Networked Restless Multi-Arm Bandits with Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Restless Multi-Armed Bandits (RMABs) are a powerful framework for sequential decision-making, widely applied in resource allocation and intervention optimization challenges in public health. However, traditional RMABs assume independence among arms, limiting their ability to account for interactions between individuals that can be common and significant in a real-world environment. This paper introduces Networked RMAB, a novel framework that integrates the RMAB model with the independent cascade model to capture interactions between arms in networked environments. We define the Bellman equation for networked RMAB and present its computational challenge due to exponentially large action and state spaces. To resolve the computational challenge, we establish the submodularity of Bellman equation and apply the hill-climbing algorithm to achieve a $1 - \frac{1}{e}$ approximation guarantee in Bellman updates. Lastly, we prove that the approximate Bellman updates are guaranteed to converge by a modified contraction analysis. We experimentally verify these results by developing an efficient Q-learning algorithm tailored to the networked setting. Experimental results on real-world graph data demonstrate that our Q-learning approach outperforms both k-step look-ahead and network-blind approaches, highlighting the importance of capturing and leveraging network effects where they exist.

1 Introduction

Public health challenges such as infectious disease control, vaccination strategies, and chronic illness management require sophisticated sequential decision-making under uncertainty, where timely decisions can significantly impact population health outcomes [World Health Organization, 2019]. The Restless Multi-Armed Bandit (RMAB) framework has emerged as a powerful tool for addressing such sequential decision-making problems under resource constraints. Prior work has successfully applied variations of RMABs to various public health settings, such as optimizing treatment strategies for infectious diseases [Mate et al., 2020], designing treatment policies for tuberculosis patients [Mate et al., 2021], efficient streaming-patient intervention planning [Mate et al., 2022], and fair resource allocation across patient cohorts [Li and Varakantham, 2022].

However, a significant limitation of traditional RMAB models is the assumption of independence among arms. In many public health applications, the state of one individual directly affects others due to network effects. For example, in epidemic processes infection propagates along contact networks, so an individual's health status changes the risk faced by their neighbors [Pastor-Satorras and Vespignani, 2001, Wang et al., 2003, Pastor-Satorras et al., 2015, Kiss et al., 2017]. During the COVID-19 pandemic, the impact of interventions such as vaccination or quarantine depended not only on who was targeted but also on the topology of the underlying interaction graph [World Health Organization, 2020, Funk et al., 2010]. Ignoring such dependencies can yield sub-optimal resource allocation, higher transmission rates, and increased morbidity. To model such interactions, the Independent Cascade (IC) model has been widely used to capture the probabilistic spread of influence through a network [Kempe et al., 2003].

Our work introduces the **Networked Restless Multi-Armed Bandit (NRMAB)** framework, which integrates the RMAB model with the Independent Cascade model to account for network effects. This enables more realistic representations of how interventions on one individual can influence the health states of others. By incorporating network effects, our model allows the action on one arm to influence not only its own state transitions but also those of neighboring arms through cascades.

We formulate Bellman's equation for this networked problem and prove that the value function is submodular. Because activation is probabilistic and arms can passively change states, traditional submodularity proofs for independent cascade no longer work [Kempe et al., 2003].We adapt the original proof to this setting, accommodating probabilistic activation and passive state changes. Submodularity in turn unlocks a greedy hill-climbing action-selection policy for Bellman equation whose return is at least $(1 - 1/e)$ of the optimal [Nemhauser et al., 1978].

We then establish that the Bellman operator with hill-climbing action selection is a γ-contraction. Because greedy selection can be sub-optimal, classical contraction proofs that rely on optimal action selection no longer apply. We design an equivalent multi-bellman operator with a meta-MDP and show this operator contracts under the supremum norm [Carvalho et al., 2023]. Value iteration converges linearly, and finite-horizon implementations inherit tight error bounds, ensuring practical algorithms remain stable and sample-efficient.

Building on this theoretical foundation, we develop a Q-learning algorithm for NRMABs. Our algorithm uses hill-climbing action selection for the Bellman equation to approximate the optimal policy without the need to compute the exact value function, which is computationally infeasible in large networks. We validate our approach through experiments on synthetic networks, demonstrating that our network-aware algorithm outperforms network-blind baselines, including the traditional Whittle Index policy [Whittle, 1988]. These results highlight the importance of capturing network effects in sequential decision-making problems and suggest that NRMABs can provide more effective intervention strategies in public health and other domains where networked interactions are significant.

2 Related Works

Restless multi-armed bandits RMABs, first introduced by Whittle [1988], extend the classic Multi-Armed Bandit framework to scenarios where each arm evolves over time regardless of whether it is selected, making a powerful model for decision-making problems in uncertain and evolving environments. Finding optimal policies for RMABs is PSPACE-hard [Papadimitriou and Tsitsiklis, 1999], leading to the development of various approximation algorithms, such as the Whittle index policy [Whittle, 1988]. RMABs have been applied in domains such as machine maintenance [Glazebrook et al., 2005], healthcare [Mate et al., 2020], and communication systems [Liu and Zhao, 2010].

Independent Cascade Model The Independent Cascade model, introduced by Kempe et al. [2003], captures the probabilistic spread of influence through networks and is a fundamental framework for studying diffusion processes in social networks. In this model, active nodes have a single chance to activate each inactive neighbor with certain probability, modeling phenomena such as information spread and epidemic propagation. Influence maximization – selecting a set of initial nodes to maximize the expected spread – is NP-hard but benefits from submodularity, which allows for efficient approximation algorithms with provable guarantees [Nemhauser et al., 1978]. Submodular function maximization has been extensively studied and applied to various network optimization problems [Leskovec et al., 2007, Chen et al., 2010].

Networked Bandits Prior work has explored extending RMABs to account for network effects. Ou et al. [2022] introduced a RMAB framework accounting for movement of people between physical locations. Herlihy and Dickerson [2023] incorporated network effect by giving each arm a "message" action that influences the transition probability of neighboring arms. Agarwal et al. [2024] modifies the restless multi-armed bandit problem such that the reward on each arm depends on the actions performed on its neighboring arms.

These works confirm the value of incorporating graph structure, yet each targets a specific form of coupling. Our NRMAB framework advances this foundation by modeling probabilistic cascades of state transitions and providing a submodular-greedy RL solution with contraction guarantees, yielding scalable policies for networked health-intervention problems.

Q-Learning Q-learning [Watkins and Dayan, 1992] is a model-free reinforcement learning algorithm that learns an optimal action-selection policy by iteratively updating Q-values based on observed rewards and transitions. The algorithm is well-suited for decision-making in Markov Decision Processes (MDPs) and has been widely applied in domains such as game-playing agents [Mnih et al., 2015]. While tabular Q-learning is effective for small state spaces, it suffers from scalability issues as the state-action space grows. Deep Q Networks (DQNs) [Mnih et al., 2015] address this limitation by approximating the Q-function using deep neural networks, enabling Q-learning to scale to large state spaces.

