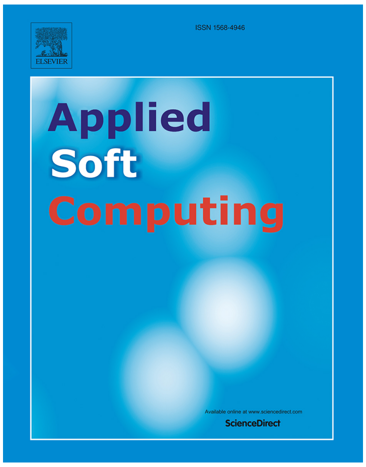

Graph-embedded reinforcement learning for dynamic pricing and advertising under network effects

Ehsan Ardjmand, Esmaeil Izadi, Ali Tavasoli,
Behnaz Moradi-Jamei, Heman Shakeri

PII:	S1568-4946(25)01369-9
DOI:	https://doi.org/10.1016/j.asoc.2025.114056
Reference:	ASOC 114056
To appear in:	*Applied Soft Computing*
Received Date:	16 June 2025
Revised Date:	4 October 2025
Accepted Date:	6 October 2025

Please cite this article as: Ardjmand E, Izadi E, Tavasoli A, Moradi-Jamei B, Shakeri H, Graph-embedded reinforcement learning for dynamic pricing and advertising under network effects, *Applied Soft Computing* (2025), doi: https://doi.org/10.1016/j.asoc.2025.114056.

Highlights

- Stochastic diffusion model links price, ads, and peer influence on networks.
- Mean-field analysis yields a reproduction threshold and trade-free equilibrium stability.
- Introduce TD3ES: RL with GCN autoencoder for joint pricing-advertising control.
- GPU simulator enables scalable training on large-scale heterogeneous graphs.
- TD3ES lifts profit on heavy-tailed networks.

Graph-Embedded Reinforcement Learning for Dynamic Pricing and Advertising under Network Effects

Ehsan Ardjmand[a,*], Esmaeil Izadi[b], Ali Tavasoli[c], Behnaz Moradi-Jamei[c], Heman Shakeri[d]

[a] *Department of Analytics and Information Systems, College of Business, Ohio University, OH, USA, 45701*
[b] *Department of Economics, Simon Fraser University, BC, Canada, V5A1S6*
[c] *Department of Mathematics & Statistics, James Madison University, Harrisonburg, VA, USA, 22807*
[d] *School of Data Science, University of Virginia, Charlottesville, VA, USA, 22904*

Abstract

Firms increasingly rely on both price discounts and advertising campaigns to shape product diffusion in socially connected markets, yet existing models rarely treat these levers jointly or account for network heterogeneity. This study develops an integrated, network-aware framework for dynamic pricing and advertising control. A stochastic compartmental model of the consumer decision-making model (CDM) is formulated on a social graph, with transition intensities modulated by price, advertising spend, and peer influence. A deterministic mean-field approximation yields closed-form expressions for a trade-free equilibrium (TFE) and a reproduction number threshold that delineates when adoption dies out versus persists. Building on this analytical core, the paper introduces twin delayed deep deterministic policy gradient with encoded state (TD3ES), a reinforcement learning (RL) controller that couples an actor-critic architecture with a graph-convolutional autoencoder, thereby compressing high-dimensional network states into a tractable latent representation. A custom GPU-accelerated simulator facilitates large-scale training. Numerical experiments on Erdős-Rényi and heavy-tailed exponential networks show that TD3ES swiftly converges to profit-maximizing joint policies and, on heterogeneous graphs, outperforms a TD3 baseline that lacks network-structural information. Error analysis reveals that the autoencoder naturally prioritizes high-degree hubs in dominant CDM compartments, explaining its superior performance. Managerially, the results demonstrate that ignoring topology can forfeit substantial revenue and that adaptive, network-aware coordination of price and advertising is both feasible and valuable. The framework thus unites rigorous diffusion theory with scalable learning, offering a practical tool for data-driven marketing in connected consumer ecosystems.

Keywords: Reinforcement learning; Graph neural networks; Network externalities; Consumer decision-making; Dynamic pricing

1. Introduction

From the initial moment when a need is ignited within a consumer's consciousness to the thrilling culmination of pressing the "Buy Now" button, there unfolds a complex array of stages and subtleties that intrigue researchers and are crucial in determining the success of businesses. In the marketing literature, this journey is traditionally conceptualized as a sequential five-step process, referred to as the consumer decision-making model (CDM) [1]. It begins with the recognition of a need, where the consumer perceives a gap between their current state and a desired state. Consumers then enter into the information search phase, where they seek out data and insights about potential solutions. Following this, the evaluation of

[*] Corresponding authors

Email addresses: ardjmand@ohio.edu (Ehsan Ardjmand), esmaeil_izadi@sfu.ca (Esmaeil Izadi), at9kf@virginia.edu (Ali Tavasoli), moradibx@jmu.edu (Behnaz Moradi-Jamei), hs9hd@virginia.edu (Heman Shakeri) alternatives stage occurs, during which the consumer compares different products or services to determine which best meets their needs. The next step is the purchase decision, where the consumer considers their budget constraint and makes a final decision on what to obtain. Finally, the process concludes with the post-purchase evaluation, where the consumer reflects on their decision and the realized utility from the purchase. There is a range of factors in each of these stages that contribute to the innate complexity of the consumer decision-making process.

Businesses frequently engage in strategies to influence the decision-making processes of consumers with the goal of boosting sales. Two key toolkit available to businesses, as profit maximizing entities, are advertising, and pricing strategies. Researchers in marketing and economics have now established that these strategies are both amenable to network effects in various forms and extents [2].

Advertising plays a critical role in this mix. Research has established that a firm's advertising strategy significantly influences consumer purchasing behavior by deploying its persuading appeal, keeping consumers informed about products or services, or a combination of both [3]. It is clear that an advertising strategy can engage with consumers at different stages of the decision-making process, from enticing the need, presenting options in consumers' information search, and presenting alternatives to even the post purchase and evaluation phases, cultivating a positive brand image becomes critical [4].

Pricing is another pivotal factor that impacts consumer decision-making. It is the key signal consumers use to find a match between their budget and value they assign to a product. In the post-purchase stage of evaluation, consumers typically assess the price relative to the quality and its anticipated benefits. This evaluation, in turn, influences their repeated purchase decision. It also impacts the information they spread in the network, affecting other consumers' purchasing behavior.

This is where the two main strategy of advertising and pricing come into interaction with each other: through consumer behavior realized over a network. An example of this interaction is when a persuasion-oriented ad raises the product's perceived benefit; if a short-lived discount simultaneously lowers its perceived cost, those first recipients cross their personal cost-benefit thresholds, buy, and broadcast their satisfaction, nudging friends to follow. Without the discount, the same ad may await peer testimonials before demand spreads; without the ad, a discount may draw only bargain hunters whose lukewarm word-of-mouth (WOM) stalls diffusion. Thus, advertising shifts the benefit side of the ledger, pricing shifts the cost side, and their combined signal propagates along network links to amplify or dampen overall adoption [5].

Additionally, businesses leverage network effects to shape consumer buying patterns. This phenomenon, largely propelled by WOM communication and social interactions, plays a crucial role in consumer choices by offering trustworthy information, personal endorsements, and reviews [6]. WOM can be visualized as a network comprising individuals (nodes) and their relationships (links). The influence that peer interactions exert on buying habits is referred to as network externalities. These externalities emerge from the collective actions of consumers (global externality) [7] or from the immediate circle of an individual's acquaintances (local externality) [8]. Research suggests that social influences drive 25-50% of purchasing decisions [9], potentially increasing the likelihood of making a purchase by up to 60% [10]. Beyond advertising, network effects have been shown to impact the stages of information search and evaluation within the CDM by altering perceptions of value, quality, and trust. This social influence can lead to a reevaluation of alternatives and ultimately affect purchasing decisions. Pricing and advertisement strategies can either encourage or dampen network effects and WOM influence [11]. Therefore, pricing and advertising strategies should be carefully crafted to enhance the positive effects of network externalities on sales. Figure 1 depicts the five stages of CDM along with the influence of network effects, advertisements, and pricing on it.

This research introduces a stochastic model of the CDM process that incorporates the effects of pricing, advertising resources, and network effects. The model posits that a firm can manipulate the price and advertising spending to strategically influence the CDM process and optimize its profits. It suggests that the calibration of pricing and the allocation of advertising resources impact not only the consumer decision-making process but also amplify the strength of network effects, thereby increasing demand and profit. The network effect within this model is conceptualized using a stochastic propagation process akin to models found in the epidemiological literature. Since analyzing such a stochastic model can become intractable for large networks, a deterministic mean-field approximation is utilized [12]. Then, the mean-field model is used to investigate the conditions under which the consumer network will reach an equilibrium over time.

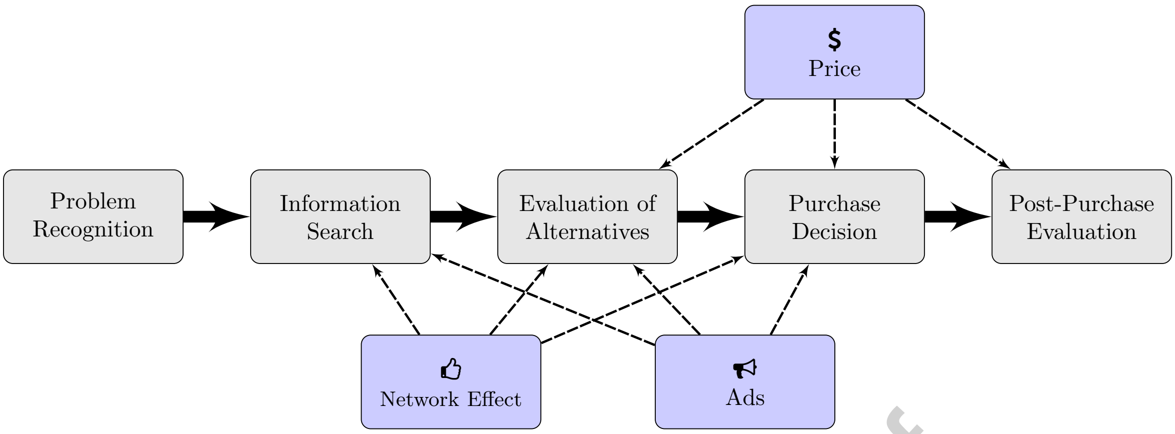

Figure 1: Consumer decision-making model (CDM) along with the influence of network effects, advertisements, and pricing on its various stages.

Consumers move through the CDM in tandem with—and often because of—their interactions inside a socially connected network. Together, these two dimensions create a fluid, tightly coupled system: a shift in one buyer's decision stage can ripple outward, reshaping the purchase likelihood of neighbouring nodes and, by extension, the aggregate market state. Because both the network topology and the distribution of decision stages evolve continuously, any fixed schedule for pricing or advertising is liable to drift out of alignment with reality. In fact, the system's high dimensionality and non-linearity render a closed-form, model-based control rule impractical. To overcome this barrier, a TD3ES RL method is introduced that observes the network at regular intervals and revises both price points and advertising outlays accordingly. The agent treats the current distribution of consumers across CDM stages as well as observed peer influences as its state, then selects the combination of price and promotional intensity expected to maximize cumulative profit. As fresh data arrive, the policy is updated, allowing the firm to translate real-time market signals into timely, evidence-driven adjustments. In short, the proposed RL framework provides a tractable, adaptive alternative to analytical control, ensuring that marketing and pricing decisions remain continuously aligned with the evolving structure and consumer network.

To incorporate both network topology and consumer state into the control policy, the TD3ES augments the standard twin delayed deep deterministic policy gradient (TD3) framework with a graph convolutional network (GCN) encoder [13]. The encoder processes the adjacency matrix alongside node-level attributes such as each consumer's current CDM stage, and compresses this high-dimensional information into a single, fixed-length vector. This embedding provides the actor–critic pair with a compact yet expressive summary of the market, enabling TD3ES to reason about peer influence patterns without an explosion in state-space dimensionality.

TD3ES proves especially valuable on heterogeneous networks whose degree distributions vary sharply from one node to the next. In such settings, densely connected hubs exert disproportionate influence over purchase cascades. The GCN's message-passing layers retain these local and global contrasts, allowing the learned policy to differentiate between hub-centric leverage points and fringe consumers. As a result, TD3ES discovers pricing and advertising strategies that conventional TD3, lacking any structural awareness, cannot match, particularly when the market graph exhibits heavy-tailed or skewed connectivity patterns.

