==================================================

Summary Statistics:
```

```
Early stage:
  Mean     : 9005.47
  Std      : 424831
  Median   : 230.341

Mid stage:
  Mean     : 1946.32
  Std      : 53466.3
  Median   : -11.3205

Late stage:
  Mean     : 295.921
  Std      : 9372.83
  Median   : -38.3282

Statistical Tests:

EARLY-MID:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.0233152
  Mean diff     : -7059.15
  Median diff   : -241.661

MID-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.0429982
  Mean diff     : -1650.4
  Median diff   : -27.0076

EARLY-LATE:
  Test used     : Mann-Whitney U
  P-value       : 0
  Significant   : Yes
  Effect size   : -0.028986
  Mean diff     : -8709.55
  Median diff   : -268.669

==================================================
Analysis for Rewards
==================================================

Summary Statistics:

Early stage:
  Mean     : -82.875
  Std      : 76.4566
```

```
  Median   : -86

Mid stage:
  Mean     : -144.471
  Std      : 85.5083
  Median   : -137

Late stage:
  Mean     : -75.6471
  Std      : 84.6633
  Median   : -67

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 0.0430657
  Significant   : Yes
  Effect size   : -0.759419
  Mean diff     : -61.5956
  Median diff   : -51

MID-LATE:
  Test used     : t-test
  P-value       : 0.0289021
  Significant   : Yes
  Effect size   : 0.808862
  Mean diff     : 68.8235
  Median diff   : 70

EARLY-LATE:
  Test used     : t-test
  P-value       : 0.80504
  Significant   : No
  Effect size   : 0.0896051
  Mean diff     : 7.22794
  Median diff   : 19

==================================================
Analysis for Cumulative Returns
==================================================

Summary Statistics:

Early stage:
  Mean     : -598.438
  Std      : 432.208
  Median   : -680.5

Mid stage:
```

```
  Mean     : -2539.53
  Std      : 620.485
  Median   : -2400

Late stage:
  Mean     : -4302.35
  Std      : 429.992
  Median   : -4042

Statistical Tests:

EARLY-MID:
  Test used     : t-test
  P-value       : 2.8519e-11
  Significant   : Yes
  Effect size   : -3.63025
  Mean diff     : -1941.09
  Median diff   : -1719.5

MID-LATE:
  Test used     : t-test
  P-value       : 1.17108e-10
  Significant   : Yes
  Effect size   : -3.30237
  Mean diff     : -1762.82
  Median diff   : -1642

EARLY-LATE:
  Test used     : t-test
  P-value       : 1.54372e-21
  Significant   : Yes
  Effect size   : -8.59174
  Mean diff     : -3703.92
  Median diff   : -3361.5
```

Listing 11: BEN, Chain Length $n = 11$.