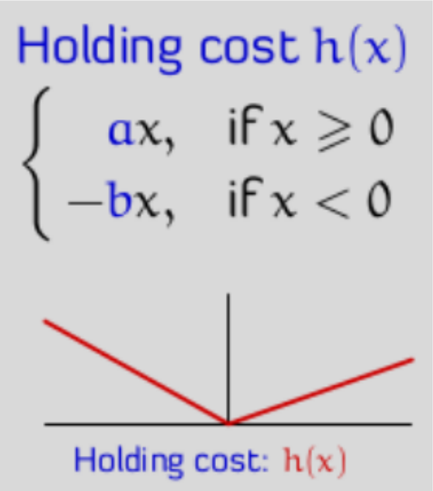
RL FINAL PROJECT PRESENTATION

YASH ADIVAREKAR - 2021101008
YASH KAWADE - 2021101032
MADHAV TANK - 2021101108
MANAV SHAH - 2021101090
NIKUNJ GARG - 2021101021

INVENTORY MANAGEMENT PROBLEM

- Retail stores stockpile products in warehouses to meet the random demand. Additional stocks are procured at regular intervals.
- Let Xt denote the amount of stock before the t-th procurement.
- At time t, the store may procure an additional stock Ut (≤ U) units for price p per unit. Thus the total procurement cost is pUt.

- The random demand Wt is i.i.d. with distribution Pw . Pw is uniform[0,10].
- The stock available at the next time is X(t+1) = Xt + Ut - Wt, where a negative stock denotes backlogged demand.
- Per-stage cost is c(Xt+1, Ut) = h(Xt+1) + pUt. The holding cost for the stock is given by h(x) where a is the per-unit storage cost and b is the per-unit backlog cost.
- We want to find the optimal inventory control strategy to minimize the expected total cost over a finite horizon.



State Space

Our states are X's (the amount of inventory). Our state's range is X's range.

$$-10 <= X <= 10$$

Actions

The amount of inventory produced in that day is taken as the action. U(t) is our action on t^{th} day.

$$0 <= U <= 10$$

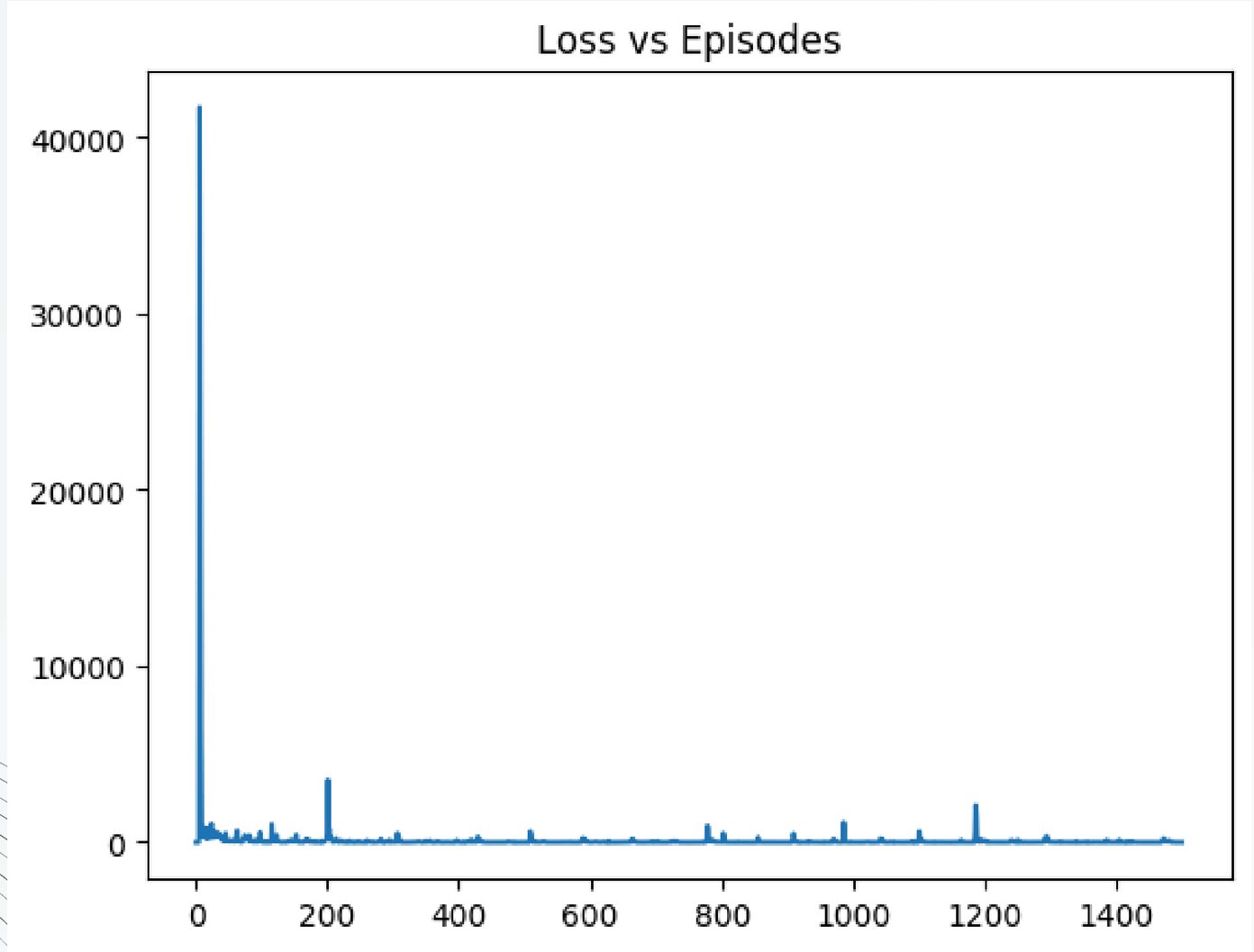
DEEP Q-NETWORK (DQN)