Particularly relevant is Khalil et al. [2017], who explored reinforcement learning for combinatorial optimization problems on graphs and demonstrated that graph structures can be leveraged to learn effective heuristics for NP-hard problems [Khalil et al., 2017]. This motivates our approach where we leverage Q-learning to optimize decision-making in dynamic networked environments.

3 Problem Setting

3.1 RMAB Problem Formulation

A RMAB [Whittle, 1988] consists of n independent arms that evolve in parallel. At each timestep the controller may activate at most k arms. Let $\mathcal{S} = \{0,1\}^n$ and $\mathcal{A} = \{\boldsymbol{a} \in \{0,1\}^n : \sum_{i=1}^n a_i = k\}$ where $s_i \in \{0,1\}$ denotes the two-state status of arm i and $a_i \in \{0,1\}$ denotes the two-action choice. Quality of the action is determined by a **reward function**, which we define in the next subsection.

Independent arm transition Given the state s and the action a of arm v, the state transitions to the next state u based on the transition probability $P_v(s,a,u)$. $P_v(s,a,u)$ is a probability distribution over the next states, which is independent for all arms. We write the independent transition of the current state of all nodes $\boldsymbol{s} = [s_v]_{v\in\mathcal{V}}$ by $P(\boldsymbol{u}|\boldsymbol{s},\boldsymbol{a}) = \prod_{v\in\mathcal{V}} P_v(s_v, a_v, u_v)$.

Assumption 1. *We assume that active actions yield higher probabilities of beneficial transitions compared to passive actions:* $P(s=0,a=1,u=1) \geq P(s=0,a=0,u=1)$ *and* $P(s=1,a=1,u=1) \geq P(s=1,a=0,u=1)$.

This compact MDP representation, standard in modern RMAB surveys (e.g. no-Mora, 2023), underpins the network extensions developed in subsequent sections.

3.2 Independent cascade

We now add the IC model [Kempe et al., 2003]. We connect arms (now called nodes) through undirected edges $e \in \mathcal{E}$, where each edge has a weight $0 < w_e < 1$ that represents the probability an active node activates its neighbor via a cascade. We use a function $P_G(\boldsymbol{s}'|\boldsymbol{u})$ to denote the probability that the temporary state $\boldsymbol{u}$ cascades to the next state $\boldsymbol{s}' = [s'_v]_{v\in\mathcal{V}}$ through the graph G and the cascade probability of each edge.

Transition kernel Coupling the arm-wise dynamics $P(\boldsymbol{u} \mid \boldsymbol{s},\boldsymbol{a})$ from the RMAB with the cascade yields the full MDP kernel

$$P(\boldsymbol{s}' \mid \boldsymbol{s},\boldsymbol{a}) \;=\; \sum\nolimits_{\boldsymbol{u}\in\{0,1\}^n} P(\boldsymbol{u} \mid \boldsymbol{s},\boldsymbol{a})\, P_G(\boldsymbol{s}' \mid \boldsymbol{u}), \tag{1}$$

where $P(\boldsymbol{u} \mid \boldsymbol{s},\boldsymbol{a}) = \prod_{v\in\mathcal{V}} P_v(u_v \mid s_v, a_v)$ is the independent arm transition introduced earlier.

Reward objective Our goal is to select the optimal k nodes at each timestep to maximize the cumulative reward over multiple timesteps t. The reward function $R(s,a)$ is the immediate reward received per step after taking action a in state s. $\mathcal{V}$ represents the set of all nodes in the graph, and $r(v)$ is the value associated with node v if it is active ($s(v)=1$), or zero otherwise. Cumulative reward is formalized using the discounted return where γ is the discount factor ($0 \leq \gamma < 1$) that prioritizes immediate rewards over distant future rewards.

$$R(\boldsymbol{s}, \boldsymbol{a}) = \sum\nolimits_{v \in \mathcal{V}} r(v), \qquad \sum\nolimits_{t=0}^{\infty} \gamma^t R(\boldsymbol{s_t}, \boldsymbol{a_t}), \tag{2}$$

3.3 Network RMAB Problem Formulation

An instance of the network RMAB problem is composed of a graph $G = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ represents an arm that can transition between different states $s \in \mathcal{S}$. Each iteration we have a budget constraint on the actions: $\sum_{i \in [n]} a_i \leq k$. We can cast the Networked RMAB as a discounted Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$.

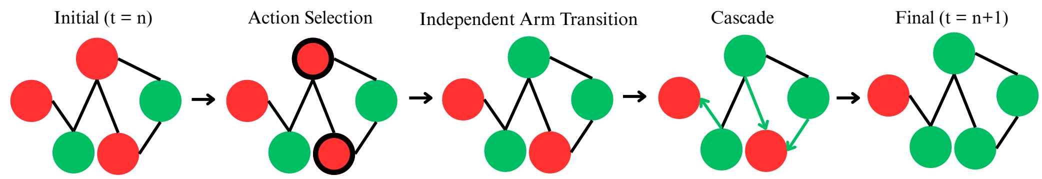

Figure 1: Visual representation of a single Networked RMAB timestep: initial state, selection of k active actions, independent transitions, cascade propagation, and resulting next state.

4 Methodology

We representing action value using the Bellman equation to select the best actions for each timestep:

$$V(\boldsymbol{s}) = \max_{\boldsymbol{a} \in \mathcal{A}} Q(\boldsymbol{s}, \boldsymbol{a}), \tag{3}$$

$$Q(\boldsymbol{s}, \boldsymbol{a}) = R(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum\nolimits_{u} P(\boldsymbol{u} \mid \boldsymbol{s}, \boldsymbol{a}) \sum\nolimits_{s'} P_G(\boldsymbol{s'} \mid \boldsymbol{u}) V(\boldsymbol{s'}), \tag{4}$$

To solve NRMAB problems, we can write down the Bellman equation in Equation 3 to apply existing RL algorithms like DQN [Watkins and Dayan, 1992]. However, the action selection in Equation 3 is computationally infeasible for large state-action spaces due to exponentially many possible actions.

Our objective is to develop an algorithm capable of consistently selecting near-optimal actions in a scalable manner. To achieve this, we exploit the submodularity of $Q(\boldsymbol{s}, \boldsymbol{a})$ to use a *hill-climbing* action selection that only takes $O(n^2)$ time. Because of submodularity, this algorithm yields a $(1 - 1/e)$-approximation to the true maximum. We refer to this algorithm as Bellman Equation with Hill-Climbing Action Selection.

Algorithm 1 Hill-Climbing Action Selection for the Bellman Equation

Require: State $s \in \mathcal{S}$, budget k, Q-function $Q(\cdot, \cdot)$
- $A \leftarrow \varnothing$ {initial empty action set}
- **for** $j = 1$ **to** k **do**
  - $a^\star \leftarrow \arg \max_{a \in \mathcal{A} \setminus A} Q\big(s,\ A \cup \{a\}\big)$
  - $A \leftarrow A \cup \{a^\star\}$
- **end for**
- **return** A {greedily constructed size-k action set}

4.1 Submodularity of the Q-Function

Establishing that the state–action value $Q(\boldsymbol{s}, \boldsymbol{a})$ is submodular in the action set $\boldsymbol{a}$ is pivotal: it unlocks the $1 - \frac{1}{e}$ performance guarantee of a greedy hill-climbing strategy that provably avoids the exponential blow-up of evaluating all $\binom{n}{k}$ action combinations [Kempe et al., 2003]. We proceed with a proof of the submodularity of $Q(\boldsymbol{s}, \boldsymbol{a})$.