This study seeks to examine two fundamental questions: (i) how network effects influence optimal pricing and advertising strategies, and (ii) how dynamic strategies evolve in response to the stochastic evolution of networks. The proposed analytical model and the designed TD3ES algorithm will address these questions through incorporating key features such as the consumer network topology and learning the complex interactions between network effects, pricing, and advertising. The contributions of the presented study are as follows:

- *Stochastic CDM model*: This study formulates the consumer decision-making model (CDM) as a stochastic process on a social network, jointly driven by price signals, advertising exposure, and peer influence.

- *Mean-field deterministic approximation and equilibrium analysis*: A tractable mean-field representation of the high-dimensional CDM stochastic model is derived that enables analytical investigation on large-scale networks. Additionally, conditions for the existence and stability of market equilibria are established.
- *Adaptive Control via TD3ES*: This study introduces TD3ES, a novel actor-critic algorithm that augments TD3 with a GCN encoder, compressing evolving network states and CDM stage distributions into a compact embedding. This embedding provides a data-driven control policy that jointly optimizes dynamic pricing and advertising budgets in real time. TD3ES exploits degree heterogeneity, identifying high-leverage hubs and tailoring interventions accordingly; benchmark TD3 (without structural encoding) fails to achieve comparable performance on heavy-tailed or skewed topologies.

The remainder of this article is organised as follows. Section 2 positions the study within the intersecting literatures on network-based consumer behavior, diffusion dynamics, and RL. Section 3 formalizes the joint pricing-advertising problem by introducing the CDM stochastic process on a consumer's graph, while Section 4 derives its mean-field approximation and establishes equilibrium and stability results. Section 5 details the proposed TD3ES framework, including the graph-convolutional state encoder and the GPU-accelerated simulation environment. Comprehensive numerical experiments are presented in Section 6, and their managerial and methodological implications are analyzed in Section 7. Finally, Section 8 summarizes the principal contributions, discusses limitations, and outlines avenues for future research.

2. Literature Review

This study investigates a firm's pricing and advertising strategies in a networked market where consumers' purchase decisions, captured by the CDM framework, both shape and are shaped by peer interactions and network connections. A twin delayed deep deterministic policy gradient with encoded state (TD3ES) algorithm is introduced that observes the aggregate state of the consumer network and adaptively sets prices and advertising expenditures to maximize profit. Drawing on insights from computational epidemiology, the CDM with network effects is modeled as a diffusion process on a graph. Accordingly, the analysis bridges three strands of scholarship: (i) economic and marketing research on consumer behavior, pricing, and resource allocation under network externalities, (ii) epidemiological studies of diffusion dynamics over networks, and (iii) reinforcement learning (RL) approaches for controlling dynamic diffusion processes.

2.1. Consumer Behavior, Pricing, and Resource Allocation Under Network Externalities

Interest in how networks shape consumer behavior dates back to the landmark contributions of Farrell and Saloner [14] and of Katz and Shapiro [7], who analyzed global externalities using complete homogeneous graph models. Their insights triggered a rich literature exploring how finer network structures and local interactions influence pricing decisions. When interactions are heterogeneous, local externalities emerge, making it natural to tailor prices to nodal importance, typically measured by degree or centrality measures such as Katz–Bonacich centrality [15]. Under such conditions, both the network's architecture and the influence of highly central nodes become key determinants of the profit maximizing price schedule [16]. Network effects, however, are not invariably beneficial; in some settings they can dampen demand [17]. Moreover, local externalities can assume intricate forms. Non-linear interaction terms, for instance, markedly reshape optimal prices, consumer preferences [18], and allocation or incentive schemes [19].

There exists substantial empirical and theoretical evidence demonstrating both the prevalence of network externalities and their pronounced influence on consumer behavior [20]. Because purchasing decisions are interdependent, the resulting network effect has become a cornerstone of modern marketing strategy. Firms commonly harness this effect by allocating promotional resources or by adopting discriminatory pricing schemes that encourage individuals to share information, thereby magnifying product awareness and accelerating adoption.

A prototypical resource–allocation problem in this context is the target set selection (TSS), in which the decision maker must identify, subject to a strict budget constraint, a small subset of highly influential consumers whose incentivization maximizes the eventual diffusion of information through the underlying social network [21]. A widely studied instantiation of TSS is the influence–and–exploit paradigm: an initial cohort of central agents receives the product for free, after which a revenue maximizing price is charged to the remaining population [22]. Other variants emphasize price–based incentives, offering the selected seed nodes favorable discounts rather than complimentary products, while retaining the same diffusion objective [15]. In essence, all of these selective approaches exploit heterogeneity in network position and employ price discrimination to trigger a cascade of adoptions.

In contrast, a parallel line of research considers uniform allocation schemes, wherein an identical level of resource or a single price is extended to every consumer, with the collective goal of maximizing information propagation. Although such uniform strategies have attracted less attention in the marketing and economics literatures, they are extensively analyzed in epidemiology. There, the analogous task is to uniformly reduce infection risk across all individuals (or within a critical subgraph) of a contact network [23, 24].

The TSS problem is conceptually intertwined with network-based pricing problems, in which a firm deploys discriminatory-pricing strategies to stimulate both product awareness and sales. Prior research on network pricing is notably diverse, examining a wide range of determinants: the spatial reach of externalities (local vs. global) [15, 25, 26]; network topology and centrality patterns [27]; competitive market structure [28]; dynamic (time-varying) price trajectories [11, 29]; and differential or nonlinear pricing schemes [18, 26, 28]. Despite this breadth, two critical gaps remain:

1. **Behavioral Nuances.** Most existing models treat purchase as a single binary event, thereby overlooking the distinct pre-purchase and post-purchase stages during which network effects, pricing, and advertising exert different influences on consumer behaviour. Capturing these stage specific interactions requires a compartmental representation akin to models in modern epidemiology, where individuals transition between well-defined states over time.

2. **Joint Resource Allocation.** The majority of prior studies concentrate on pricing decisions alone and neglect advertising intensity as a co-equal lever. In practice, however, firms must simultaneously determine both price and advertising spend, allocating finite resources across these channels to maximize diffusion and revenue.

The present study addresses these shortcomings by introducing a compartmentalized consumer dynamics model based on the well-established CDM framework. Within this framework, price and advertising are treated as joint control variables, enabling the systematic exploration of how different allocation policies drive network diffusion, sales growth, and long-run profitability.

2.2. Diffusion Dynamics Over Networks

Consumer behavior under network externalities can be modeled as a stochastic information–propagation process, a formalism that parallels classical epidemic models. In such models, each individual's probability of adopting (or recommending) a product evolves according to random interactions with neighbors, making the system's future trajectories intrinsically unpredictable even for seemingly simple network structures [30]. This inherent intractability has motivated a large body of work that replaces the exact, high–dimensional stochastic description with a deterministic mean–field approximation. Under the mean–field hypothesis, the influence of each neighbor is substituted by an average effect, reducing the analysis to a system of coupled ordinary differential equations that track the expected state of each consumer class over time [12]. The resulting deterministic system often admits closed-form equilibria and stability conditions, enabling researchers to derive sharp structural insights such as thresholds for widespread adoption or conditions under which diffusion dies out in networks of arbitrary size [31]. Consequently, mean–field models have become an indispensable tool for describing information cascades in epidemiology and the broader study of spreading processes on complex networks.

Consistent with prevailing methodological practice, the present study represents the CDM as a compartmental stochastic process unfolding on a social network and subsequently derives a mean–field approximation of that process. The stochastic formulation is employed to generate Monte-Carlo trajectories that faithfully reproduce individual-level adoption events, whereas the mean–field system furnishes closed-form equilibria and stability conditions that guide both analytical inference and the construction of test scenarios.

Crucially, accurate simulation of the stochastic spread process constitutes the *environment* with which a reinforcement learning (RL) agent must interact repeatedly. In large-scale networks the computational burden becomes prohibitive: state-of-the-art simulators for spreading processes, such as generalized epidemic modeling framework (GEMF) [32] and its recent fast variant [33], still exhibit lengthy CPU runtimes that impede iterative RL training. To overcome this bottleneck, the present work introduces a graphics processing unit (GPU)-accelerated implementation that parallelizes all event-update operations and leverages the sparse-tensor data structures provided by PyTorch Geometric [34]. This computational advance is a prerequisite for scalable, data-efficient RL algorithms that seek to optimize price–advertising policies in realistic, high-dimensional consumer networks.

2.3. Reinforcement Learning in Diffusion Processes

A principal application of reinforcement learning (RL) in spreading processes is the *control* of diffusion dynamics, either by promoting beneficial cascades (e.g., product adoption) or by containing harmful ones (e.g., epidemics or misinformation) [35]. Extensive empirical evidence demonstrates the utility of RL for suppressing undesirable phenomena on networks, including infectious diseases [36]. In each of these studies the environment is formalized as a graph whose nodes represent interacting entities (e.g., individuals or regions) where spread dynamics unfold along the edges of this graph.

Given this inherently networked structure, graph neural networks (GNNs) arises as a natural policy-learning architecture within the RL framework. GNNs can extract high-level representations that encode both topological context (e.g., node centrality, community membership) and dynamic state information (e.g., infection probability, adoption intensity). When integrated with RL, a GNN-based agent can learn to map these rich representations to fine-grained intervention policies, such as targeted vaccination, content suppression, or price–advertising adjustments that optimize a long-term objective. This synergy between RL and GNN architectures has achieved promising results well beyond diffusion control [37]. These successes underscore the broader potential of GNN-augmented RL methods for solving complex, graph-structured decision problems.

Over the past several years, RL methods have begun to find applications in advertising, pricing, and diffusion-related tasks [38, 39]. Despite this progress, the networked nature of the underlying systems, such as the interconnections among consumers, has often been overlooked. While RL has been extensively employed to contain and mitigate harmful diffusion processes (e.g., the spread of epidemics or rumors), its use in fostering beneficial diffusion, as desired in viral marketing and related contexts, has remained relatively limited. Only a small number of recent studies have investigated RL-driven strategies for influence maximization in consumer networks [40, 41]. However, these works primarily emphasize the selection of an initial seed set of consumers and largely neglect the critical managerial levers of dynamic pricing and advertising allocation. To the best of the authors' knowledge, no prior research has jointly optimized these two levers while explicitly accounting for the multi-stage decision-making processes of consumers, as formalized through the CDM framework.

The present research addresses this gap by introducing a twin delayed deep deterministic policy gradient with encoded state (TD3ES) RL framework that employs a GNN encoder to represent the network's evolving state. At each decision epoch the agent selects one uniform price and advertising budget, eschewing node-specific interventions typical of influence-maximization studies. Extensive numerical experiments benchmark the proposed TD3ES against an actor–critic baseline with a simplified state definition and delineate the conditions under which the new method delivers significant improvements in profit.

In this context, the study makes three principal contributions:

1. It formulates an RL framework that jointly optimizes dynamic pricing and advertising spend in a consumer network.

2. It integrates a GNN state encoder within the proposed RL algorithm, yielding a scalable controller suited to large, real-world graphs.

3. It provides a systematic empirical evaluation that clarifies the conditions under which uniform, network-aware pricing–advertising policies outperform established RL baselines.

3. Problem Statement

As was discussed in Section 1 and visually represented in Figure 1, a consumer moves through various stages before and after the purchase of a commodity. Furthermore, consumers may either opt for repurchasing a commodity or disengage from the process, typically influenced by factors such as pricing or insufficient exposure to the product, exemplified by low advertisement levels or a limited network effect. To capture the interactions between pricing, advertisement, and the network effect, the CDM framework shown in Figure 1 can be conceptualized as a Markov process. In such Markov model depicted in Figure 2 (left), each state corresponds to a phase within the CDM framework, with transition rates contingent upon the interplay of price dynamics, advertising efforts, and the magnitude of the network effect. In this study, the network effect is considered as a local externality that emanates from a consumer's contact network, encompassing both online and physical connections. The contact structure within a consumer population is modeled as an undirected graph denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, N\}$ represents the set of individuals or consumers (i.e., vertices), and $\mathcal{E}$ signifies the set of connections (i.e., edges). For an individual $i \in \mathcal{V}$, the collection of i's neighbors (i.e., other individuals in contact with i) is denoted as $\mathcal{N}_i$. Figure 2 (right) illustrates a network comprising four consumers, each in a distinct state.