Pseudo code

Algorithm 1 Deep Q-learning with Experience Replay

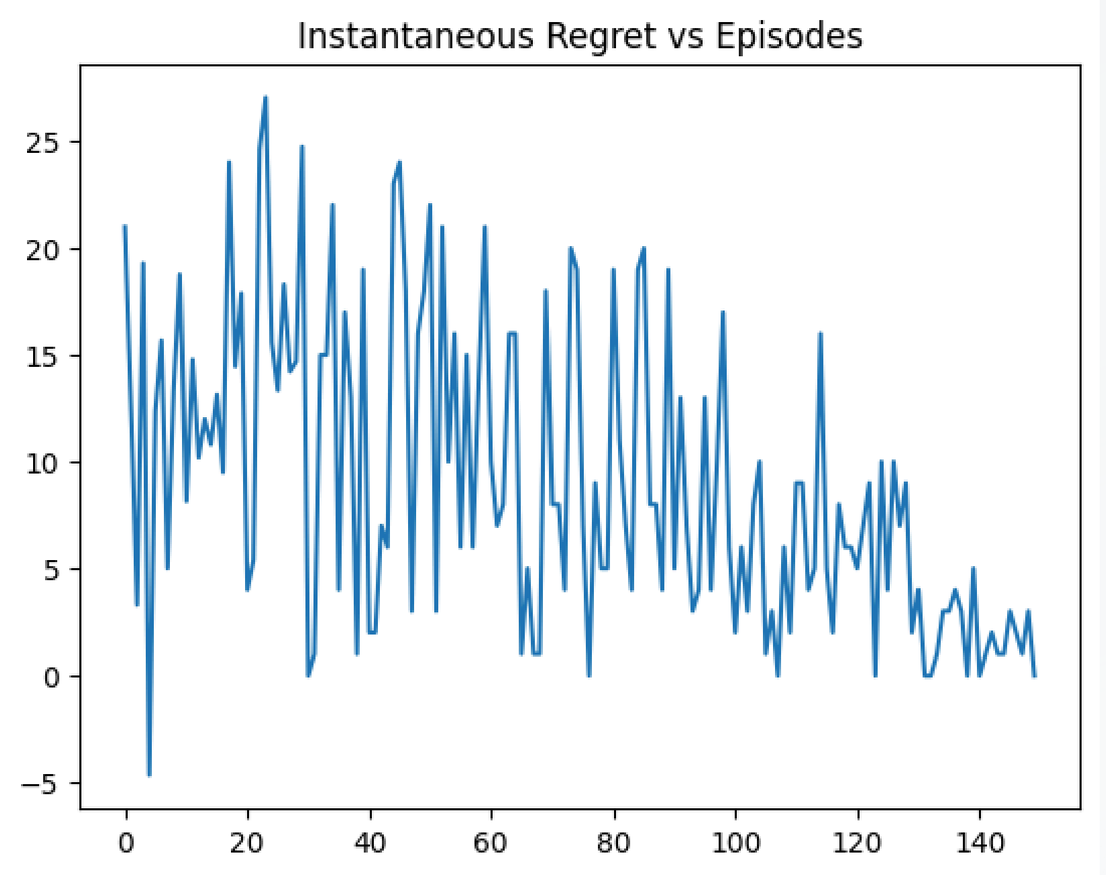
Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$
  end for
end for

DQN Loss plot



Cumulative Regret

Instantaneous Regret



Experimenting with various parameters

1

The DQN algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 3**
- **Backlog Cost (b): 3**
- **Purchase Cost per Unit (p): 0**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	3	2	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0
3	3	2	1	0	0	0	0	0	0	0	0
4	2	1	0	0	0	0	0	0	0	0	0

2

The DQN algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 50**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 15**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	3	2	1	0	0	0	0	0	0	0	0
1	5	5	4	3	2	1	0	0	0	0	0
2	5	5	4	3	2	1	0	0	0	0	0
3	5	4	3	2	1	0	0	0	0	0	0
4	4	3	2	1	0	0	0	0	0	0	0

3

The DQN algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 2**
- **Backlog Cost (b): 2**
- **Purchase Cost per Unit (p): 10**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	3	2	1	0	0	0	0	0	0	0	0
1	3	2	1	0	0	0	0	0	0	0	0
2	4	3	2	1	0	0	0	0	0	0	0
3	5	4	3	2	1	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0

4

The DQN algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 5**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 5**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	4	3	2	1	0	0	0
1	5	5	5	5	5	4	3	2	1	0	0
2	5	5	5	5	5	5	4	3	2	1	0
3	5	5	5	5	5	4	3	2	1	0	0
4	5	5	5	5	5	4	3	2	1	0	0

POLICY GRADIENT METHODS

THE REINFORCE ALGORITHM (MONTE CARLO REINFORCE)

Pseudo code

REINFORCE (Monte-Carlo policy gradient) relies on an estimated return by Monte-Carlo methods using episode samples to update the policy parameter θ. REINFORCE works because the expectation of the sample gradient is equal to the actual gradient:

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_\pi[Q^\pi(s,a)\nabla_\theta \ln \pi_\theta(a|s)] \\
&= \mathbb{E}_\pi[G_t \nabla_\theta \ln \pi_\theta(A_t|S_t)] \qquad ; \text{Because } Q^\pi(S_t,A_t)=\mathbb{E}_\pi[G_t|S_t,A_t]
\end{aligned}
$$

Therefore we are able to measure G_t from real sample trajectories and use that to update our policy gradient. It relies on a full trajectory and that's why it is a Monte-Carlo method.

The process is pretty straightforward:

1. Initialize the policy parameter θ at random.
2. Generate one trajectory on policy π_θ: $S_1, A_1, R_2, S_2, A_2, \ldots, S_T$.
3. For t=1, 2, … , T:
    1. Estimate the the return G_t;
    2. Update policy parameters: $\theta \leftarrow \theta + \alpha\gamma^t G_t \nabla_\theta \ln \pi_\theta(A_t|S_t)$

The Reinforce algorithm's policy for the given parameters:

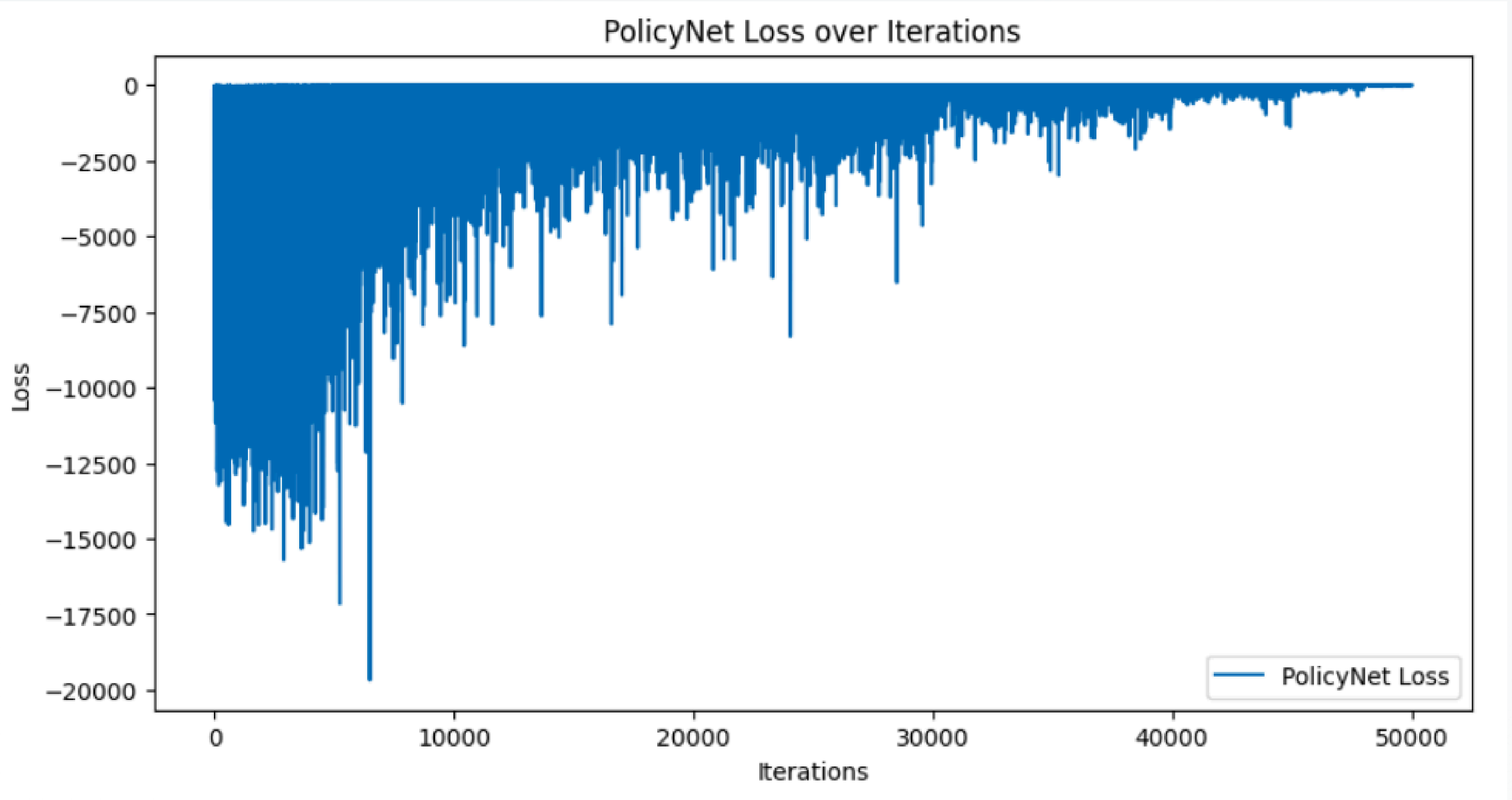
- **Horizon (T): 5**
- **Holding Cost (a): 2**
- **Backlog Cost (b): 50**
- **Purchase Cost per Unit (p): 15**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

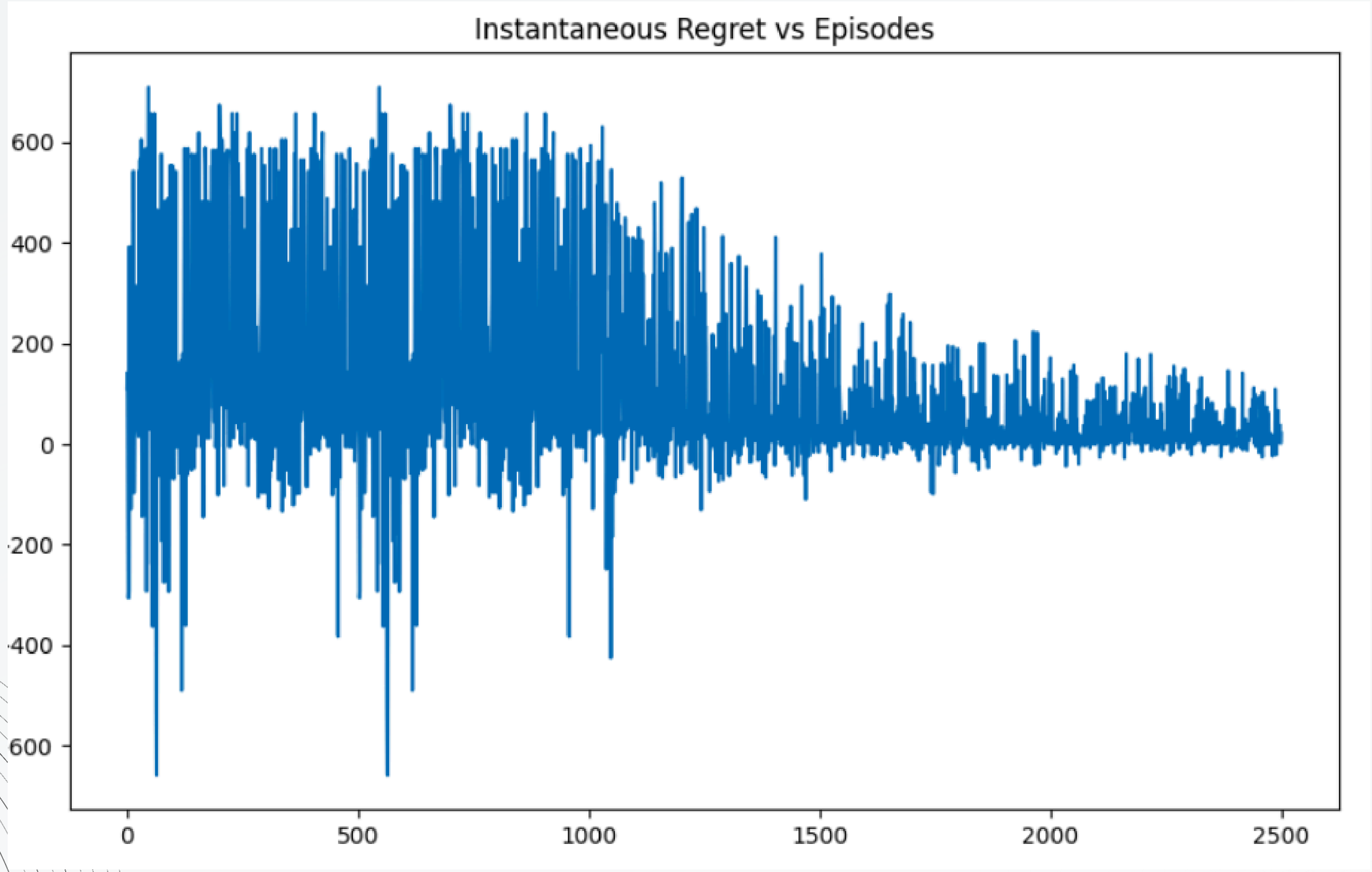
Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	5	5	5	5	5	4	3
1	5	5	5	5	5	4	5	5	5	5	5
2	5	5	5	5	5	5	5	5	4	3	1
3	5	5	5	5	5	5	5	4	3	2	2
4	5	5	5	5	4	3	2	2	1	0	0