Theorem 1 (Submodularity)**.** *Given a known $V(\boldsymbol{s})$ function and a constant state $\boldsymbol{s}$, we show that $Q(\boldsymbol{s}, \boldsymbol{a})$ is submodular with respect to a.*

Proof sketch. (Full proof in A.1)

We show that for any $A \subseteq B \subseteq N$ and $t \notin B$:

$$Q(s, A \cup \{t\}) - Q(s, A) \geq Q(s, B \cup \{t\}) - Q(s, B)$$

In $Q(s, a)$, the reward function $R(s, a)$ is inherently submodular. We focus on the expected future value component:

$$\sigma(a) = \sum\nolimits_{s',u} P_G(s' \mid u) P(u \mid s, a) V(s')$$

We model the state transitions and cascades using coupled probabilistic simulations. For each node $v \in \mathcal{V}$, we simulate two coin flips: x_v, which represents the node's outcome in the transition step under a passive action, and y_v, the same node's outcome under an active action. We couple these coinflips such that if x results in activation, the corresponding y must also result in activation. For each edge $e \in \mathcal{E}$, we simulate a coin flip z_e to determine if an active node activates its neighbor via a cascade. With a full set of coinflips X, Y, Z, we deterministically know the set of active nodes after applying an action. Let $\sigma_{XY}(A)$ denote the set of active nodes after the Transition Step, and $\sigma_Z(\sigma_{XY}(A))$ denote the set of active nodes after both Transition and Cascade Steps.

We know that $\sigma_{XY}(A) \subseteq \sigma_{XY}(B)$ for $A \subseteq B$ and $\sigma_{XY}(A \cup \{t\}) \setminus \sigma_{XY}(A) = \sigma_{XY}(B \cup \{t\}) \setminus \sigma_{XY}(B)$. Using these properties, the submodularity inequality can be rewritten for the independent cascade as:

$$\begin{aligned} &\sigma_Z(\sigma_{XY}(A \cup \{t\})) - \sigma_Z(\sigma_{XY}(A)) \\ \geq\ &\sigma_Z(\sigma_{XY}(B \cup \{t\})) - \sigma_Z(\sigma_{XY}(B)). \end{aligned}$$

Since $V(s)$ depends directly on the total weighted node value and each active node contributes positively, $V(\sigma_{X,Y,Z}(A))$ is also submodular. Consequently, our expected future value can be formulated as:

$$\sigma(A) = \sum\nolimits_{X,Y,Z} P(XYZ) \cdot V(\sigma_{X,Y,Z}(A))$$

which is a non-negative linear combination of submodular functions, maintaining submodularity. Therefore, $Q(s, a)$, being a sum of submodular functions, is itself submodular. $\square$

This submodularity allows us to utilize a $1 - \frac{1}{e}$ optimality guarantee when using hill-climbing algorithm in Algorithm 1, a $O(n^2)$ algorithm scalable to larger problems.

4.2 Proof of Contraction for the Bellman Equation with Hill-Climbing Action Selection

Given submodularity of $Q(s, a)$, each action set of Bellman equation with hill-climbing action selection has an optimality guarantee of $1 - \frac{1}{e}$. We want to show that even under this approximate action selection, the Bellman operator for Bellman equation with hill-climbing action selection (Definition 1) is guaranteed to converge.

Definition 1 (Bellman Operator for Bellman Equation with Hill-Climbing Action Selection)**.** *Let $\mathcal{S}$ be the set of global states, and let $\mathcal{A}$ be the set of all* single *actions (e.g., single nodes) that can be targeted at a given timestep. Suppose we want to build a final action set of size k from $\mathcal{A}$, subject to a budget of k.*

We define the Bellman Operator for this variation of the Bellman equation as B,

$$BV(s) \ = \ \max_{a^{hc} \in \mathcal{A}} \{R(s, a^{hc}) + \gamma \sum\nolimits_{u \in \mathcal{S}} P\big(u \mid s, a^{\mathrm{hc}}(s)\big) \sum\nolimits_{s' \in \mathcal{S}} P_G\big(s' \mid u\big)\, V(s')\}, \qquad (5)$$

where a^{hc} is the output of Algorithm 1 and Q is given in Equation (4).

Based on Theorem 4.3 in [Nemhauser et al., 1978] and the submodularity given by Theorem 1, we can show that the greedy algorithm in Algorithm 1 discovers a set of actions that yields a $V(s)$ at least $1 - 1/e$ of the true maximum.

However, because Algorithm 1 does not always yield the optimal action, the traditional Banach Fixed-Point argument for Bellman Operator contraction does not work (see Appendix A.2), and we instead construct an alternative approach exploiting the structure of repeated hill-climbing updates.

Theorem 2 (Contraction). *Bellman Operator for Bellman Equation With Hill-Climbing Action Selection (B) is a γ contraction under the supremum norm $\|\cdot\|_\infty$.*

Proof Sketch. Our proof centers around redefining B as a multi-bellman operator as defined by Carvalho et al. [2023].

Intuition Because the Bellman equation with hill climbing action selection finds the approximate rather than the best set of actions at each timestep, the traditional proof for contraction does not work. Thus, we deconstruct our Bellman Operator for Hill-Climbing Action Selection (definition 1) into a multi-bellman operator $\widetilde{B}$ as defined by [Carvalho et al., 2023]. Each application of this operator selects the single next best action to take given a state and partial action set, effectively mimicking one step in the hill-climbing algorithm. Applying this k times becomes equivalent to one application of B. We prove this equivalence, and borrowing the proof of convergence for the multi-bellman operator, prove that B thus converges.

Definition 2 (Multi-Bellman Operator for the Hill-Climbing Variant). *We recast our incremental set-building procedure as follows. Let each "meta-state" be denoted by $\widetilde{s} = (s, A, t)$, where s is the environment state, $A \subseteq \mathcal{A}$ is the set of actions selected so far, and t is the current timestep. Additionally, define $\widetilde{\gamma}^k = \gamma$. Write $\widetilde{s}_0 = \big(s, \varnothing, 0\big)$, $\widetilde{s}_1 = \big(s, \{a_0\}, 1\big)$, $\ldots$, $\widetilde{s}_k = \big(s, \{a_0, \ldots, a_{k-1}\}, k\big)$. Define the modified reward*

$$\widetilde{R}\big(\widetilde{s}, a\big) = \frac{1}{\widetilde{\gamma}^t}\big(R\left(s, A \cup \{a\}\right) - R(s, A)\big).$$

Then for $t < k$, picking a single *new action a corresponds to moving from $\widetilde{s}_j = (s, A, t)$ to $\widetilde{s}_{j+1} = \big(s, A \cup \{a\}, t+1\big)$, and we define*

$$(\widetilde{B}V)\big(\widetilde{s}_j\big) \;=\; \max_{a \in \mathcal{A} \setminus A} \Big\{ \widetilde{R}\big(\widetilde{s}_j,\, a\big) \;+\; \widetilde{\gamma}\, V\big(\widetilde{s}_{j+1}\big) \Big\}.$$

***At** $t = k$**:** the action set A is fully chosen (i.e. $\widetilde{s}_k = (s, \{a_0, \ldots, a_{k-1}\}, k)$). One more application of $\widetilde{B}$ (when $t = k$) then* applies *these actions to s, causing a transition to s'.*

$$(\widetilde{B}V)(s, A, k) \;=\; \widetilde{\gamma}\, \mathbb{E}_{s'}\big[\, V\big(s', \varnothing, 0\big)\big]$$

Hence, $\widetilde{B}$ captures both the step-by-step incremental selection of actions for $t < k$, and the final transition applying the chosen set A when $t = k$.