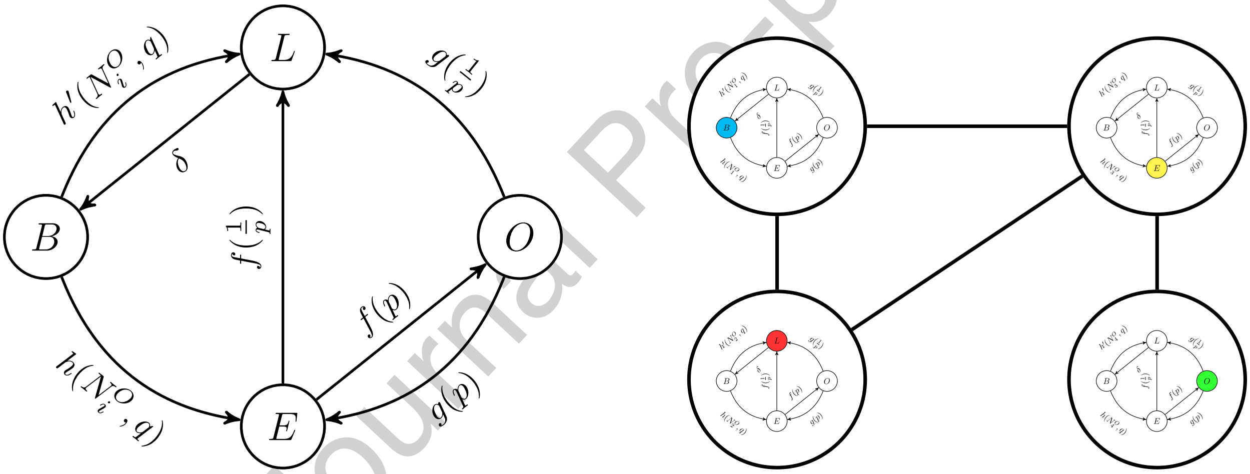

Figure 2: Left: Markov process representation of the CDM framework. Right: A contact network where each node (i.e., consumer) behaves according to the CDM's Markov process. Shadowed compartments depict the state of each node in the contact network.

In Figure 2 (left), a consumer can be in any four states of potential buyer (B), exposed (E), owner (O), or leaving (L). State B signifies an individual who has identified a need or issue and is actively seeking a product to address it. This potential buyer may transition to the exposed state (E) upon encountering the commodity offered by the firm, a progression facilitated by the consumer's information search process and influenced by the firm's investment in advertising and the network effects. The transition rate from state B to state E is expressed through a multiplicative advertisement response function in the form of $h(N_i^O, q) = (\alpha_1 q^{\alpha_2} + \beta) N_i^O$, where N_i^O represents the number of neighbors of consumer i within her contact network who possess the product, q denotes the level of advertisement spending by the firm, and α_1, α_2, and β denote the coefficients of the function. This multiplicative functional form, adapted from prior research [42], is employed here for its established utility. It is important to highlight that within the function $h(N_i^O, q)$, the parameter denoting the level of advertisement, q, influences sales by amplifying the impact of network externality. Although advertisements can boost sales indirectly by fostering brand awareness or through other intermediary mechanisms, a plausible assumption is that they primarily augment sales by intensifying the network effect within a social network environment. When q is set to zero, the scenario simplifies to one where sales are solely influenced by the network effect.

A potential buyer may fail to encounter the commodity due to inadequate advertisement exposure or weak network effects, leading to their departure from the purchasing process (state L). This transition to the leaving state L can be represented using a similar functional form, $h'(N_i^O, q) = \left[(\alpha_1 q^{\alpha_2} + \beta)N_i^O + \epsilon\right]^{-1}$, where the inputs are reciprocals to reflect a diminished probability of departure when N_i^O and q are greater, and conversely, an increased probability when they are smaller. The constant ϵ represents a minute value utilized to prevent division by zero when computing reciprocals. Furthermore, consumers utilizing different products may transition into potential buyers and enter state B. This shift typically occurs when consumers are dissatisfied with their current products. Such transitions are modeled using a fixed rate denoted by δ.

Following exposure to the product, a consumer proceeds to assess available alternatives and make a purchasing decision, potentially resulting in the consumer acquiring ownership of the product (state O). The transition from state E to O is characterized by the widely utilized negative exponential function [43–45] $f(p) = d_1 e^{-d_2 p}$, where p denotes the price, d_1 represents the maximum transition rate from E to O (i.e., the transition rate when the product is free), and d_2 signifies the demand elasticity. An individual who has been exposed to the product may also exit the process, with a transition rate represented by $f(\frac{1}{p})$, which escalates as the price increases.

Upon purchasing the product (state O) and engaging with it, a consumer proceeds to assess the purchase, determining whether it justifies the price paid. Assuming consistent quality, a lower price augments the likelihood of the consumer contemplating repurchase. Put differently, a reduced price sustains the consumer's exposure to the product, thereby heightening the probability of subsequent repurchase. Conforming to the definition of $f(p)$, the transition rate from O to E is articulated as the function $g(p) = d_1 e^{-d_3 p}$. Moreover, if the consumer's post-purchase evaluation yields dissatisfaction with the product, the consumer will exit the process with a transition rate denoted by $g(\frac{1}{p})$.

In this investigation, a firm manufactures a product whose consumers' behavior adheres to the conceptual framework of the CDM (depicted in Figure 1) and is formalized through a Markov process (illustrated in Figure 2). The firm retains the flexibility to adjust the product's price (p) and advertisement expenditure (q) to optimize its profitability. It is noteworthy that the firm operates within an environment where consumers possess the option to discontinue the purchasing process (state L). The behavioral dynamics of each consumer i are contingent upon her contact network neighbors ($\mathcal{N}_i$) and their respective states. Notably, the likelihood of a consumer transitioning to product ownership is positively correlated with the prevalence of product owners within her network. This dynamic evolves over time as consumers transition between states and the structure of their networks undergoes corresponding changes.

In this study, a reinforcement learning (RL) approach, namely, twin delayed deep deterministic policy gradient with encoded state (TD3ES) is employed. TD3ES allows for periodic adjustments to price and advertisement spending with the aim of maximizing profitability over the problem's horizon. Within the TD3ES framework, the timeline is discretized into sequential time steps t. At the outset of each time step t, the TD3ES agent determines the values of p and q, after which the consumers' states evolve over the ensuing time step before the subsequent decision-making opportunity arises. Figure 3 illustrates this rolling interaction cycle: each epoch commences with the joint setting of p and q, followed by the propagation of information and adoption throughout the network until the subsequent decision point is reached. This procedure enables the agent to learn an adaptive pricing-advertising policy that reacts to real-time changes in consumer behavior while accounting for the long-term consequences of each intervention.

During each evolution phase, the product's adoption propagates stochastically through the contact network, and every consumer transitions between behavioral compartments according to the continuous–time Markov process depicted in Figure 2. Accurately projecting these state trajectories requires a simulation engine that is computationally efficient. The present study therefore introduces a GPU-enabled event–driven simulator that preserves the statistical fidelity of Gillespie's Stochastic Simulation Algorithm, [46, 47] while substantially reducing runtime relative to existing CPU-bound implementations [32, 33].

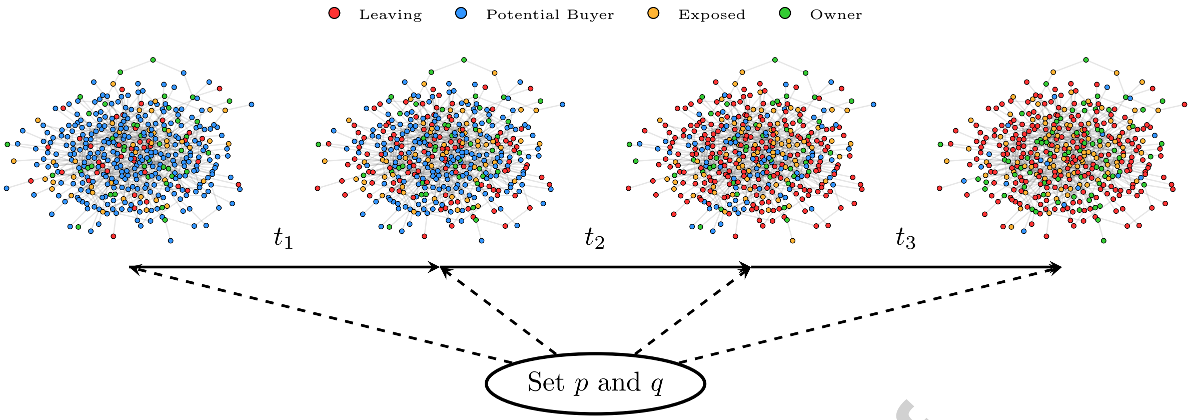

Figure 3: Decision–making timeline of the TD3ES agent. At the start of each decision epoch t, a uniform price p and an advertising expenditure q are selected. During the remainder of the interval $(t, t + 1]$, the consumer network evolves according to the chosen controls, producing immediate revenue and an updated state for the next decision point.

4. Model Development

In this section, a stochastic model capturing the diffusion of product adoption under the CDM framework, as described in Section 3, is first developed. Building on this formulation, a mean-field approximation is derived, yielding a deterministic system of differential equations that approximates the underlying stochastic dynamics. The resulting system admits a TFE, i.e., a point when no consumer transitions to state O, indicating a scenario where no sales occur. Conversely, an endemic equilibrium point occurs when a consistent non-zero proportion of consumers persistently remain in state O. Investigating the TFE provides a ground for testing pricing and advertisement strategies. The existence and local stability of the TFE are established by analyzing the associated Jacobian matrix and identifying parameter ranges that ensure all eigenvalues possess negative real parts. These theoretical insights provide a rigorous foundation for constructing a suite of numerical test problems used in later sections to benchmark solution methods and to illustrate the practical implications of the model parameters on adoption dynamics.

4.1. Stochastic Model

Let $b_i \in [0, 1]$ represent the probability of consumer $i \in \{1, 2, \ldots, N\}$ being in state B, $e_i \in [0, 1]$ denote the probability of being in state E, $o_i \in [0, 1]$ signify the probability of being in state O, and $l_i \in [0, 1]$ represent the probability of being in state L, satisfying the constraint $b_i + e_i + o_i + l_i = 1$. Define $X_i \in \{B, E, O, L\}$ as a random variable corresponding to the state of consumer i, and let X_i^t denote the value of X_i at time t. The diffusion dynamics of the Markov model depicted in Figure 2 can be expressed through the following stochastic equations:

$$
\begin{aligned}
&\Pr\left(X_i^{t+\Delta t}=B|X_i^t=L,\boldsymbol{X}^t\right)=\delta\Delta t+o\left(\Delta t\right)\\
&\Pr\left(X_i^{t+\Delta t}=E|X_i^t=B,\boldsymbol{X}^t\right)=\Delta t\left(\alpha_1 q^{\alpha_2}+\beta\right)\sum_{j\in\mathcal{N}_i}1_{\{X_j^t=O\}}+o\left(\Delta t\right)\\
&\Pr\left(X_i^{t+\Delta t}=E|X_i^t=O,\boldsymbol{X}^t\right)=d_1\exp\left(-d_3 p\right)\Delta t+o\left(\Delta t\right)\\
&\Pr\left(X_i^{t+\Delta t}=O|X_i^t=E,\boldsymbol{X}^t\right)=d_1\exp\left(-d_2 p\right)\Delta t+o\left(\Delta t\right)\\
&\Pr\left(X_i^{t+\Delta t}=L|X_i^t=O,\boldsymbol{X}^t\right)=d_1\exp\left(\frac{-d_3}{p}\right)\Delta t+o\left(\Delta t\right)\\
&\Pr\left(X_i^{t+\Delta t}=L|X_i^t=E,\boldsymbol{X}^t\right)=d_1\exp\left(\frac{-d_2}{p}\right)\Delta t+o\left(\Delta t\right)\\
&\Pr\left(X_i^{t+\Delta t}=L|X_i^t=B,\boldsymbol{X}^t\right)=\Delta t\left(\left(\alpha_1 q^{\alpha_2}+\beta\right)\sum_{j\in\mathcal{N}_i}1_{\{X_j^t=O\}}+\epsilon\right)^{-1}+o\left(\Delta t\right)
\end{aligned}
\tag{1}
$$

where $\Pr(.)$ represents probability, $\boldsymbol{X}^t$ denotes the joint state of the consumers' network at time t, $\Delta t>0$ represents a small time step, $\mathcal{N}_i$ is the set of neighbors of consumer i, and $1_{\{\mathcal{X}\}}$ stands for the indicator function. The $o\left(\Delta t\right)$ terms within the equations denote higher-order infinitesimal terms, where $\lim\limits_{\Delta t\to 0}\frac{o(\Delta t)}{\Delta t}=0$. These terms encapsulate higher-order effects that diminish as Δt approaches zero. The equations provide transition probabilities between states from time t to $t+\Delta t$ based on the current states of node i and its neighbors.