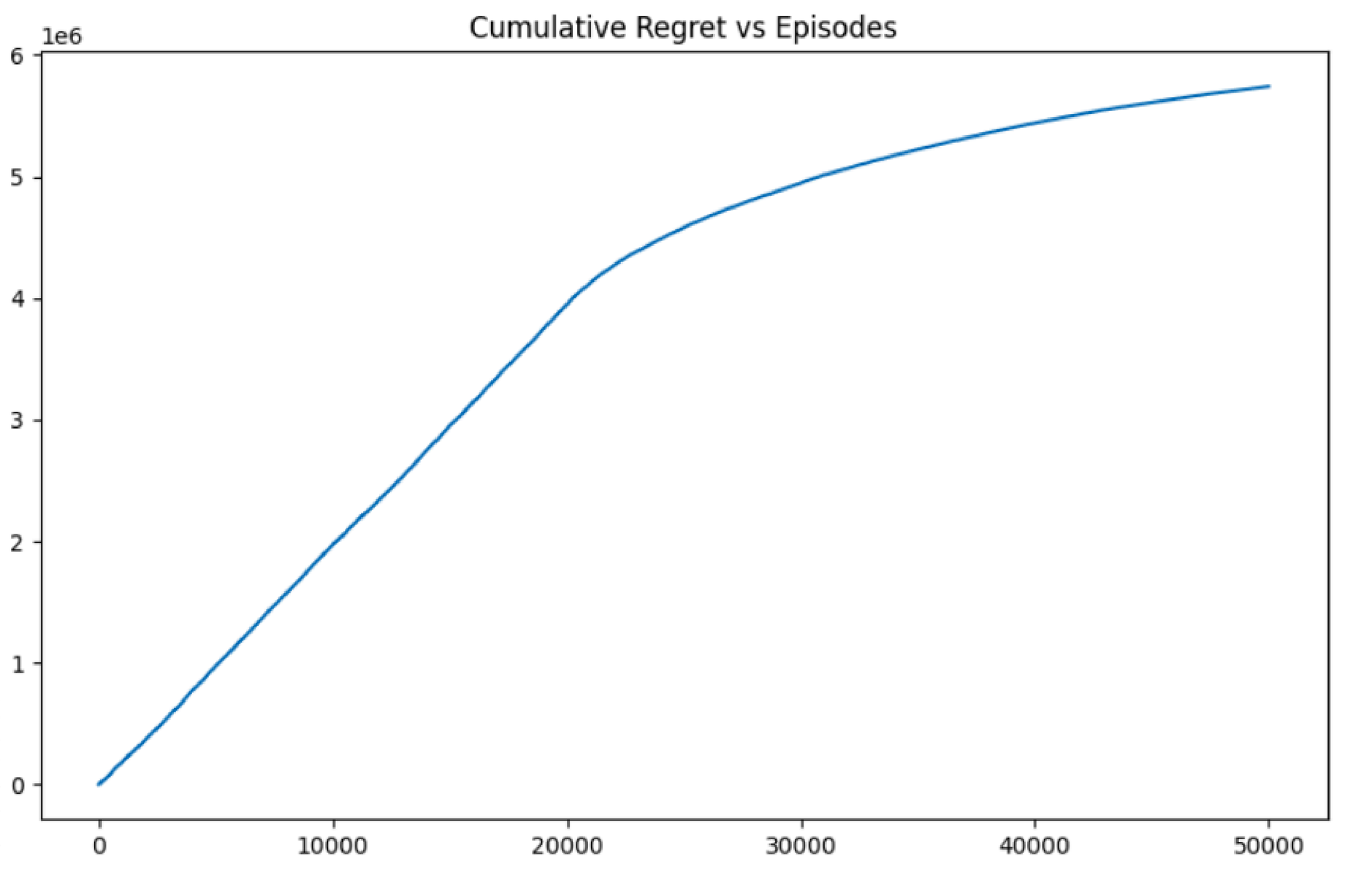
Policy network loss plot



Instantaneous Regret vs Episodes



Cumulative Regret vs Episodes



Experimenting with various parameters

1

The Reinforce algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 3**
- **Backlog Cost (b): 3**
- **Purchase Cost per Unit (p): 0**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	3	2	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0
3	3	2	1	0	0	0	0	0	0	0	0
4	2	1	0	0	0	0	0	0	0	0	0

2

The Reinforce algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 50**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 15**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	4	3	2	1	0	0	0	0	0	0	0
1	5	4	3	2	1	0	0	0	0	0	0
2	4	3	2	1	0	0	0	0	0	0	0
3	5	5	5	5	4	3	2	1	0	0	0
4	5	5	5	5	4	3	2	1	0	0	0
5	4	3	2	1	0	0	0	0	0	0	0

3

The Reinforce algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 2**
- **Backlog Cost (b): 2**
- **Purchase Cost per Unit (p): 10**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	4	3	2	1	0	0	0
1	5	4	3	2	1	0	0	0	0	0	0
2	3	2	1	0	0	0	0	0	0	0	0
3	4	3	2	1	0	0	0	0	0	0	0
4	4	3	2	1	0	0	0	0	0	0	0

4

The Reinforce algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 5**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 5**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	4	3	2	1	0	0	0
1	5	5	5	5	4	3	2	1	0	0	0
2	5	5	5	5	4	3	2	1	0	0	0
3	5	4	3	2	1	0	0	0	0	0	0
4	3	2	1	0	0	0	0	0	0	0	0

ACTOR CRITIC

Pseudo code

Pseudocode of Actor-Critic algorithm[6]

1. Sample {s_t, a_t}using the policy πθ from the actor-network.
2. Evaluate the advantage function A_t. It can be called as TD error δt. In Actor-critic algorithm, advantage function is produced by the critic-network.