After defining $\widetilde{B}$, we leverage it to construct a MDP $\mathcal{M}_{hc}$ for this Bellman operator – one that is equivalent to the MDP for B.

Theorem 3 (Hill-Climbing Equivalence). *Applying k iterations of Multi-Bellman Operator for Hill-Climbing Variant (Definition 2) is equivalent to one application of Bellman Operator for Bellman equation with hill climbing action selection (Definition 1) with an action budget of k.*

Proof Sketch. (Full proof in Appendix A.4) Apply $\widetilde{B}$ exactly k times. Each step adds a *marginal* reward $\widetilde{R}(\widetilde{s}_t, a_t) = \widetilde{\gamma}^{-(k-t)}\big(R(s, A_t \cup \{a_t\}) - R(s, A_t)\big)$, so the discounted sum telescopes:

$$\sum\nolimits_{t=0}^{k-1} \widetilde{\gamma}^{\,t} \widetilde{R}(\widetilde{s}_t, a_t) = R\big(s, \{a_0, \ldots, a_{k-1}\}\big).$$

After the k-th pick the augmented state resets to the environment state s', and because $\widetilde{\gamma}^{\,k} = \gamma$ the future value is $\gamma V(s')$. Hence

$$(\widetilde{B}^{\,k} V)(s) = R\big(s, \{a_0, \ldots, a_{k-1}\}\big) + \gamma V(s') = (BV)(s).$$

Therefore $\widetilde{B}^{\,k}$ coincides with the standard Bellman update, so all usual contraction and convergence results carry over. This concludes the proof of Theorem 3. ☐

Given the equivalence shown in Theorem 3, we can directly apply the proof of Lemma 1 in Carvalho et al. [2023] to $\widetilde{B}$. By following their proof, we find that $\widetilde{B}^k$ is a $\widetilde{\gamma}^k$ contraction (for the full process see Appendix A.3). Because $\widetilde{\gamma}^k = \gamma$ and $\widetilde{B}^k = B$, B is a γ-contraction. This concludes the proof of Theorem 2. $\square$

Thus, we have show that even though greedy hill-climbing action selection only provides an action $1 - \frac{1}{e}$ of the optimum, Bellman equation using hill-climbing action selection is still a γ-contraction.

4.3 Deep Q-Learning with Hill-Climbing

To solve a NRMAB problem, we propose a Deep Q-Network using hill-climbing. Similar to a traditional DQN, our neural network takes in a representation of the state and action and pass it through three fully connected hidden layers to producing a Q-value for each state-action pair $Q(\boldsymbol{s}, \boldsymbol{a})$. To scale this to large action spaces, we iterate through the list of all possible single actions and utilize a neural network to predict the Q value of each single action a. Then, we greedily select the k actions with the highest Q-values. This leverages the submodular properties of the Bellman equation to achieve the $1 - \frac{1}{e}$ performance guarantee. We combine DQN and hill-climbing action selection to design a scalable Q-learning algorithm (Algorithm 2) to solve NRMAB problems.

Algorithm 2 Hill-Climbing DQN

Initialization: Neural network $Q(s, a; \theta)$, replay buffer
while until θ converges **do**
  Hill-climbing action: intervention set $A = \emptyset$.
  while $|A| < k$ (budget for intervention) **do**
    Solve $v^* = \arg\max_{v \in V} Q(s, 1_{A \cup \{v\}})$
    Update $A \leftarrow A \cup \{v\}$
  end while
  Execute $a = 1_A$ and collect experience
  DQN Updates: Sample mini-batches from the replay buffer and run gradient descent to update θ.
end while
Output: Q network parameter θ

Graph Neural Network Optimization In order to better account for network effects, we optimize this approach by implementing a graph neural network in addition to a simple DQN to leverage relational dependencies within the network. We maintain all other properties for the GNN, including using hill-climbing action selection.

5 Experiments

5.1 Domain

The motivating application is health-care intervention planning, where limited resources (e.g., vaccinations, diagnostic tests, treatment slots, adherence reminders) must be allocated over time while infection or non-adherence spreads through a contact network. Classical RMAB models treat patients (arms) as independent, yet in epidemiology and behavioral health network spill-overs can impact outcomes. Our experiments therefore contrast the effectiveness of network-blind approaches with our network-aware algorithm developed for the NRMAB model.

5.2 Simulation

We evaluate our algorithms on a real network collected from a village in India through household surveying Ou et al. [2021], augmenting it with synthetic node attributes and cascade probabilities. The data is given as an edgelist where each edge represents real world contact. The data depicts a multigraph, but we remove redundant edges to create a simple graph. After processing, the network contains 202 nodes and 692 edges. We use the edgelist to build our graph, then randomly generate attributes for each node in the graph and set the cascade probability. The results of the comparison are shown in Figure 2.

The DQN is independently defined and trained using TianShou and PyTorch for the neural network, and Gymnasium for the simulation environment. GNN uses PyTorch, PyTorch Geometric, and Gymnasium. DQN is trained over seven epochs of 1000 steps, and GNN is trained for 100 episodes. Higher training times show minimal improvement.

After training, each algorithm is evaluated using 10 random seeds, with 50 simulations per seed, each running for 30 timesteps. We collect mean cumulative reward, mean reward per timestep, mean activation percentage from each seed. All experiments can be run locally on a single RTX 3050Ti GPU in < 12 hours.

5.3 Real World Environment

The India contact graph ($n = 202,\ |E| = 692$) is adopted as a high-fidelity synthetic contact network: each person is an individual arm and every edge carries a fixed cascade probability $w_{vw} = 0.03$, representing the chance that health resources or infections propagate between close contacts. Empirical work shows that such digitally inferred graphs capture the dominant pathways of disease and behavioral diffusion in real populations [Salathé and Jones, 2010].

We initialize the system with no active nodes and impose an intervention budget of $k = 30$ actions per timestep, mimicking limited daily vaccine or test capacity. Policies are evaluated against other intervention and no-intervention baselines so that cumulative-reward gains translate directly into expected infections or adverse events averted.

5.4 Baseline Algorithms

To evaluate the effectiveness of our DQN and GNN algorithms, we compare them with three other algorithms. Tabular Q-learning solves the full Bellman equation for each state-action pair for small state-action sizes. 1-Step Look-Ahead performs hill-climbing by calculating the value of activating a node in a state and taking into account network effect but ignoring future states. Whittle Index is a traditionally optimal method for solving RMABs without considering network effects.