It is worth emphasizing that, although the notation in (1) resembles the transition kernel of a Markov decision process (MDP), the present formulation is instead a continuous-time Markov jump process (MJP) defined over a consumer network. In this setting, the conditional probabilities are expressed as functions of the joint state vector $\boldsymbol{X}^t$, rather than an externally chosen action, meaning that transitions are driven endogenously by network interactions. This construction, however, still satisfies the Markov property, since the probability of future states depends only on the present configuration of consumers and control variables, not on the full history. When integrated into the RL framework, the MJP naturally induces an MDP representation, where the state is given by the network-wide configuration (or its graph-encoded abstraction), the actions correspond to pricing and advertising levels (p,q), the transitions follow directly from the stochastic CDM dynamics, and the reward is defined by per-step profit.

4.2. Mean-Field Model

Employing a mean-field approximation, the Markov process (1) can be approximated as the following coupled nonlinear differential equations:

$$
\begin{aligned}
\dot{b}_i&=\delta l_i-b_i\left[\left(\alpha_1 q^{\alpha_2}+\beta\right)\sum_{j\in\mathcal{N}_i}a_{ij}o_j\right]-b_i\left[\left(\alpha_1 q^{\alpha_2}+\beta\right)\sum_{j\in\mathcal{N}_i}a_{ij}o_j+\epsilon\right]^{-1}\\
\dot{e}_i&=b_i\left[\left(\alpha_1 q^{\alpha_2}+\beta\right)\sum_{j\in\mathcal{N}_i}a_{ij}o_j\right]+o_i d_1\exp\left(-d_3 p\right)-e_i\left(d_1\exp\left(-d_2 p\right)+d_1\exp\left(\frac{-d_2}{p}\right)\right)\\
\dot{o}_i&=e_i d_1\exp\left(-d_2 p\right)-o_i\left(d_1\exp\left(-d_3 p\right)+d_1\exp\left(\frac{-d_3}{p}\right)\right)\\
\dot{l}_i&=b_i\left[\left(\alpha_1 q^{\alpha_2}+\beta\right)\sum_{j\in\mathcal{N}_i}a_{ij}o_j+\epsilon\right]^{-1}+e_i d_1\exp\left(\frac{-d_2}{p}\right)+o_i d_1\exp\left(\frac{-d_3}{p}\right)-\delta l_i
\end{aligned}
\tag{2}
$$

To express (2) in a more concise form, the following definitions are adopted:

$$
\begin{aligned}
&B = \operatorname{diag}([b_i]) = \operatorname{diag}(b), \quad E = \operatorname{diag}([e_i]) = \operatorname{diag}(e), \quad O = \operatorname{diag}([o_i]) = \operatorname{diag}(o),\\
&L = \operatorname{diag}([l_i]) = \operatorname{diag}(l), \quad \gamma_1 = \alpha_1 q^{\alpha_2} + \beta, \quad \gamma_2 = d_1 e^{-d_2 p},\\
&\gamma_3 = d_1 e^{-d_3 p}, \quad \gamma_4 = d_1 e^{-d_2/p}, \quad \gamma_5 = d_1 e^{-d_3/p}
\end{aligned}
\tag{3}
$$

By substituting $b_i = 1 - e_i - o_i - l_i$ into (2), the original model (2) is reduced to a form involving three variables e, o, and l:

$$
\begin{aligned}
\dot{e} &= \gamma_1 (I - E - O - L) A O - (\gamma_2 + \gamma_4) e + \gamma_3 o\\
\dot{o} &= \gamma_2 e - (\gamma_3 + \gamma_5) o\\
\dot{l} &= (I - E - O - L)(\gamma_1 A o + \epsilon I)^{-1} + \gamma_4 e + \gamma_5 o - \delta l
\end{aligned}
\tag{4}
$$

where I is the identity matrix of the appropriate dimension and A is the adjacency matrix of the consumers' network.

Model (4) furnishes a mean-field representation of the stochastic product adoption process defined in (1). Figure 4 (left) juxtaposes the aggregate adoption trajectories produced by the two formulations on a representative random geometric network. The deterministic mean-field curves align closely with the realizations of the stochastic system, indicating that the approximation captures the dominant dynamics. Figure 4 (right) offers a complementary view by plotting the marginal probabilities of occupying each compartment B, O, L, or E for all individuals as functions of time. The simulation procedure used to generate sample paths of model (1) is detailed in Section 5.

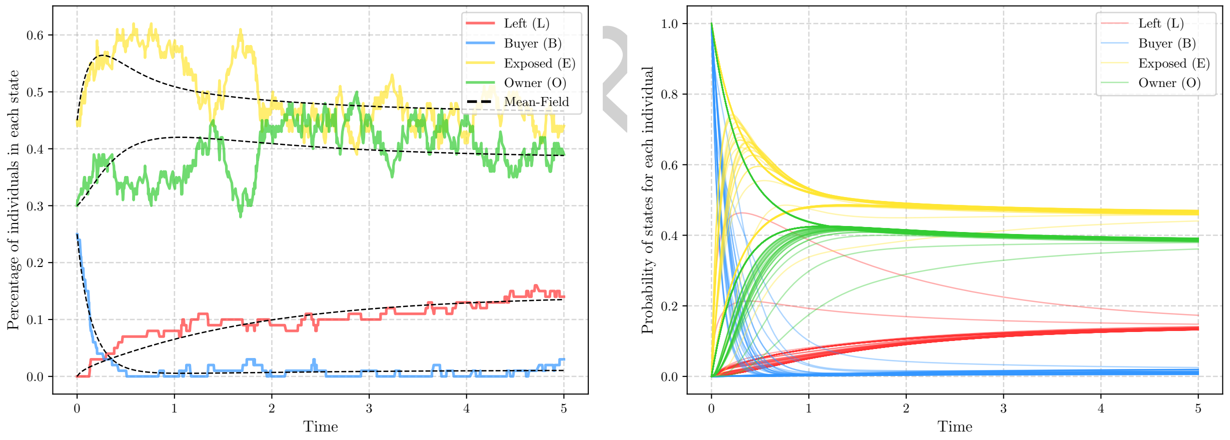

Figure 4: Left: Percentage of individuals in L, B, E, and O states over time. Right: Probability of individuals being in each state over time. Visualizations are for a random geometric contact network with 100 individuals and parameters $\alpha_1 = 1$, $\alpha_2 = 0.5$, $\beta = 1$, $\delta = 0.5$, $d_1 = 2$, $d_2 = 1$, $d_3 = 0.5$, $p = 0.2$, and $q = 10$. Initially, approximately 30% of the nodes are in the state B, 40% are in state E and the rest are in the state O.

4.3. Trade-Free Equilibrium (TFE) and Stability

In this section, the TFE point of the model (4) is investigated.

Proposition 1 (Existence and Uniqueness of the TFE)**.** *Consider the system (4). A trade-free equilibrium (TFE) is defined as an equilibrium point where $e^* = 0$ and $o^* = 0$. Under the assumptions $d_1 > 0$, $\delta > 0$, $\epsilon > 0$, and $p > 0$, there exists a unique TFE given by*

$$
e^* = 0, \quad o^* = 0, \quad l^* = \frac{1}{1 + \delta\,\epsilon} I, \quad b^* = \frac{\delta\,\epsilon}{1 + \delta\,\epsilon} I. \tag{5}
$$

Proof. Setting $\dot{e} = 0$ and $\dot{o} = 0$ in (4) forces $e^* = 0$ and $o^* = 0$ to satisfy those equations identically. Then, from the third equation $\dot{l} = 0$ with $e^* = 0$ and $o^* = 0$, one obtains

$$(I - l^*)\,(\epsilon I)^{-1} - \delta\, l^* = 0 \quad \Longrightarrow \quad \frac{1}{\epsilon}(I - l^*) = \delta\, l^* \quad \Longrightarrow \quad I = l^* + \delta\,\epsilon\, l^* = (1 + \delta\,\epsilon)\, l^*.$$

Hence

$$l^* = \frac{1}{1 + \delta\,\epsilon}\, I,$$

and thus

$$b^* = I - e^* - o^* - l^* = \frac{\delta\,\epsilon}{1 + \delta\,\epsilon}\, I.$$

No other solutions satisfy the requirement $e^* = o^* = 0$. Therefore, the TFE in (5) is unique. □

Theorem 1 (Local Stability of the TFE)**.** *Let* $x = \begin{bmatrix} e \\ o \\ l \end{bmatrix}$ *denote the state of* (4) *and consider the unique TFE* x^* *from Proposition 1. Let* J^* *be the Jacobian of* (4) *evaluated at* x^**. Suppose the adjacency matrix* A *is symmetric (or diagonalizable) with eigenvalues* $\lambda(A)$*. Then:*

(1) There are always N *negative real eigenvalues of* J^**, each equal to*

$$-\frac{1 + \delta\,\epsilon}{\epsilon} < 0.$$

(2) The remaining $2N$ *eigenvalues of* J^* *arise from a coupled* 2×2 *block whose characteristic equation yields the condition*

$$\frac{\lambda^2}{\gamma_2} + \left(1 + \frac{\gamma_4}{\gamma_2}\right)\lambda + \frac{(1 + \gamma_4)(\gamma_3 + \gamma_5)}{\gamma_2} + \gamma_5 \;=\; \lambda' \quad \Big(\textit{where } \lambda' = \frac{\gamma_1\,\delta\,\epsilon}{1 + \delta\,\epsilon}\,\lambda(A)\Big).$$

(3) ***Stability Criterion.*** *All eigenvalues have negative real parts (so the TFE is locally asymptotically stable) if and only if:*

$$\gamma_2 + \gamma_4 > 0 \quad \textit{and} \quad (1 + \gamma_4)\,(\gamma_3 + \gamma_5) + \gamma_2\,\gamma_5 - \gamma_2\,\lambda' \;>\; 0,$$

which simplifies to

$$\lambda' = \frac{(\alpha_1 q^{\alpha_2} + \beta)\,\delta\,\epsilon}{1 + \delta\,\epsilon}\,\lambda(A) \;<\; \frac{(1 + \gamma_4)\,(\gamma_3 + \gamma_5) + \gamma_2\,\gamma_5}{\gamma_2}. \tag{6}$$

In particular, $\gamma_2 + \gamma_4 > 0$ *is always true if* $d_1 > 0$ *and* $p > 0$*. Hence the main condition is* (6)*.*

(4) ***Expanded Form.*** *Substituting* γ_i *from Section 4 into* (6) *yields*

$$\frac{(\alpha_1\, q^{\alpha_2} + \beta)\,\delta\,\epsilon}{1 + \delta\,\epsilon}\,\lambda(A) \;<\; e^{d_2 p}\Big[\big(e^{-d_3 p} + e^{-\frac{d_3}{p}}\big)\,\big(1 + d_1\, e^{-\frac{d_2}{p}}\big) + d_1\, e^{-d_2 p - \frac{d_3}{p}}\Big].$$

If this holds for every relevant eigenvalue $\lambda(A)$ *(in particular, for* $\lambda_{\max}(A)$*), then the TFE is stable.*

Proof of Theorem 1. A detailed derivation of the Jacobian J^* at x^* shows it to be block lower-triangular except for a $2N \times 2N$ portion responsible for the dynamics of e and o. One finds:

$$J^* \;=\; \begin{bmatrix} -(\gamma_2 + \gamma_4)\, I & \frac{\gamma_1\,\delta\,\epsilon}{1+\delta\epsilon}\, A + \gamma_3\, I & 0 \\ \gamma_2\, I & -(\gamma_3 + \gamma_5)\, I & 0 \\ (-\frac{1}{\epsilon} + \gamma_4) I & -\frac{\gamma_1}{\epsilon^2}\, A & -\frac{1+\delta\epsilon}{\epsilon}\, I \end{bmatrix},$$

where the last block row leads to N eigenvalues at $-\frac{1+\delta\epsilon}{\epsilon} < 0$. The remaining $2N$ eigenvalues come from the characteristic equation of the top-left $2N \times 2N$ block:

$$\Lambda = \begin{bmatrix} -(\gamma_2+\gamma_4)\, I & \frac{\gamma_1\,\delta\,\epsilon}{1+\delta\epsilon}\, A + \gamma_3\, I \\ \gamma_2\, I & -(\gamma_3+\gamma_5)\, I \end{bmatrix}.$$

By examining $\det(\Lambda - \lambda I) = 0$ and taking into account that $\frac{\gamma_1\,\delta\,\epsilon}{1+\delta\epsilon}\,\lambda(A) = \lambda'$, one obtains the quadratic condition

$$\frac{\lambda^2}{\gamma_2} + \left(1 + \frac{\gamma_4}{\gamma_2}\right)\lambda + \frac{(1+\gamma_4)(\gamma_3+\gamma_5)}{\gamma_2} + \gamma_5 \;=\; \lambda'.$$

Standard Routh–Hurwitz criteria for a real-coefficient quadratic show that all such λ have negative real parts precisely if $\gamma_2 + \gamma_4 > 0$ and $(1+\gamma_4)(\gamma_3+\gamma_5) + \gamma_2\,\gamma_5 - \gamma_2\,\lambda' > 0$. The former is automatically true if $d_1 > 0$ and $p > 0$, since $\gamma_2, \gamma_4 \geq 0$. The latter rearranges to

$$\lambda' \;<\; \frac{(1+\gamma_4)(\gamma_3+\gamma_5) + \gamma_2\,\gamma_5}{\gamma_2},$$

as stated in (6). Substituting γ_i completes the expansion. Therefore, if the condition holds for every eigenvalue $\lambda(A)$ (or at least the largest one, $\lambda_{\max}(A)$), then the real part of every λ is negative, establishing local asymptotic stability of the TFE. □

Intuitively, when the advertising intensity $(\alpha_1 q^{\alpha_2} + \beta)$, network structure $\lambda_{\max}(A)$, and other parameters make the left-hand side of (6) large, the TFE is destabilized. Conversely, higher prices p reduce the exponential terms on the right, thus shifting the balance toward a stable TFE in which no one becomes an owner.