$$A_{\pi_\theta}(s_t, a_t) = r(s_t, a_t) + V_{\pi_\theta}(s_{t+1}) - V_{\pi_\theta}(s_t)$$

3. Evaluate the gradient using the below expression:

$$\nabla J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t, s_t) A_{\pi_\theta}(s_t, a_t)$$

4. Update the policy parameters, θ

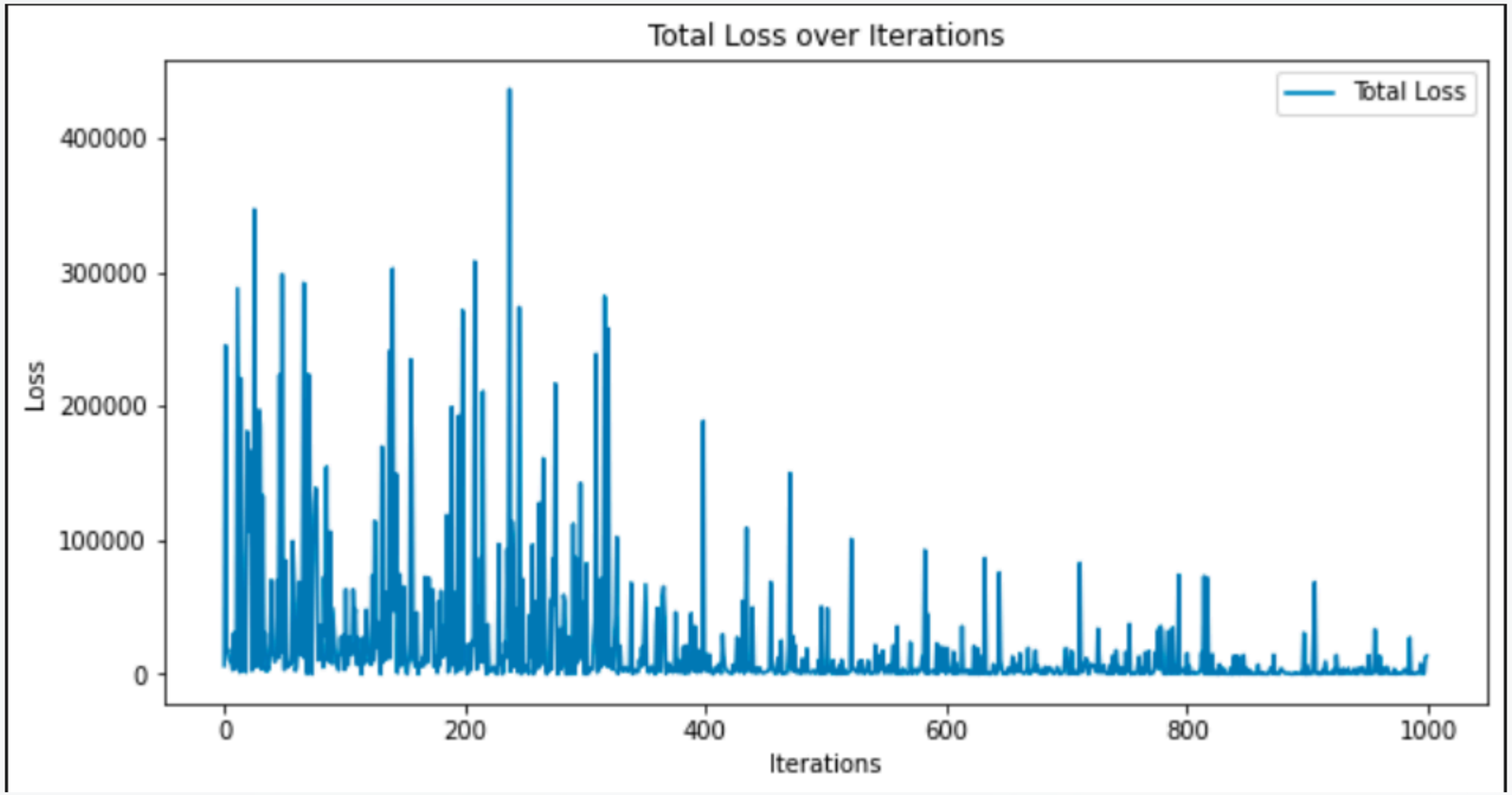
$$\theta = \theta + \alpha \nabla J(\theta)$$