6 Results & Discussion

6.1 Performance on Real-World Graph

Figure 2 shows the average percentage of activated nodes over 30 timesteps on the India contact network, aggregated across simulations on 10 random seeds. The GNN-based policy consistently achieves the highest activation, converging to over 82% by timestep 30. DQN and Whittle follow closely, leveling off around 80–81%, while the 1-step lookahead trails slightly behind. In contrast, the no-intervention baseline stabilizes under 71%, highlighting the effectiveness of all intervention strategies relative to doing nothing.

These results suggest that explicitly accounting for network structure can substantially improve the reach of health interventions over time. In practical terms, this means more individuals are consistently reached and maintained in a healthy state, even when resources are limited. Though in this experiment the performance gap relative to naive baselines is relatively small, the significant number and diversity of trials run suggest a statistically significant improvement. More complex scenarios should significantly increase the performance disparity in favor of GNN and DQN algorithms.

6.2 Optimality Verification

Figure 3 shows the performance of DQN with hill-climbing compared to Tabular Q-learning, an approach that gives near-optimal solutions at every timestep. We validate the optimality guarantee delivered by the submodular greedy algorithm as shown in Theorem 1: DQN performs at a very similar level to Tabular Q-learning. The extreme similarity in performance may be due to the small graph size tested, as the runtime of Tabular Q Learning rapidly explodes on larger graphs.

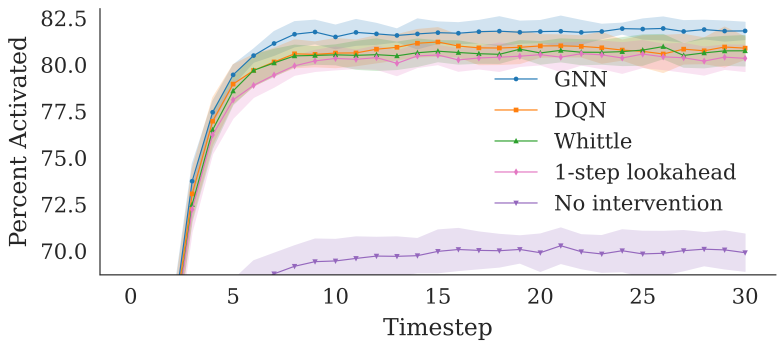

Figure 2: Mean ± SD fraction of activated nodes over 30 timesteps on the India contact network ($n = 202,\ |E| = 692;\ k = 20;$ 10 seeds $\times$ 50 runs) shows the GNN stabilizing near 82% activation and consistently outperforming DQN, Whittle index, 1-step look-ahead, and the no-intervention baseline.

6.3 Computational Cost

Figure 4 shows the runtime difference between GNN, DQN with hill-climbing, and Tabular Q-learning. Tabular Q-learning runtime increases exponentially with increasing nodes, while GNN and DQN with hill-climbing increases about linearly. This aligns with theoretical runtime benefits in Algorithm 1 while achieving comparative performance to the optimal algorithm(see Figure 3).

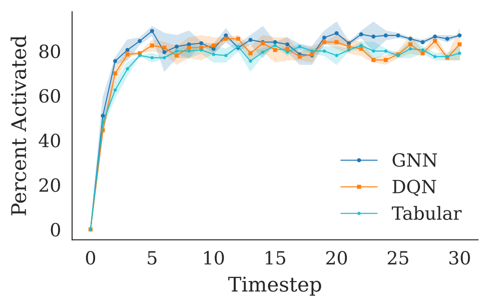

Figure 3: Mean ± SD activation fraction over 30 timesteps on a 10-node graph. DQN and GNN match tabular Q-learning's near-optimal performance in networked RMABs.

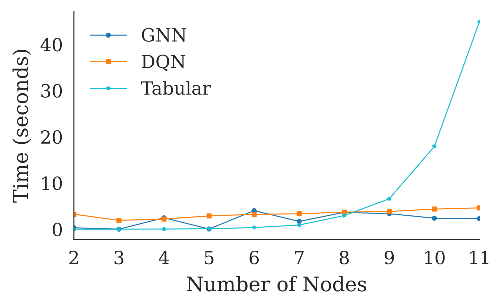

Figure 4: Total runtime (per epoch runtime for DQN and GNN) versus graph size n. Results reveal tabular Q-learning's exponential run-time growth, while DQN and GNN grow linearly.

7 Conclusion & Future Work

We introduced the *NRMAB* model to capture network spill-overs in sequential resource allocation, proved its Bellman operator is both submodular and a γ-contraction – so a greedy hill-climbing policy enjoys a $(1 - 1/e)$ guarantee – and demonstrated a GNN implementation that outperforms strong baselines on contact-network data. Current experiments still rely on synthetic node attributes and fixed cascade rates; transforming real world observations into node attributes and cascade weights is pivotal to further experimentation. Current assumptions of full observability and static cascade probabilities can also be relaxed for broader applicability.

In future work, collecting data tailored to NRMAB enables experiments that bridge theory and practice. Partial observability modeled via a belief-state (POMDP) NRMAB can align the framework with real-world deployments. Allowing edge-specific cascade probabilities that evolve over time can capture changing behavior. Finally, our results hint at fairness–efficiency trade-offs when node values vary widely; embedding fairness constraints directly into the hill-climbing step could yield more socially responsible policies.

References

A. Agarwal, A. Agarwal, L. Masoero, and J. Whitehouse. Multi-armed bandits with network interference. *arXiv preprint arXiv:2405.18621*, 2024.

D. S. Carvalho, P. A. Santos, and F. S. Melo. Multi-bellman operator for convergence of q-learning with linear function approximation, 2023. URL `https://arxiv.org/abs/2309.16819`.

W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, page 1029–1038, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300551. doi: 10.1145/1835804.1835934. URL `https://doi.org/10.1145/1835804.1835934`.

S. Funk, M. Salathé, and V. A. A. Jansen. Modelling the influence of human behaviour on the spread of infectious diseases: a review. *Journal of the Royal Society Interface*, 7(50):1247–1256, 2010.

K. Glazebrook, H. Mitchell, and P. Ansell. Index policies for the maintenance of a collection of machines by a set of repairmen. *European Journal of Operational Research*, 165(1):267–284, 2005. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2004.01.036. URL `https://www.sciencedirect.com/science/article/pii/S0377221704000876`.

C. Herlihy and J. Dickerson. Networked restless bandits with positive externalities. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-03)*, pages 137–146, New York, NY, USA, 2003. ACM. doi: 10.1145/956750.956754.

E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song. Learning combinatorial optimization algorithms over graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/d9896106ca98d3d05b8cbdf4fd8b13a1-Paper.pdf`.

I. Z. Kiss, J. C. Miller, and P. L. Simon. *Mathematics of Epidemics on Networks*. Springer, 2017.

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Vanbriesen, and N. Glance. Cost-effective outbreak detection in networks. volume 420-429, pages 420–429, 08 2007. doi: 10.1145/1281192.1281239.