Remark 1 (Basic Reproduction Number $\mathcal{R}_0$)**.** *In analogy with epidemiological models, one can interpret the threshold condition*

$$\lambda' \;=\; \frac{(\alpha_1\, q^{\alpha_2} + \beta)\,\delta\,\epsilon}{1+\delta\,\epsilon}\,\lambda(A) \;<\; \frac{(1+\gamma_4)\,(\gamma_3+\gamma_5) \;+\; \gamma_2\,\gamma_5}{\gamma_2}$$

as requiring a suitable "basic reproduction number" $\mathcal{R}_0$ to be below 1. Concretely, define

$$\mathcal{R}_0(\lambda(A)) \;=\; \frac{\gamma_2\,\left(\alpha_1\, q^{\alpha_2} + \beta\right)\delta\,\epsilon}{\left(1+\delta\,\epsilon\right)\left[(1+\gamma_4)\,(\gamma_3+\gamma_5) \;+\; \gamma_2\,\gamma_5\right]} \;\times\; \lambda(A). \tag{7}$$

From the stability condition in Theorem 1, the TFE is locally asymptotically stable precisely when

$$\mathcal{R}_0(\lambda(A)) \;<\; 1 \quad \textit{for all relevant eigenvalues } \lambda(A).$$

In particular, if $\lambda_{\max}(A)$ is the largest eigenvalue of the adjacency matrix A, then

$$\mathcal{R}_0(\lambda_{\max}(A)) \;<\; 1 \quad\Longleftrightarrow\quad \textit{TFE is stable.}$$

Equivalently, if $\mathcal{R}_0(\lambda_{\max}(A))$ exceeds 1, then the TFE becomes unstable and yields non-trivial (endemic) behavior with nonzero ownership. In this sense, $\mathcal{R}_0$ captures the combined effect of (i) the network structure via $\lambda_{\max}(A)$, (ii) the impact of advertising $\left(\alpha_1\, q^{\alpha_2} + \beta\right)$, (iii) the price p embedded in the exponentials of $\gamma_2, \ldots, \gamma_5$, and (iv) the transition rates $\delta, \epsilon, d_1, d_2, d_3$. A higher price p drives down the exponential terms, favoring TFE stability ($\mathcal{R}_0 < 1$), while a greater advertising level q or a denser network (larger $\lambda_{\max}(A)$) inflates $\mathcal{R}_0$ and can destabilize the TFE. Figure 5 illustrates the sensitivity of $\mathcal{R}_0$ to simultaneous variations in the p and q. Each surface corresponds to a distinct configuration of the remaining model parameters.

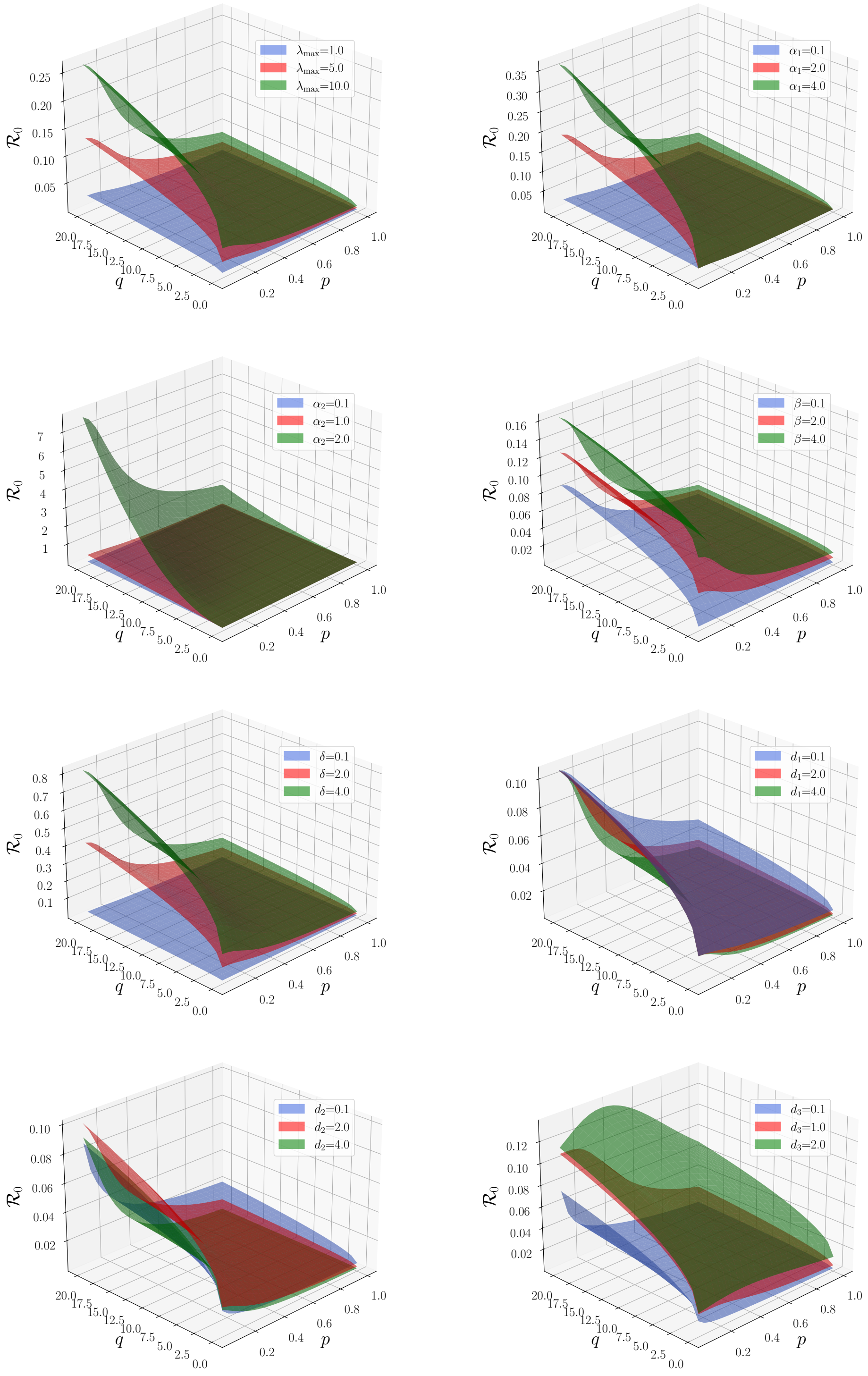

Figure 5: Sensitivity of $\mathcal{R}_0$ to simultaneous variations in the p and q. Each surface corresponds to a distinct configuration of the remaining model parameters. Each plot corresponds to a scenario in which one parameter is perturbed while the remaining parameters are held fixed at their baseline values: $\alpha_1 = 1.0$, $\alpha_2 = 0.5$, $\beta = 1.0$, $\delta = 0.5$, $d_1 = 2.0$, $d_2 = 1.0$, $d_3 = 0.5$, and $\lambda_{\max} = 4.0$.

5. Twin Delayed Deep Deterministic Policy Gradient with Encoded State (TD3ES)

This section introduces a novel twin delayed deep deterministic policy gradient with encoded state (TD3ES) approach, which integrates a graph autoencoder to encode the underlying state space of the problem. Similar to the TD3 algorithm [48], TD3ES employs an actor-critic architecture composed of artificial neural networks (ANNs). In this framework, the actor determines both the price and advertisement level, while the critics estimate the corresponding expected returns. After an action is issued, the consumer-interaction network evolves until the next decision epoch. This evolution is simulated with a parallel, GPU-accelerated Monte-Carlo procedure that accurately captures stochastic adoption dynamics on large graphs. The components of the proposed framework, as well as its training procedure, are elaborated in the following subsections.

To clarify the algorithmic choice, the pricing–advertising controls in this study are continuous (real-valued price and ad intensity), which makes deterministic actor–critic methods a natural fit. TD3 is adopted because it (i) natively handles continuous actions without discretization; (ii) is off-policy and thus more sample-efficient, a practical advantage given that each rollout in the GPU-accelerated simulator remains computationally nontrivial; and (iii) incorporates stabilizing design elements (twin critics to curb overestimation, delayed policy updates, and target-policy smoothing) that are well suited to continuous control. By contrast, deep Q-network (DQN) targets discrete action spaces and would require artificial discretization of price and ad levels, reducing control resolution and introducing approximation error. proximal policy optimization (PPO) can operate in continuous spaces, but being on-policy, it requires fresh trajectories after each update, which is less data-efficient in this simulator-bound setting and can lead to slower progress for finely tuned controls. Importantly, the proposed graph autoencoder is backbone-agnostic: the same latent state can be paired with PPO, soft actor-critic (SAC), or other continuous-control methods. TD3 is therefore used as a standard, stable, and sample-efficient baseline to isolate the contribution of the graph-embedded state representation rather than differences among learning rules.

5.1. State Space

Defining the state space is a critical aspect of any RL algorithm. The state representation encompasses all relevant information that the RL agent receives from the environment and subsequently uses to select actions. In the problem considered here, the environment consists of a networked set of consumers, where their interactions drive both the diffusion of sales and each consumer's position within the CDM Markov process (illustrated in Figure 2). The state definition directly impacts the quality of the agent's actions, underscoring the importance of carefully designing and engineering the state space in order to achieve optimal learning performance.

A straightforward initial approach to defining the state space for this study is to use the percentage of consumers in each compartment of the CDM Markov process. Specifically, this approach tracks the fractions of consumers who have left or are leaving the market (L), those who are still potential buyers (B), those who have been exposed to the product (E), and those who already own the product (O). While this simple definition can perform reasonably well for networks characterized by a uniform node degree distribution, it lacks the necessary granularity to capture the more complex diffusion dynamics that arise in heterogeneous networks. To illustrate this shortcoming, consider two consumer networks that display the same overall proportions of consumers in each CDM compartment. In one network, the node degrees are uniformly distributed, whereas in the other network, the node degrees follow an exponential distribution. Despite having identical compartment percentages, these two networks exhibit fundamentally different topological structures and diffusion behaviors. A state definition based solely on compartment proportions would label both networks with the same state, thereby neglecting critical structural differences that significantly influence the spread of the product. This example highlights the importance of defining a more expressive state space that accounts for both the consumers' compartment memberships and the underlying network structure, enabling more accurate modeling and prediction of diffusion processes.

A second major challenge in specifying the state space arises from the sheer size of real-world consumer networks, which can comprise millions of nodes and connections. Storing or even fully detecting the entire network structure may be computationally prohibitive. Therefore, an effective state definition must balance completeness and tractability: it should encode essential structural and dynamical information, possibly by relying on representative subsets of the network, while remaining compact enough to be computationally feasible. Striking an appropriate compromise between brevity and expressiveness is crucial for developing a state space definition that can realistically support large-scale analysis and deliver reliable insights into the product diffusion process.