5. Update the weights of the critic based value-based RL(Q-learning). δt is equivalent to advantage function.

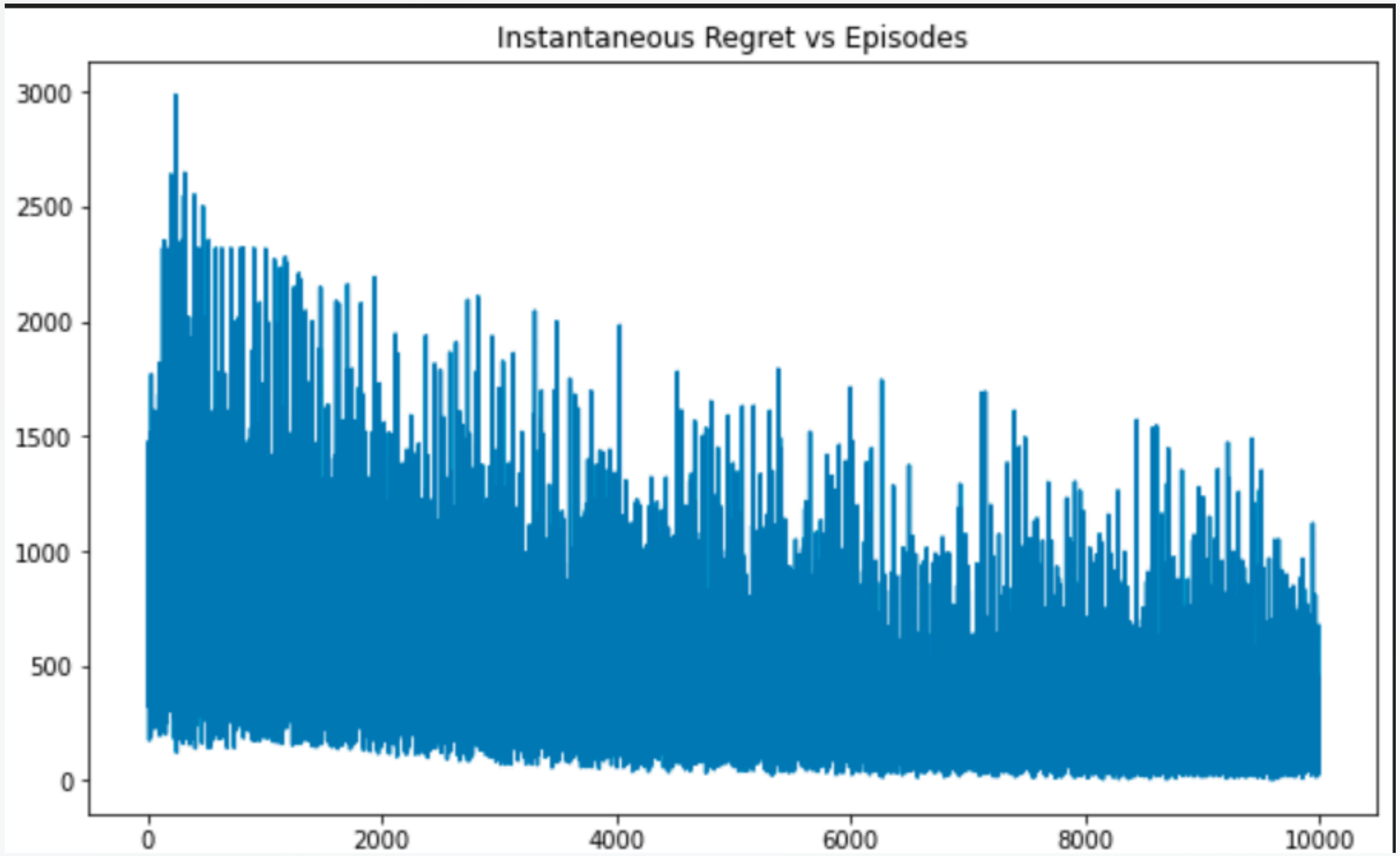
$$w = w + \alpha \delta_t$$

6. Repeat 1 to 5 until we find the optimal policy πθ.

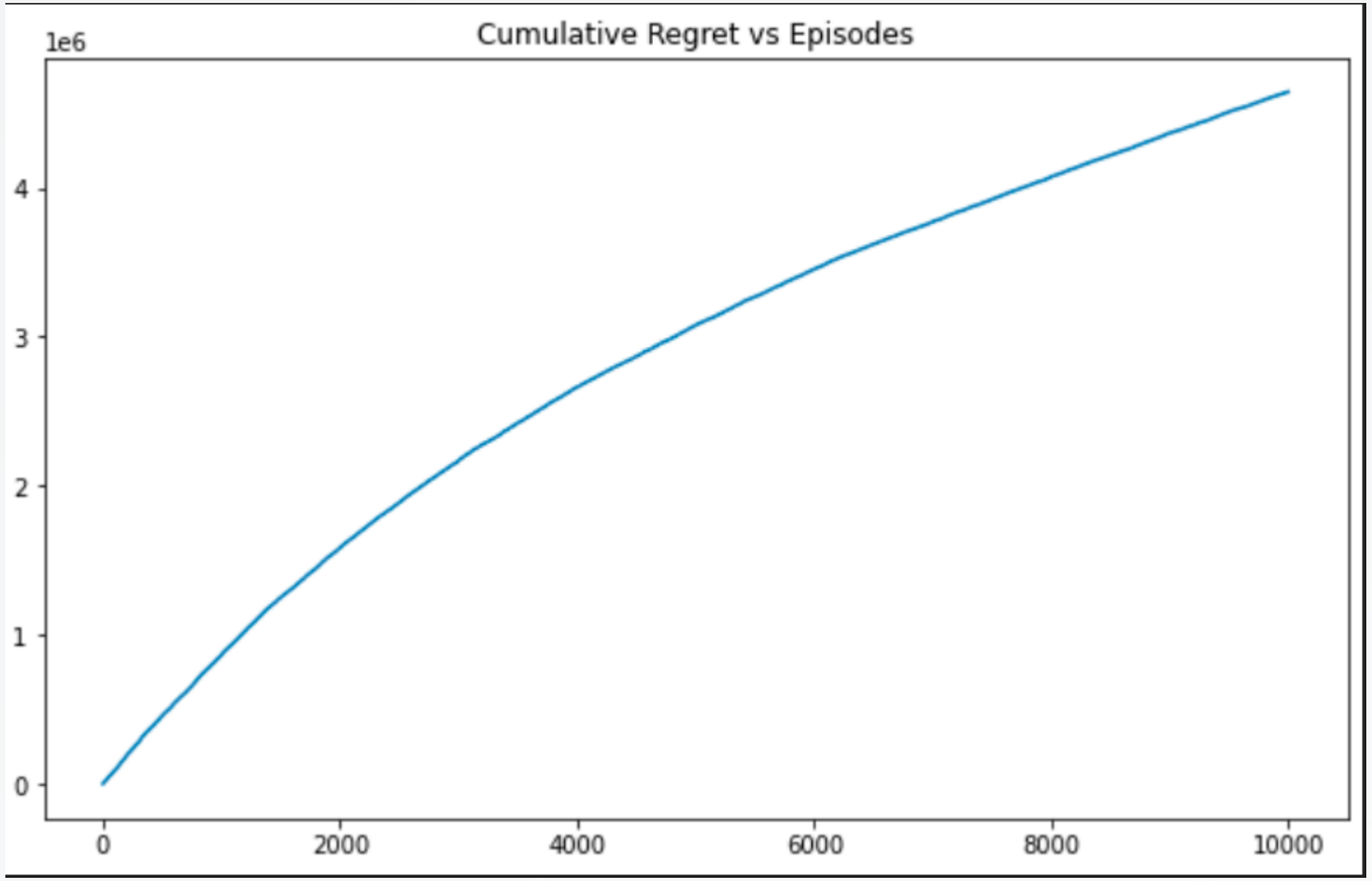
Total Loss Plot



Instantaneous Regret vs Episodes



Cumulative Regret vs Episodes



Experimenting with various parameters

1

The Actor Critic algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 3**
- **Backlog Cost (b): 3**
- **Purchase Cost per Unit (p): 0**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	1	0	0	0	0	0	0	0	0	0	0
1	4	3	2	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0	0	0	0
4	4	3	2	1	0	0	0	0	0	0	0
5	4	3	2	1	0	0	0	0	0	0	0