D. Li and P. Varakantham. Efficient resource allocation with fairness constraints in restless multi-armed bandits. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022)*, pages 1158–1167, 2022.

K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2010. doi: 10.1109/TIT.2010.2068950.

A. Mate, J. A. Killian, H. Xu, A. Perrault, and M. Tambe. Collapsing bandits and their application to public health interventions. *Neural Information Processing Systems*, 2020.

A. Mate, A. Perrault, and M. Tambe. Risk-aware interventions in public health: Planning with restless multi-armed bandits. In *AAMAS '21*, 2021.

A. Mate, A. Biswas, C. Siebenbrunner, S. Ghosh, and M. Tambe. Efficient algorithms for finite horizon and streaming restless multi-armed bandit problems. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)*, pages 880–889. IFAAMAS, 2022.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

J. N. no-Mora. Markovian restless bandits and index policies: A review. *Mathematics*, 11(7):1639, 2023.

H. Ou, H. Chen, S. Jabbari, and M. Tambe. Active screening for recurrent diseases: A reinforcement learning approach. *CoRR*, abs/2101.02766, 2021. URL `https://arxiv.org/abs/2101.02766`.

H.-C. Ou, C. Siebenbrunner, J. Killian, M. B. Brooks, D. Kempe, Y. Vorobeychik, and M. Tambe. Networked restless multi-armed bandits for mobile interventions. 2022. URL `https://arxiv.org/abs/2201.12408`.

C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queueing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.

R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86:3200–3203, Apr 2001. doi: 10.1103/PhysRevLett.86.3200.

R. Pastor-Satorras, C. Castellano, P. V. Mieghem, and A. Vespignani. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87(3):925–979, 2015.

M. Salathé and J. H. Jones. Dynamics and control of diseases in networks with community structure. *PLoS Computational Biology*, 6(4):e1000736, 2010.

Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. *22nd International Symposium on Reliable Distributed Systems*, pages 25–34, 2003.

C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992. ISSN 1573-0565. doi: 10.1007/BF00992698.

P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.

World Health Organization. *World Health Statistics 2019: Monitoring Health for the SDGs, Sustainable Development Goals*. World Health Organization, Geneva, Switzerland, 2019.

World Health Organization. *World Health Organization guidelines on COVID-19: Evidence synthesis and recommendations*. World Health Organization, 2020.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract (lines 10–21) and Introduction (§1) list the paper's four concrete contributions: (i) formulation of the Networked RMAB model, (ii) proof of submodularity enabling a $(1 - 1/e)$ hill-climbing approximation, (iii) contraction analysis guaranteeing convergence, and (iv) empirical validation with Q-learning and GNN baselines. These are exactly the theoretical developments in §4–§5 and the experimental results in §6; no claims are made beyond this scope or outside the stated assumptions (finite discounted MDP, fixed cascade probabilities).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The "Limitations & Future directions" paragraph in §7 explicitly flags sparse real-world validation, the assumption of fixed, known cascade probabilities, and the need for richer data, online re-estimation, and POMDP extensions, thereby squarely addressing the work's key constraints.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Each theoretical result (Theorems 1, 2, and 3) states all required assumptions (e.g., Assumption 1 on transition probabilities) and is backed by a full, numbered proof in the appendices (with sketches in Sec. 4), satisfying the checklist criteria.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 5 reports key graph statistics, hyper-parameters, and training details, and the full source code for running experiments code plus processed India-contact dataset will be released in the supplemental zip, enabling end-to-end replication of experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The supplementary zip furnishes the full source code, the processed India-contact edgelist, and a README with exact run commands, giving reviewers open access and step-by-step instructions to recreate every experimental figure.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Key choices in experimental setting are found in §4 and §5. Full experimental details are present in the source code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are present in all graphs in §5 for all figures. Methodology for calculating the error bars (standard deviation across average values in random seeds) is explained.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: §5 specifies that experiments are lightweight and can be run in most environments without specialized equipment in < 12 hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Datasets are used ethically and review of the paper shows no conflicts with the NeurIPS code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: §1 introduces the potential positive societal impacts of the work performed. §7 touches upon potential issues regarding fairness and equity that can arise from misuse of the model, addressing negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We determine that the model proposed in the paper does not have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The dataset used for the real-graph experiment is §5 is explicitly credited to Ou et al. [2021], and all licenses and terms of use can be found by following the citation. All licenses and terms of use are respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper releases an implementation of NRMAB (code asset); the supplemental zip includes a README describing files, run commands, dependencies, and license, thus meeting the documentation requirement for new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not include crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not include research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Technical Appendices and Supplementary Material

Technical appendices with additional results, figures, graphs and proofs may be submitted with the paper submission before the full submission deadline (see above), or as a separate PDF in the ZIP file below before the supplementary material deadline. There is no page limit for the technical appendices.

A.1 Full Submodularity Proof

Theorem 1 (Submodularity)**.** *Given a known $V(\boldsymbol{s})$ function and a constant state $\boldsymbol{s}$, we show that $Q(\boldsymbol{s}, \boldsymbol{a})$ is submodular with respect to a.*

Proof. We demonstrate that our implementation of the Bellman equation exhibits submodular properties, which are crucial for the efficiency and effectiveness of greedy algorithms. Submodularity ensures that the marginal gain of adding an element to a set decreases as the set grows, a property leveraged in influence maximization.

Formally, we aim to show that for any $A \subseteq B \subseteq N$ and $v \notin B$:

$$Q(s, A \cup \{t\}) - Q(s, A) \geq Q(s, B \cup \{t\}) - Q(s, B)$$

where $Q(s, A)$ represents the value of taking action set A in state s.

The reward function $R(s, a)$ is inherently submodular. We focus on the expected future value component:

$$\sigma(a) = \sum_{s', u} P_G(s' \mid u) P(u \mid s, a) V(s')$$

To analyze $\sigma(a)$, we model the state transitions using coupled probabilistic simulations. Specifically, for each node $v \in \mathcal{V}$, we simulate two coin flips:

- x_v: Outcome under passive action with probability $P(s = s_0, a = 0, u = u_1)$.
- y_v: Outcome under active action with probability $P(s = s_0, a = 1, u = u_1)$.

We couple these coin flips such that if x_v results in activation ($u = 1$), then y_v also results in activation. This coupling reflects the assumption that active actions have transition probabilities at least as good as passive actions, ensuring:

$$P(s = 0, a = 1, u = 1) \geq P(s = 0, a = 0, u = 1),$$
$$P(s = 1, a = 1, u = 1) \geq P(s = 1, a = 0, u = 1).$$

Additionally, for each edge $e \in \mathcal{E}$, we simulate a coin flip z_e with bias $p_{v,w}$ to determine if an active node activates its neighbor via a cascade. A heads outcome denotes an active edge, leading to activation, while tails denote no activation.

Through these coupled simulations, we can deterministically determine the set of active nodes after applying an action. Let $\sigma_{XY}(A)$ denote the set of active nodes after the Transition Step, and $\sigma_Z(\sigma_{XY}(A))$ denote the set of active nodes after both Transition and Cascade Steps.