In this study, a graph convolutional network (GCN) autoencoder [13] is utilized obtain a compact yet expressive representation of the consumers' network state. Consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of $N = |\mathcal{V}|$ consumers with the adjacency matrix A. The graph is first augmented with self-loops, $\tilde{A} = A + I_N$ and the associated degree matrix $\tilde{D} = \text{diag}(\sum_j \tilde{A}_{ij})$ is formed. Given node features $\boldsymbol{X} \in \mathbb{R}^{N \times F}$, whose rows encode local information of CDM Markov model compartment membership, and normalized degree, the encoder stacks L_{enc} graph convolution layers as follows:

$$H^{(l+1)} = \text{ReLU}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W_{enc}^{(l)}) \tag{8}$$

where l is the layer index, $W_{enc}^{(l)}$ is the trainable weight matrix, and $H^{(l)}$ is the activation matrix where $H^{(0)} = \boldsymbol{X}$. To summarize the entire network in a manner agnostic to graph size, a permutation-invariant function, i.e., a global mean pooling, is applied,

$$\boldsymbol{Z} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} h_v^{(L_{enc})} \tag{9}$$

where $\boldsymbol{Z} \in \mathbb{R}^{F^*}$ is the graph embedding and $h_v^{L_{enc}}$ is the row in $H^{(L_{enc})}$ corresponding with node v.

To reconstruct node-level attributes from the graph embedding, a graph-level decoder with L_{dec} graph-convolution layers is appended to the autoencoder. The decoder operates as follows. The fixed-length embedding $\boldsymbol{Z} \in \mathbb{R}^{F^*}$ obtained in (9) is broadcast to every node $v \in \mathcal{V}$ and linearly projected into a hidden feature space of dimension F_{hid}:

$$h_v^{(0)} = \text{ReLU}(\boldsymbol{Z} W_{init}) \tag{10}$$

where $W_{init} \in \mathbb{R}^{F^* \times F_{hid}}$. The propagated features are then refined through L_{dec} layers of graph convolutions that share the same symmetric normalization used in the encoder:

$$H^{(l+1)} = \text{ReLU}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W_{dec}^{(l)}) \tag{11}$$

where $H^l = [h_1^l, \ldots, h_N^l]^T \in \mathbb{R}^{N \times F_{hid}}$ and $W_{dec}^{(l)} \in \mathbb{R}^{F_{hid} \times F_{hid}}$. Although every node starts from the same vector $h_v^{(0)}$, the repeated message-passing updates allow the adjacency structure to modulate each trajectory $h_v^{(l)}$, enabling the decoder to differentiate nodes according to their local topology. After the final convolutional layer, each node representation is mapped back to the original feature dimensionality, i.e., five channels corresponding to the CDM Markov model compartments and normalized degrees via a linear layer:

$$\hat{\boldsymbol{X}} = \text{ReLU}(H^{L_{dec}} W_{out}) \tag{12}$$

where $W_{out} \in \mathbb{R}^{F_{hid} \times F}$. The reconstructed matrix $\hat{X} \in \mathbb{R}^{N \times F}$ serves as the decoder's estimate of the original node-feature matrix. All decoder parameters $\{W_{init}, W_{dec}^{(0)}, \ldots, W_{dec}^{(L_{dec})}, W_{out}\}$ are optimized jointly with the encoder by minimizing a mean squared error (MSE) reconstruction loss between $\hat{\boldsymbol{X}}$ and $\boldsymbol{X}$. After being trained, the encoder is utilized to retrieve the consumers' network state during the TD3ES training.

The GCN maps the contact network to a low-dimensional latent $\boldsymbol{Z}$ using degree-normalized message passing $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ and Lipschitz activations, followed by permutation-invariant global pooling. This architecture is stable by design: perturbations to edges or node features produce bounded changes in intermediate node embeddings, and pooling averages these variations across the network, attenuating localized shocks. Training further enhances robustness. The autoencoder is trained using domain randomization, sampling graph families and behavior-dynamics parameters from the same generators and ranges employed for training TD3ES. Hence, abrupt yet in-distribution changes (e.g., edge rewires, hub fluctuations, or sudden state-mix shifts) remain within the encoder's learned support. In practice, this yields latents that are insensitive to moderate shocks while remaining responsive to genuine regime changes. For rare out-of-distribution shifts (e.g., structural breaks far outside the training generators), one possible adaptation is to temporarily unfreeze the encoder and perform a few low-learning-rate updates on recent data (while keeping the actor–critic fixed), then refreeze. This optional step re-centers $\boldsymbol{Z}_t$ on the new regime without destabilizing control.

It is worth noting that the graph-encoded representation $\boldsymbol{Z}$ employed in TD3ES can be interpreted as a form of state abstraction in the reinforcement learning literature. In classical RL, state abstraction refers to the process of mapping the original state space into a reduced representation that preserves decision-relevant information while discarding superfluous detail [49]. The graph-convolutional autoencoder in the proposed framework performs exactly this role: it compresses the high-dimensional, node-level consumer network states into a compact latent vector that still retains the essential structural and behavioral patterns needed for effective control. Framing the encoder as a learned state abstraction situates our approach within a broader theoretical tradition and highlights how autoencoder-based embeddings can serve as a practical instantiation of abstraction mappings for complex, networked environments.

5.2. Simulation of Diffusion Process Over a Consumers' Network

After TD3ES selects the control parameters (i.e., price p and advertisement level q) the consumer network evolves as a stochastic process defined by model (1). This evolution can be simulated with several methods rooted in the Gillespie algorithm [46, 47], including GEMFsim [32] and FastGEMF [33]. The Gillespie algorithm is a Monte Carlo method adept at generating statistically precise sample trajectories of interlinked chemical reactions. It functions by sampling the time until the occurrence of the next reaction event from an exponential distribution, with the rate parameter equating to the summation of propensities associated with all feasible reactions. Subsequently, the reaction to transpire is selected probabilistically in proportion to its propensity function. Notably, the algorithm has been modified in [32] and [33] to simulate spreading processes on networks, where node state transitions are regarded as reaction events. The propensity functions are computed based on nodal transition rates and edge-based transition rates stipulated within a generalized epidemic modeling framework. Leveraging this adapted algorithm, realizations of the stochastic spreading process across the network can be generated. At each iteration, the duration until the subsequent node state transition is sampled, the node along with its new state are probabilistically chosen based on the transition rates, and the rates are updated before iterating the process. By generating multiple realizations, pertinent statistics concerning the spread of the product's sales can be accurately estimated.

A limitation of simulation-based algorithms is their high computational cost when executed on CPUs. This limitation becomes more pronounced when repeated simulations are required during the interaction of an RL agent with its environment, especially on large-scale networks. To alleviate this computational burden, the present study exploits the GPU capabilities of the message-passing framework implemented in PyTorch Geometric [34], which allows transition updates and rate calculations to be performed in parallel.

Algorithm 1 outlines the GPU-accelerated stochastic simulation employed to propagate behavioral states across a consumer network. At every time step Δt, the routine first computes the state-dependent transition rates from the one-hot node-state matrix $\mathbf{x}$ and the aggregated neighbor message matrix $\mathbf{m}$. For each node, the total exit rate is obtained by summing the competing hazards associated with its current state. The probability that any transition occurs during Δt is then $1 - \exp(-r\Delta t)$. If a transition occurs, a uniform random draw selects the destination compartment according to the relative magnitudes of the competing rates. To emphasize the parallelizability of the procedure, all nodes execute these steps independently and simultaneously on the GPU. The algorithm uses $\mathbf{x}' \leftarrow \mathbf{x}$ to initialize the updated matrix: this ensures that nodes which do not transition in the current step are correctly retained, while only those undergoing a state change are overwritten. This separation between $\mathbf{x}$ and $\mathbf{x}'$ also prevents unintended side effects in a parallel implementation and makes the update semantics explicit. The resulting design eliminates serial looping, exploits GPU concurrency, and preserves the stochastic semantics of the underlying continuous-time Markov process within an event-driven discretization. Finally, note that $\mathbf{x}$ in Algorithm 1 differs from the node feature matrix $\boldsymbol{X}$ used for training the autoencoder, as it contains no node degree information.

Algorithm 1 GPU-Accelerated Diffusion Simulation over Consumers' Network

```
Require: Node-state matrix $\mathbf{x} \in \{0,1\}^{N\times 4}$          ▷ rows are one–hot over $(L, B, E, O)$
Require: Aggregated neighbor messages $\mathbf{m} \in \mathbb{R}^{N\times 4}$          ▷ $\mathbf{m}_{(:,3)}$ = # neighbors in $O$
Require: Model params $\left(\delta, \alpha_1, \alpha_2, \beta, \epsilon, d_1, d_2, d_3, p, q, \Delta t\right)$
Ensure: Updated node-state matrix $\mathbf{x}'$
Precompute rates (vectorized over $i$):
$r_{LB} \leftarrow \delta$          ▷ $L \rightarrow B$
$r_{BE}(i) \leftarrow (\alpha_1 q^{\alpha_2} + \beta) \cdot \mathbf{m}_{(i,3)}$          ▷ $B \rightarrow E$
$r_{BL}(i) \leftarrow \left(r_{BE}(i) + \epsilon\right)^{-1}$          ▷ $B \rightarrow L$
$r_{EO} \leftarrow d_1 e^{-d_2 p}$,   $r_{EL} \leftarrow d_1 e^{-d_2/p}$          ▷ $E \rightarrow O, L$
$r_{OE} \leftarrow d_1 e^{-d_3 p}$,   $r_{OL} \leftarrow d_1 e^{-d_3/p}$          ▷ $O \rightarrow E, L$
Encode state index: $s(i) \leftarrow \arg\max_{k\in\{0,1,2,3\}} \mathbf{x}_{(i,k)}$          ▷ 0=$L$, 1=$B$, 2=$E$, 3=$O$
$\mathbf{x}' \leftarrow \mathbf{x}$
for all $i \in \mathcal{V}$ in parallel do
    Get outgoing rate $r$ for current state $s(i)$:
      if $s_i = 0$ $(L)$: $r \leftarrow r_{LB}$
      if $s_i = 1$ $(B)$: $r \leftarrow r_{BE}(i) + r_{BL}(i)$
      if $s_i = 2$ $(E)$: $r \leftarrow r_{EO} + r_{EL}$
      if $s_i = 3$ $(O)$: $r \leftarrow r_{OE} + r_{OL}$
    if rand() $< 1 - \exp(-r\Delta t)$ then
        $u \sim \mathrm{Unif}(0, 1)$
        switch $s_i$:
          case 0 (L):   $s' \leftarrow 1$          ▷ only $L \rightarrow B$ with rate $r_{LB}$
          case 1 (B):   if $u < \frac{r_{BE}(i)}{r}$ then $s' \leftarrow 2$ else $s' \leftarrow 0$
          case 2 (E):   if $u < \frac{r_{EO}}{r}$ then $s' \leftarrow 3$ else $s' \leftarrow 0$
          case 3 (O):   if $u < \frac{r_{OE}}{r}$ then $s' \leftarrow 2$ else $s' \leftarrow 0$
        $\mathbf{x}'_{(i,:)} \leftarrow \mathrm{onehot}(s')$
    end if
end for
return $\mathbf{x}'$
```

5.3. Actor-Critic Architecture

As outlined in Algorithm 2, TD3ES adapts the TD3 framework [48] to a network-diffusion marketing setting in which the agent's state is not a raw list of node features but a compact graph embedding obtained from a graph-convolutional autoencoder. At the start of training, the actor (parameterised by θ) and two critics (ϕ_1, ϕ_2) are initialized, together with their exponentially moving-average target copies. An empty replay buffer $\mathcal{R}$ stores past transitions for off-policy learning. Each episode begins by drawing a fresh consumer network $\mathcal{G}$ with feature matrix $\boldsymbol{X}$, then encoding that graph via the encoder's final layer activations and a global mean pooling to produce a single latent vector $\boldsymbol{Z}$ that summarizes the entire market.