2

The Actor Critic algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 50**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 15**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	2	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	2	1	0	0	0	0	0	0	0	0	0
3	4	3	2	1	0	0	0	0	0	0	0
4	5	4	3	2	1	0	0	0	0	0	0
5	2	1	0	0	0	0	0	0	0	0	0

3

The Actor Critic algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 2**
- **Backlog Cost (b): 2**
- **Purchase Cost per Unit (p): 10**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	4	3	2	1	0	0	0	0
1	5	5	4	3	2	1	0	0	0	0	0
2	3	2	1	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0
4	3	2	1	0	0	0	0	0	0	0	0

4

The Actor Critic algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 5**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 5**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 4

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	4	3	2	1	0	0	0
1	5	5	5	5	4	3	2	1	0	0	0
2	5	5	5	5	5	4	3	2	1	0	0
3	5	5	5	5	5	4	3	2	1	0	0
4	5	5	5	5	4	3	2	1	0	0	0

NATURAL ACTOR CRITIC

NATURAL GRADIENT WITH FISCHER PROJECTION MATRIX

Pseudo code

Input: Parameterized policy $\pi(\boldsymbol{u}|\boldsymbol{x}) = p(\boldsymbol{u}|\boldsymbol{x}, \boldsymbol{\theta})$ with initial parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, its derivative $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \pi(\boldsymbol{u}|\boldsymbol{x})$.

For $u = 1, 2, 3, \ldots$ **do**
  For $e = 1, 2, 3, \ldots$ **do**
    Execute Rollout: Draw initial state $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$.
    For $t = 1, 2, 3, \ldots, N$ **do**
      Draw action $\boldsymbol{u}_t \sim \pi(\boldsymbol{u}_t|\boldsymbol{x}_t)$, observe next state $\boldsymbol{x}_{t+1} \sim p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)$, and reward $r_t = r(\boldsymbol{x}_t, \boldsymbol{u}_t)$.
    end.
  end.
  Critic Evaluation (Episodic): Determine value function $J = V^{\pi}(\boldsymbol{x}_0)$, compatible function approximation $f_{\boldsymbol{w}}^{\pi}(\boldsymbol{x}_t, \boldsymbol{u}_t)$.
  Update: Determine basis functions: $\boldsymbol{\phi}_t = \left[\sum_{t=0}^{N} \gamma^t \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \pi(\boldsymbol{u}_t|\boldsymbol{x}_t)^T, 1\right]^T$; reward statistics: $R_t = \sum_{t=0}^{N} \gamma^t r$;
  Actor-Update: When the natural gradient is converged, $\measuredangle(\boldsymbol{w}_{t+1}, \boldsymbol{w}_{t-\tau}) \leq \epsilon$, update the policy parameters: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \boldsymbol{w}_{t+1}$.
6: **end.**

The Natural Actor critic policy for the given parameters:

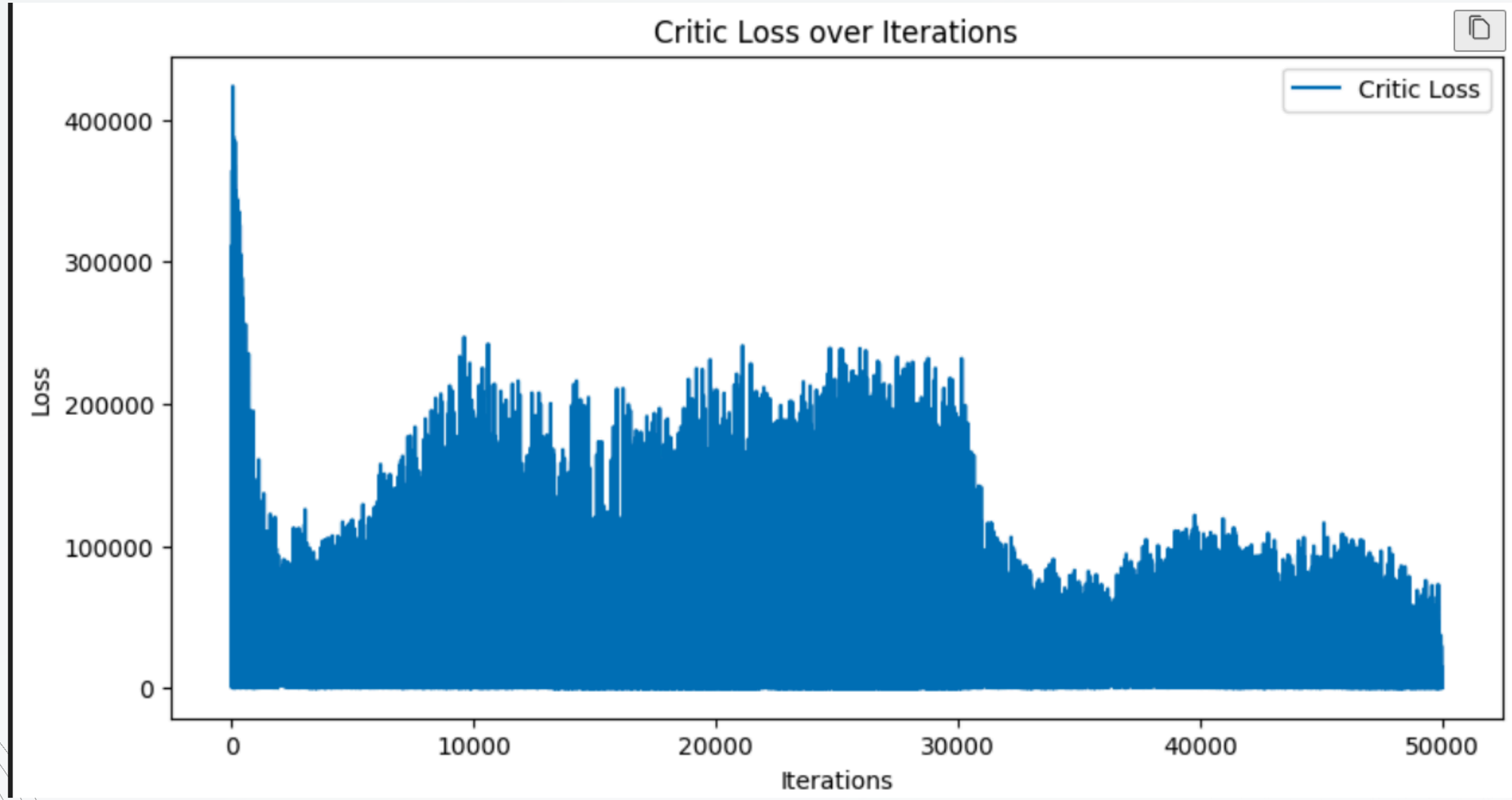
- **Horizon (T): 5**
- **Holding Cost (a): 2**
- **Backlog Cost (b): 50**
- **Purchase Cost per Unit (p): 15**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