We establish the following properties:

1. $\sigma_{XY}(A) \subseteq \sigma_{XY}(B)$ for $A \subseteq B$. This is because adding more actions (from A to B) cannot decrease the set of active nodes due to the coupling of x_n and y_n.
2. $\sigma_{XY}(A \cup \{t\}) \setminus \sigma_{XY}(A) = \sigma_{XY}(B \cup \{t\}) \setminus \sigma_{XY}(B) = v'$, where v' represents the newly activated nodes resulting from adding action t. This holds because the additional action t affects nodes in the same manner regardless of the existing set A or B, thanks to the coupling ensuring $y_v \geq x_v$.

Using these properties, the submodularity inequality can be rewritten for the independent cascade as:

$$\sigma_Z(\sigma_{XY}(A \cup \{v\})) - \sigma_Z(\sigma_{XY}(A))$$
$$\geq \sigma_Z(\sigma_{XY}(B \cup \{v\})) - \sigma_Z(\sigma_{XY}(B)).$$

This inequality demonstrates that the number of active nodes after an action is submodular with respect to the size of the action set.

Since $V(s)$ depends directly on the total weighted node value and each active node contributes positively, $V(\sigma_{X,Y,Z}(A))$ is also submodular. Consequently, our expected future value can be formulated as:

$$\sigma(A) = \sum_{X,Y,Z} P(XYZ) \cdot V(\sigma_{X,Y,Z}(A))$$

This is a non-negative linear combination of submodular functions, maintaining submodularity. Therefore, $Q(s, a)$, being a sum of submodular functions, is itself submodular.

□

A.2 Failure in Traditional Proof for Bellman Equation Convergence

For any two value functions $V, W : \mathcal{S} \to \mathbb{R}$, let

$$(HV)(s) = \max_{a \in \mathcal{A}} \Big\{ R(s,a) + \gamma\, \mathbb{E}[V(S') \mid s, a] \Big\},$$

and define HW analogously. Choose $a^\star(s) \in \arg\max_a Q_V(s,a)$, where $Q_V(s,a) = R(s,a) + \gamma \mathbb{E}[V(S') \mid s, a]$. Then

$$\begin{aligned}
|(HV)(s) - (HW)(s)| \;&=\; \Big| Q_V\big(s, a^\star(s)\big) - \max_a Q_W(s,a) \Big| \\
&\le \Big| Q_V\big(s, a^\star(s)\big) - Q_W\big(s, a^\star(s)\big) \Big| \\
&\le \gamma\, \|V - W\|_\infty,
\end{aligned}$$

and taking the supremum over s yields the γ-contraction.

The proof relies on re-using the *same* action $a^\star(s)$ under both V and W. Algorithm 1, however, returns $\widetilde{a}(s, V)$ that is merely near-optimal for V; when W differs from V, the algorithm may choose an entirely different $\widetilde{a}(s, W)$. Consequently we can bound only

$$\big|(\widetilde{H}V)(s) - (\widetilde{H}W)(s)\big| \;\le\; \underbrace{\big|Q_V\big(s, \widetilde{a}(s,V)\big) - Q_W\big(s, \widetilde{a}(s,V)\big)\big|}_{\le \gamma \|V - W\|_\infty} \;+\; \underbrace{\big|Q_W\big(s, \widetilde{a}(s,V)\big) - Q_W\big(s, \widetilde{a}(s,W)\big)\big|}_{\text{loss from approximate actions}},$$

and the second term has *no* γ factor. Hence $\widetilde{H}$ need not be a contraction, and classical Banach-fixed-point arguments fail. The convergence analysis in Section 4.2 circumvents this obstacle by treating B as a Multi-Bellman operator and exploiting the structure of repeated hill-climbing updates instead of relying on the traditional contraction argument.

A.3 Expanded Proof of Contraction from Carvalho et al. [2023]

Theorem 2 (Contraction)**.** *Bellman Operator for Bellman Equation With Hill-Climbing Action Selection (B) is a γ contraction under the supremum norm $\|\cdot\|_\infty$.*

Proof. Recall from Theorem 3 that $\widetilde{\mathbf{B}}^k$ is equivalent to the k-fold composition of a single-step operator $\widetilde{\mathbf{B}}$, where the "state" is the meta-state $\widetilde{s} = (s, A, t)$, and we add one action at a time.

Single-step contraction. Take any two Q-functions, Q_1 and Q_2. We compute:

$$\|\widetilde{\mathbf{B}}Q_1 - \widetilde{\mathbf{B}}Q_2\|_\infty \;=\; \max_{(\widetilde{s}_0, \widetilde{a}_0)} \Big| \big[\widetilde{R}(\widetilde{s}_0, a_0) + \widetilde{\gamma} \max_{a_1} Q_1(\widetilde{s}_1, a_1)\big] \;-\; \big[\widetilde{R}(\widetilde{s}_0, a_0) + \widetilde{\gamma} \max_{a_1} Q_2(\widetilde{s}_1, a_1)\big] \Big|.$$

Since $\widetilde{R}(\widetilde{s}_0, a_0)$ cancels, we get

$$\|\widetilde{\mathbf{B}}Q_1 - \widetilde{\mathbf{B}}Q_2\|_\infty \;=\; \widetilde{\gamma} \max_{(\widetilde{s}_0, a_0)} \Big| \max_{a_1} Q_1(\widetilde{s}_1, a_1) \;-\; \max_{a_1} Q_2(\widetilde{s}_1, a_1) \Big|.$$

Using the standard inequality $\big|\max_x f(x) - \max_x g(x)\big| \;\le\; \max_x \big|f(x) - g(x)\big|$, we obtain

$$\|\widetilde{\mathbf{B}}Q_1 - \widetilde{\mathbf{B}}Q_2\|_\infty \;\le\; \widetilde{\gamma} \max_{(\widetilde{s}_0, a_0)} \big|Q_1(\widetilde{s}_1, a_1) - Q_2(\widetilde{s}_1, a_1)\big| \;=\; \widetilde{\gamma}\, \|Q_1 - Q_2\|_\infty.$$

Hence $\widetilde{\mathbf{B}}$ is indeed a $\widetilde{\gamma}$-contraction under the supremum norm.