During the episode, the agent repeatedly selects a continuous action $a = (p, q)$ (price and advertising intensity) by passing the latent state through the deterministic actor and adding Gaussian exploration noise. The chosen controls drive the stochastic adoption dynamics: Algorithm 1 is executed to propagate product awareness and ownership across the network for the simulation interval t, returning an updated node-state matrix $\mathbf{x}'$, from which a new embedded state $\boldsymbol{Z}'$ is computed. The reward Re is the per-consumer profit accrued in that interval: the number of newly adopted nodes constitutes revenue, while advertising expenditure is proportional to the chosen intensity q and the network size. At each decision step t, the agent receives an immediate reward defined as

$$Re_t = \frac{p\Delta O_t - cqN\Delta t}{N}, \tag{13}$$

where ΔO_t denotes the number of new consumers adopting the product during the interval $(t, t+1]$, p is the price, q is the advertising intensity, N is the population size, c is the unit advertising cost per consumer per time unit, and Δt is the length of the interval $(t, t+1]$. The objective of the RL agent is to maximize the expected discounted return:

$$DR_t = \mathbb{E}\left[\sum_{k=0}^{\infty} \Gamma^k Re_{t+k}, \middle| \boldsymbol{Z}_t, a_t\right], \tag{14}$$

where $\Gamma \in (0, 1]$ is the discount factor, $\boldsymbol{Z}_t$ represents the state at time t, and $a_t = (p_t, q_t)$ denotes the action composed of price and advertising intensity. A binary flag κ indicates whether the time horizon T has been reached. The tuple $(\boldsymbol{Z}, a, Re, \boldsymbol{Z}', \kappa)$ is then stored in the replay buffer $\mathcal{R}$ for later use. For clarity, note that in Algorithm 2, the subscript t is omitted to simplify notation.

The action space is defined as

$$\mathcal{A} = \left\{(p, q) \in \mathbb{R}^2 \;\middle|\; p_{\min} \le p \le p_{\max},\; q_{\min} \le q \le q_{\max}\right\}, \tag{15}$$

where p denotes the product's price and q denotes the advertising intensity chosen by the firm. The bounds $[p_{\min}, p_{\max}]$ and $[q_{\min}, q_{\max}]$ reflect feasible operational ranges. At each decision epoch, the actor network μ_θ maps the state embedding $\boldsymbol{Z}$ to a continuous action $a = (p, q) \in \mathcal{A}$, where $\mathcal{A}$ is the bounded action space defined in Eq. (15). The clip$(\cdot, a_{Low}, a_{High})$ operator in Algorithm 2 implements these bounds.

Whenever the pre-set "update period" elapses, the algorithm performs one or more gradient steps. For each update, a minibatch is sampled from the buffer. Target actions are produced with the target actor plus clipped noise and target Q-values are computed with the smaller of the two target critics, mitigating positive bias in value estimates. Both critics are then regressed toward these targets by minimizing a MSE loss. In accordance with TD3's delayed-policy update rule, the actor is updated only every *policy_ delay* iterations by maximizing the first critic's Q-value. Immediately afterwards, all target networks receive a Polyak-averaged update with factor ρ to maintain slowly moving targets and stabilize training.

By unifying a graph-level state encoder, a market-specific reward function, and a GPU-accelerated diffusion simulator within TD3's twin-critic architecture, the TD3ES algorithm learns a pricing and advertising strategy that maximizes long-run profit over dynamically evolving consumer networks while remaining computationally tractable for large-scale problems.

Figure 6 translates Algorithm 2 into a systems-level block diagram. The workflow starts at the actor panel, where the primary policy network μ_θ receives the graph embedding $\boldsymbol{Z}$ and outputs a noisy control vector $a = (p, q)$. This action is carried to the environment, which combines the GPU-accelerated diffusion simulator and the underlying consumer network $\mathcal{G}$. After one simulation step, the updated node states are passed to the Encoder, whose mean-pooled output yields the next global state $\boldsymbol{Z}'$. The quintuple $(\boldsymbol{Z}, a, Re, \boldsymbol{Z}', \kappa)$ then flows into the Replay Buffer, from which both the actor and the Critic panel sample training minibatches. Inside the Critic panel, two independent value networks, Q_{ϕ_1} and Q_{ϕ_2}, and their corresponding target copies assess the quality of actions. Dashed red arrows, annotated with Polyak-averaging formulas, highlight the delayed soft updates that keep target networks close to their online counterparts. Additional dashed red paths show gradient back-propagation for the actor and critics.

5.4. Training TD3ES

During training, each episode begins with a freshly synthesized consumers' network so that the agent never sees the same problem twice. A random draw first decides whether the contact graph is exponential

Algorithm 2 Twin Delayed Deep Deterministic Policy Gradient With Encoded State (TD3ES)

```
initiate actor parameters $\theta$, critics' parameters $\phi_1$ and $\phi_2$, and an empty replay buffer $\mathcal{R}$
set target parameters $\theta_{targ} \leftarrow \theta$, $\phi_{1,targ} \leftarrow \phi_1$, $\phi_{2,targ} \leftarrow \phi_2$
for each episode do
    generate a random consumers' network $\mathcal{G}$ with node features $\boldsymbol{X}$
    using the GCN autoencoder, observe the consumers' network embedded state $\boldsymbol{Z} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} h_v^{(L_{enc})}$
    while end time $T$ is not reached do
        set action $a = \text{clip}(\mu_\theta(\boldsymbol{Z}) + \zeta, a_{Low}, a_{High})$, where $\zeta$ is drawn from a normal distribution with average zero and $a = (p, q)$
        simulate the stochastic model (1) using Algorithm 1 and obtain next node-state matrix $\mathbf{x}'$
        augment $\mathbf{x}'$ with normalized node degrees and obtain $\boldsymbol{X}$
        observe the next state $\boldsymbol{Z}'$ using equations (8) and (9), reward $Re$, and the termination binary indicator $\kappa$
        store $(\boldsymbol{Z}, a, Re, \boldsymbol{Z}', \kappa)$ in the replay buffer $\mathcal{R}$
        if time to update then
            for $k$ in number of updates do
                pick a random batch $\mathcal{B} = \{(\mathcal{Z}, a, Re, \boldsymbol{Z}', \kappa)\}$ from $\mathcal{R}$
                $a'(\boldsymbol{Z}') = \text{clip}(\mu_{\theta_{targ}}(\boldsymbol{Z}') + \text{clip}(\zeta, -c, c), a_{Low}, a_{High})$        ▷ Compute target actions
                $y(Re, \boldsymbol{Z}', \kappa) = Re + \Gamma(1-\kappa)\min_{j=1,2} Q_{\phi_{j,targ}}(\boldsymbol{Z}', a'(\boldsymbol{Z}'))$        ▷ Compute targets
                take one step of gradient descent on the critic loss $\nabla_{\phi_j}\mathcal{L}_Q(\phi_j, \mathcal{R}) = \nabla_{\phi_j}\frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{Z},a,Re,\boldsymbol{Z}',\kappa)\in\mathcal{B}}\left[\left(Q_{\phi_j}(\boldsymbol{Z}, a) - y(Re, \boldsymbol{Z}', \kappa)\right)^2\right]$ for $j = 1, 2$
                if $k$ mod $policy\_delay = 0$ then
                    take one step of gradient descent on the actor loss $\nabla_\theta\mathcal{L}_\mu(\theta, \mathcal{R}) = \nabla_\theta\frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{Z})\in\mathcal{Z}} Q_{\phi_1}(\boldsymbol{Z}, \mu_\theta(\boldsymbol{Z}))$
                    $\phi_{j,targ} \leftarrow \rho\phi_{j,targ} + (1-\rho)\phi_j$        $j = 1, 2$        ▷ Update the critic target networks
                    $\theta_{targ} \leftarrow \rho\theta_{targ} + (1-\rho)\theta$        ▷ Update the actor target network
                end if
            end for
        end if
    end while
end for
```

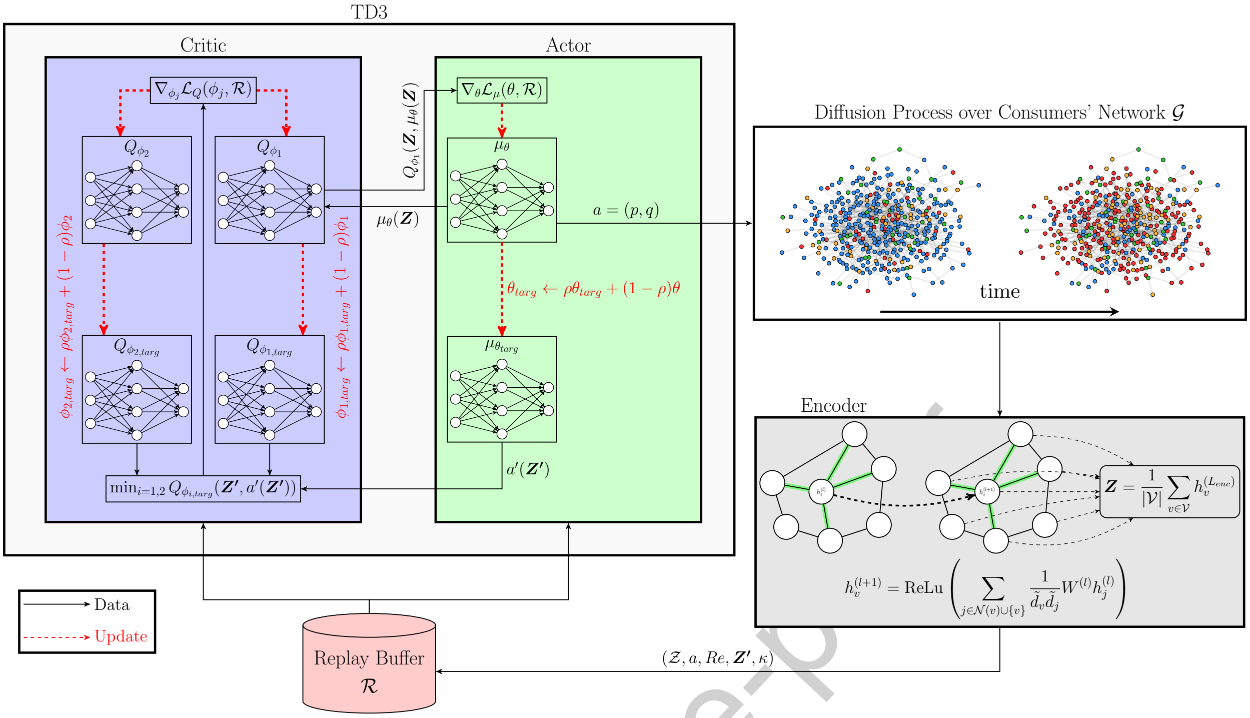
TD3
Critic
Actor
$\nabla_{\phi_j}\mathcal{L}_Q(\phi_j,\mathcal{R})$
Q_{ϕ_2}
Q_{ϕ_1}
$Q_{\phi_{2,targ}}$
$Q_{\phi_{1,targ}}$
$\phi_{2,targ} \leftarrow \rho\phi_{2,targ} + (1-\rho)\phi_2$
$\phi_{1,targ} \leftarrow \rho\phi_{1,targ} + (1-\rho)\phi_1$
$\min_{i=1,2} Q_{\phi_{i,targ}}(\boldsymbol{Z'}, a'(\boldsymbol{Z'}))$
$Q_{\phi_1}(\boldsymbol{Z}, \mu_\theta(\boldsymbol{Z}))$
$\mu_\theta(\boldsymbol{Z})$
$\nabla_\theta\mathcal{L}_\mu(\theta,\mathcal{R})$
μ_θ
$a = (p, q)$
$\theta_{targ} \leftarrow \rho\theta_{targ} + (1-\rho)\theta$
$\mu_{\theta_{targ}}$
$a'(\boldsymbol{Z'})$
Data
Update
Replay Buffer
$\mathcal{R}$
$(\mathcal{Z}, a, Re, \boldsymbol{Z'}, \kappa)$
Diffusion Process over Consumers' Network $\mathcal{G}$
time
Encoder
$h_i^{(l)}$
$h_i^{(l+1)}$
$\boldsymbol{Z} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} h_v^{(L_{enc})}$
$h_v^{(l+1)} = \text{ReLu}\left(\sum_{j\in\mathcal{N}(v)\cup\{v\}} \frac{1}{\tilde{d}_v\tilde{d}_j} W^{(l)} h_j^{(l)}\right)$

Figure 6: Twin Delayed Deep Deterministic Policy Gradient with Encoded State (TD3ES).

or Erdős–Rényi; the former is created by sampling node degrees from an exponential distribution with rate $\lambda_{\text{exp}} \sim \mathcal{U}(0.03, 0.8)$, while the latter is generated with an average degree drawn from $\mathcal{U}(2, 25)$. All graphs contain $N = 10,000$ individuals, each initialized in one of the four CDM compartments L, B, E, and O. Key parameters that steer the stochastic adoption process are resampled independently every reset: advertising sensitivity $\alpha_1, \alpha_2 \in [0.1, 3]$, network externality $\beta \in [0.1, 3]$, transition rate $\delta \in [0.1, 3]$, and the price-demand triplet $d_1, d_2, d_3 \in [0.1, 3]$.