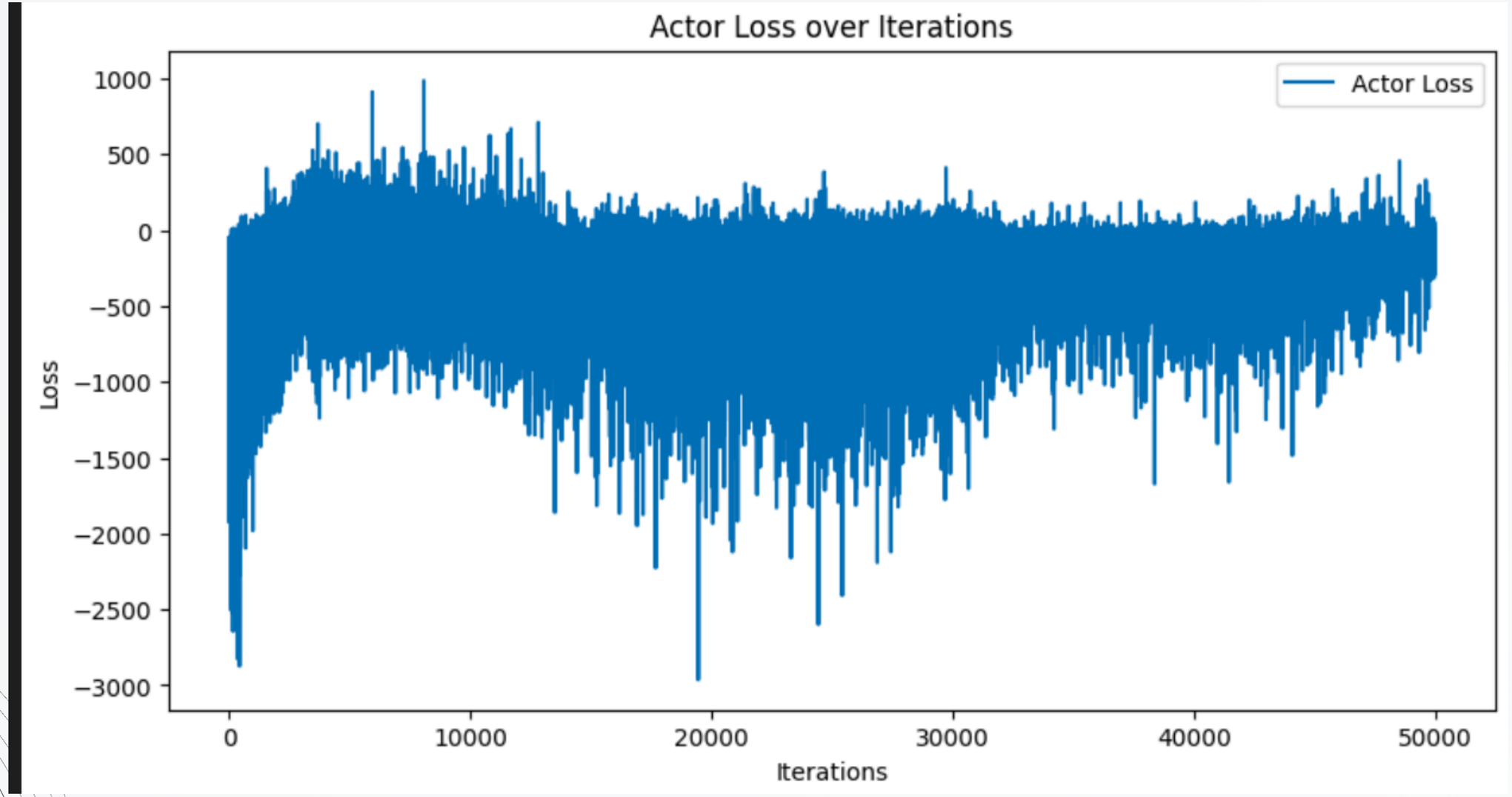
Time from 0 to 5

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	5	5	5	5	5	5	4
1	5	5	5	5	5	5	4	3	2	1	0
2	4	5	5	4	5	5	5	5	5	4	3
3	5	5	5	5	5	5	5	4	3	2	1
4	5	5	5	5	4	3	3	1	0	0	0
5	5	5	4	3	2	1	0	0	0	0	0

Critic loss plot



Actor network loss plot



Experimenting with various parameters

1

The Natural Actor-Critic policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 3**
- **Backlog Cost (b): 3**
- **Purchase Cost per Unit (p): 0**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 5

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0
4	2	1	0	0	0	0	0	0	0	0	0
5	2	1	0	0	0	0	0	0	0	0	0

2

The Natural Actor-Critic policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 50**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 15**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 5

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	4	3	2	1	0	0	0	0
1	5	5	4	3	2	1	0	0	0	0	0
2	5	5	4	3	2	1	0	0	0	0	0
3	5	5	5	4	3	2	1	0	0	0	0
4	5	5	4	3	2	1	0	0	0	0	0
5	2	1	0	0	0	0	0	0	0	0	0

3

The Natural Actor Critic policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 2**
- **Backlog Cost (b): 2**
- **Purchase Cost per Unit (p): 10**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 5

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	4	3	2	1	0	0	0	0	0
1	5	4	3	2	1	0	0	0	0	0	0
2	5	5	5	4	3	2	1	0	0	0	0
3	4	3	2	1	0	0	0	0	0	0	0
4	3	2	1	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0

4

The Natural Actor-Critic algorithm's policy for the given parameters:

- **Horizon (T): 5**
- **Holding Cost (a): 5**
- **Backlog Cost (b): 5**
- **Purchase Cost per Unit (p): 5**
- **State Space (S): -5 to 5**
- **Action Space: 0 to 5**
- **Demand Range: 0 to 5**

States from -5 to 5

Time from 0 to 5

	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	5	5	5	5	4	3	2	1	0	0	0
1	5	4	3	2	1	0	0	0	0	0	0
2	5	5	5	5	4	3	2	1	0	0	0
3	5	5	5	5	4	3	2	1	0	0	0
4	5	5	5	4	3	2	1	0	0	0	0
5	5	5	4	3	2	1	0	0	0	0	0

THANK YOU