Composition into $(\widetilde{\mathbf{B}})^k$. By definition,

$$(\widetilde{\mathbf{B}})^k \;=\; \underbrace{\widetilde{\mathbf{B}} \circ \widetilde{\mathbf{B}} \circ \cdots \circ \widetilde{\mathbf{B}}}_{k \text{ times}}.$$

To show $(\widetilde{\mathbf{B}})^k$ is a γ-contraction, we proceed by induction on k:

- For $k = 1$, we have just shown $\widetilde{\mathbf{B}}$ itself contracts by factor $\widetilde{\gamma}$.
- Assume $(\widetilde{\mathbf{B}})^k$ is a $\widetilde{\gamma}^k$-contraction. Then

  $$\big\|(\widetilde{\mathbf{B}})^{k+1}Q_1 - (\widetilde{\mathbf{B}})^{k+1}Q_2\big\|_\infty \;=\; \big\|\widetilde{\mathbf{B}}\big[(\widetilde{\mathbf{B}})^k Q_1\big] - \widetilde{\mathbf{B}}\big[(\widetilde{\mathbf{B}})^k Q_2\big]\big\|_\infty \;\le\; \widetilde{\gamma}\,\|(\widetilde{\mathbf{B}})^k Q_1 - (\widetilde{\mathbf{B}})^k Q_2\|_\infty$$

  by the single-step contraction. Applying the induction hypothesis,

  $$\|(\widetilde{\mathbf{B}})^k Q_1 - (\widetilde{\mathbf{B}})^k Q_2\|_\infty \;\le\; \widetilde{\gamma}^k\,\|Q_1 - Q_2\|_\infty.$$

  Consequently,

  $$\big\|(\widetilde{\mathbf{B}})^{k+1}Q_1 \;-\; (\widetilde{\mathbf{B}})^{k+1}Q_2\big\|_\infty \;\le\; \widetilde{\gamma}\,\widetilde{\gamma}^k\,\|Q_1 - Q_2\|_\infty \;=\; \widetilde{\gamma}^{k+1}\,\|Q_1 - Q_2\|_\infty.$$

Thus by induction, $(\widetilde{\mathbf{B}})^k$ is a $\widetilde{\gamma}^k$-contraction. By definition 2 $\widetilde{\gamma}^k = \gamma$, thus $\widetilde{B}^k$ is a γ contraction under the supremum norm $\|\cdot\|_\infty$.

Conclusion. Since our "Bellman equation With hill-climbing action selection" is equivalent to $(\widetilde{\mathbf{B}})^k$, we conclude it is a γ-contraction in the sup norm. Hence, like the standard multi-Bellman operator of Carvalho et al. [2023], it converges to a unique fixed point under $0 \le \gamma < 1$ and bounded rewards. $\square$

A.4 Full Proof of Equivalence for Definition 1 and 2

Theorem 3 (Hill-Climbing Equivalence)**.** *Applying k iterations of Multi-Bellman Operator for Hill-Climbing Variant (Definition 2) is equivalent to one application of Bellman Operator for Bellman equation with hill climbing action selection (Definition 1) with an action budget of k.*

We introduce the concept of a Multi-Bellman Operator defined by Carvalho et al. [2023]. Given Bellman operator H and Q function q, a Multi-Bellman operator is defined as:

$$(\mathbf{H}^n q)(x_0, a_0) \;=\; \mathbb{E}\Big[r(x_0, a_0) \;+\; \gamma \max_{a_1 \in \mathcal{A}} \mathbb{E}\Big[r(x_1, a_1) \;+\; \gamma \max_{a_2 \in \mathcal{A}} \mathbb{E}\Big[\cdots \;+\; \gamma \max_{a_n \in \mathcal{A}} q\big(x_n, a_n\big)\Big]\Big]\Big].$$

From definition 2, we can reduce the algorithm from definition 1 into a Multi-Bellman operator as such:

$$(\widetilde{\mathbf{B}}^k Q)(s, a) \;=\; \mathbb{E}\Big[\widetilde{R}(\widetilde{s}_0, a_0) + \gamma \max_{a_1 \in \mathcal{A}} \mathbb{E}\Big[\widetilde{R}(\widetilde{s}_1, a_1) + \gamma \max_{a_2 \in \mathcal{A}} \mathbb{E}\Big[\cdots + \gamma \max_{a_{k-1} \in \mathcal{A}} Q\big(\widetilde{s}_{k-1}, a_{k-1}\big)\Big]\Big]\Big].$$

In order words, we apply Bellman Operator for Hill-Climbing Action Selection once per action until we reach state $\widetilde{s}_{k-1}$, which represents $(s, \{a_0, ..., a_k\}, k-1)$. We can then apply the actions in $\widetilde{s}_k$ to s to transition into state s'.

Proof. We proceed by showing that applying k iterations of the Bellman Operator for Hill-Climbing Variant is equivalent to one iteration of the Bellman equation with Hill-Climbing Action Selection, which is the value $R(s, \{a_0, a_1, ..., a_{k-1}\}) + \gamma V(s')$.

$$\widetilde{B}^k V(s) = \max_{a_0 \in \mathcal{A}}\Big[\widetilde{R}(\widetilde{s}_0, a_0) + \widetilde{\gamma}\Big[\max_{a_1 \in \mathcal{A}\setminus A}(\widetilde{R}(\widetilde{s}_1, a_1) + \widetilde{\gamma}\Big[... + \widetilde{\gamma} \max_{a_k \in \mathcal{A}\setminus A}(\widetilde{R}(\widetilde{s}_{k-1}, a_{k-1})) + \widetilde{\gamma} V(\widetilde{s}_k)\Big]\Big]\Big]$$

Expanding this and simplifying, we get:

$$\widetilde{B}^k V(s) = (R(s, \{a_0\}) - R(s, \varnothing)) + \widetilde{\gamma}\Big[\frac{1}{\widetilde{\gamma}}(R(s, \{a_0, a_1\}) - R(s, \{a_0\})\Big] + \widetilde{\gamma}^2\Big[\frac{1}{\widetilde{\gamma}^2}(R(s, \{a_0, a_1, a_2\}) - R(s, \{a_0, a_1\})\Big]$$

$$+ ... + \widetilde{\gamma}^{k-1}\Big[\frac{1}{\widetilde{\gamma}^{k-1}}(R(s,\{a_0,...,a_{k-1}\}) - R(s,\{a_0,...,a_{k-2}\}) + \widetilde{\gamma}^k V(\widetilde{s}_k)\Big]$$

Notice that $R(s,\{a_0\}) - \widetilde{\lambda}\frac{1}{\widetilde{\lambda}}(R(s,\{a_0\}) = 0$, $\widetilde{\lambda}\frac{1}{\widetilde{\lambda}}R(s,\{a_0,a_1\}) - \widetilde{\lambda}^2\frac{1}{\widetilde{\lambda}^2}R(s,\{a_0,a_1\}) = 0$, and so on. Thus, our $\widetilde{B}^k V(s)$ simplifies to

$$\widetilde{B}^k V(s) = -R(s,\widetilde{\varnothing}) + R(s,\{a_0,...,a_{k-1}\})$$

Since $R(s,\varnothing) = 0$ and $\widetilde{\gamma}^k = \gamma$, we ultimately arrive at

$$\widetilde{B}^k V(s) = R(s,\{a_0,...,a_{k-1}\}) + \gamma V(\widetilde{s_k})$$

However, at s_k, $t = k$ and by Definition 2 we transition our state into $\widetilde{s}' = (s', \varnothing, 0)$ which can be simplified to s'. So, with an additional application of $\widetilde{B}$, our final formulation becomes

$$\widetilde{B}^k V(s) = R(s,\{a_0,...,a_{k-1}\}) + \gamma V(s')$$

Note that by Definition 2, we have selected the exact same actions as we would have using Bellman equation with Hill-Climbing Action Selection, and the final value of $\widetilde{B}^k V(s)$ is equivalent to that of $BV(s)$. This concludes our proof. $\square$