The observation presented to the agent concatenates (i) a 32-dimensional graph-level latent vector produced by the pre-trained autoencoder and (ii) 14 scalar summaries of the current population state and physical parameters, yielding a fixed 46-component state signal. Learning proceeds similar to TD3: two critics are updated at every step, whereas the actor is updated only after every second critic update; target networks are softly blended with rate $\rho = 0.005$. Exploration noise is zero-mean Gaussian with standard deviation 0.1, and the learning rate follows a linear decay from 10^{-3} to 10^{-5} over the full budget of 10^5 environment interactions.

To ensure robustness of the reinforcement learning results, the hyperparameters of the TD3ES agent were selected following the recommended default values provided by the Stable-Baselines3 (SB3) library, which are widely adopted in the literature. These defaults were further validated through trial-and-error adjustments to confirm stable convergence in the present setting. All final hyperparameter values used in the training and episode generation are reported in Table 1.

An ablation study was conducted on the autoencoder to assess the impact of hidden layer width and latent size on reconstruction performance. The results show that very small latent sizes tend to underfit, while very large latent sizes increase reconstruction loss, with the best performance obtained at moderate latent sizes (8–16) paired with wider hidden layers (32–64). The baseline configuration used in this work (hidden dimension of 64 and latent dimension of 32) lies close to this optimum and achieves competitive accuracy, confirming that the chosen design is well-calibrated. Figure 7 illustrates the reconstruction loss across the different configurations, highlighting both the optimal region and the robustness of the baseline setting.

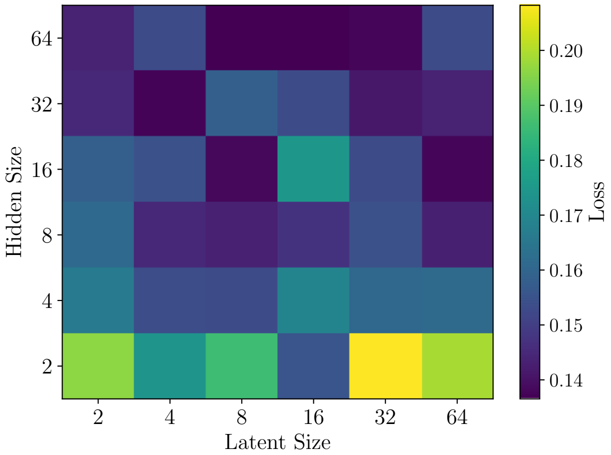

Figure 7: Autoencoder loss for different values of hidden and latent dimensions.

Table 1: Key hyperparameters for environment creation and TD3ES training.

Hyperparameter	**Value or Range**	**Purpose**
	Episode generation	
Number of nodes N	$10,000$	population size
Graph type	exponential or Erdős–Rényi	diverse topologies
Exponential rate λ_{exp}	0.03–0.8	tail of degree distribution
Erdős–Rényi average degree	2–25	connectivity density
α_1, α_2 (ad sensitivity)	0.1–3	marketing efficacy
β (network externality)	0.1–3	peer influence
δ (spontaneous L→B)	0.1–3	latent churn
d_1, d_2, d_3 (price response)	0.1–3	demand elasticity
Autoencoder layers size	2 layers (64, 32)	hidden and latent dim
Graph embedding size	32	autoencoder latent dim
	TD3ES	
Total environment interactions	$100,000$	training horizon
Learning-rate schedule	$10^{-3} \rightarrow 10^{-5}$ linear	stable convergence
Replay-buffer capacity	1×10^6	off-policy memory
Mini-batch size	100	stochastic updates
Discount factor Γ	0.99	long-term return
Soft-update rate ρ	0.005	target smoothing
Policy-delay interval	2 steps	mitigate bias
Action-noise ζ std. dev.	0.1	exploration

All training procedures and numerical experiments were conducted on two NVIDIA RTX 4090 GPUs. Without access to GPU acceleration, the practical upper limit for the network size in terms of number of nodes N was approximately 500 with an average degree of 5. Under these conditions, a single training session required nearly 12 hours when implemented using the GEMFsim algorithm [32]. Attempting to scale beyond this threshold led to prohibitively long runtimes, rendering experiments with larger networks infeasible for the authors. By incorporating GPU acceleration through Algorithm 1, the computational efficiency was dramatically enhanced. The feasible network size increased to 10,000 nodes with an average degree of up to 30, while the total training time was reduced to approximately 6 hours. This advancement represents a substantial improvement in scalability, enabling the proposed method to be applied to far larger and more complex networks than was previously possible.

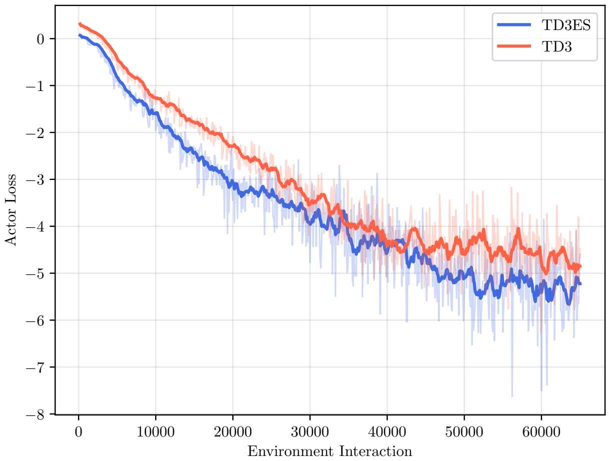

Figure 8: Actor-loss convergence.

To quantify learning efficiency, the study tracks the actor loss (Figure 8), which decreases as the policy improves. Across runs, the TD3ES agent's actor loss stabilizes after approximately 5×10^4 environment interactions, indicating convergence of the policy update. In contrast, TD3 baseline converges to a higher terminal loss, consistent with noisier gradient signals. Consequently, an offline training phase with $\mathcal{O}(10^4 - 10^5)$ simulator interactions suffices for a stable policy that can be executed with negligible computational overhead at deployment.

5.5. Positioning and Comparability of RL Backbones

The RL component of this study is designed to illustrate the value of graph-embedded state representations rather than to conduct a competition across alternative RL algorithms. The encoder–decoder architecture proposed here is algorithm-agnostic: its latent state vector can, in principle, be coupled with a wide variety of continuous-control RL backbones, including TD3, deep deterministic policy gradient (DDPG), PPO, SAC, or even model-based controllers. The decision to instantiate the approach with twin delayed deep deterministic policy gradient (TD3) reflects its status as a widely used benchmark in continuous-action problems and provides a controlled environment for isolating the effect of the proposed graph-convolutional autoencoder.

Accordingly, the empirical comparisons in this paper focus on with/without encoder ablations within the same TD3 framework. This design ensures that improvements in convergence speed, stability, and profit can be attributed to the representation itself, rather than to differences in algorithmic hyperparameters, exploration schedules, or optimization heuristics across families of RL methods. While it is conceivable that another RL backbone (e.g., SAC or PPO) might outperform TD3 on the pricing–advertising task, the essential point is that such algorithms could also benefit from the same graph-based state encoding. Thus, the encoder provides a plug-and-play enhancement to the broader class of RL controllers for networked diffusion problems.

6. Numerical Experiments

This section presents a comprehensive suite of numerical experiments devised to assess the performance of the proposed TD3ES framework across a broad spectrum of diffusion scenarios. Guided by the analytical insights from Section 4, in particular, Theorem (1) and Remark (1), each experiment is configured so that the basic reproduction number, $\mathcal{R}_0$, can take values on both sides of the critical threshold 1 for different combinations of price p and avertising intensity q. Consequently, some test instances represent inherently “easy” markets where $\mathcal{R}_0 > 1$ for most control settings, while others mimic “hard” markets in which $\mathcal{R}_0$ remains below 1 unless interventions are particularly aggressive.

Structuring the experiments around $\mathcal{R}_0$ makes it possible to evaluate TD3ES under distinct diffusion regimes- sub-critical, near-critical, and super-critical- and thereby obtain a nuanced picture of the algorithm’s ability to accelerate adoption when network effects are weak and to restrain over-expansion when contagion is strong. The specific parameter configurations used in these tests are summarized in Table 2, where each parameter is assigned multiple “levels” that capture differing degrees of structural complexity or behavioral sensitivity within the consumer network.

Table 2: Parameter Levels Used in the Test Problems. Each parameter has multiple levels capturing different regimes of sensitivity or structure.

Parameter	Levels	Interpretation
Initial Compartments $(L\%, B\%, E\%, O\%)$	(0.5, 0.3, 0.1, 0.1)	High fraction of individuals initially left (L), moderate potential buyers (B), small fraction exposed (E), and small percentage of owners (O).
	(0.3, 0.2, 0.4, 0.1)	Fewer individuals left, fewer potential buyers, more initially exposed, and similar small ownership.
	(0.1, 0.1, 0.5, 0.3)	Low fraction of left and potential buyers, higher initial exposure, and more owners.
Advertisement Sensitivity (α_1, α_2)	(0.5, 0.5)	Low ad sensitivity; network effect is less amplified by advertising.
	(1.5, 1.5)	Medium ad sensitivity.
	(2.5, 2.5)	High ad sensitivity; advertisement strongly boosts exposure through network connections.
Price Sensitivity (d_1, d_2, d_3)	(0.5, 0.5, 0.5)	All three parameters (d_1, d_2, d_3) are relatively low, so the maximum transition rate is relatively low, and product demand and transition rates are less sensitive to price.
	(0.5, 2.0, 2.0)	Low d_1 but high d_2, d_3; consumers are more sensitive to price changes in post-exposure or post-ownership phases.
	(2.5, 0.5, 0.5)	High d_1 but low d_2, d_3; maximum transition rate is high, but leaving/repurchase transitions are less price-sensitive.
	(2.5, 2.0, 2.0)	High price sensitivity overall, so both demand and repurchase decisions respond strongly to price.
Network Externality β	0.2	Low externality; peers owning the product have a modest influence on new sales.
	1.5	Medium externality.
	2.5	High externality; strong peer effects significantly boost adoption.
Average Degree $\bar{d}$	4	*Low* average degree; consumers have fewer connections on average.
	12	*Medium* connectivity.
	20	*High* average degree; each consumer is well-connected, facilitating rapid exposure via the network.

To benchmark the efficacy of the proposed TD3ES framework, a deliberately simplified TD3 baseline is constructed. In this reference model the 32-dimensional graph-embedding vector learned by TD3ES's autoencoder and ordinarily included in the environment's state is removed. It is replaced by four scalar features that report the instantaneous percentages of nodes occupying the compartments of the CDM Markov model. All remaining elements of the agent architecture, training schedule, and reward structure are held identical between the two approaches, ensuring a controlled, like-for-like comparison.

By contrasting TD3ES with this compartment-only TD3 baseline, the study isolates the informational value of the learned graph embedding. Any observed performance gains can thus be attributed to the autoencoder's ability to encode topological nuances such as clustering, degree heterogeneity, and community structure that simple aggregate counts cannot convey. The experiment therefore provides a direct measure of how much network-level context enhances the agent's capacity to anticipate diffusion dynamics and to select price–advertising actions that maximize long-run profit. To further evaluate the robustness of this approach, the experiments are conducted on two distinct network topologies: Erdős-Rényi and exponential networks.

Finally, to evaluate the benefits of adopting an RL-based sequential decision-making approach, the proposed TD3ES framework is compared against a static baseline policy derived from the equilibrium state of the underlying network. In this baseline, both the price and the advertising intensity are held fixed throughout the entire episode horizon, allowing for directly contrasting static and adaptive strategies.

6.1. Test Problems with Erdős-Rényi Consumers' Network

Table A.3 in Appendix A reports the profits generated by the baseline TD3 agent and the enhanced TD3ES agent over a spectrum of problem-parameter configurations. Complementing these numerical results, Figure 9 visualizes the average profit achieved by each agent as the key parameters vary. A Wilcoxon signed-rank test indicates that the difference in average profit between TD3ES and TD3 is not statistically significant (p-value=0.424). Consistent with this statistical result, inspection of Figure 9 shows that TD3ES does not systematically outperform TD3 for any particular setting of the parameters.

This outcome is particularly informative given the structure of the underlying consumer network. Because the simulations employ an Erdős-Rényi topology-where every node has, in expectation, a similar degree-the network offers little heterogeneity for a graph-embedding model to exploit. In such uniformly connected settings, the autoencoder used by TD3ES cannot gain a substantive informational advantage over the aggregate state representation used by plain TD3. Consequently, the absence of a performance gap should not be interpreted as evidence against the broader effectiveness of TD3ES; rather, it highlights that the added representational power becomes valuable only when the network exhibits richer topological diversity (e.g., pronounced clustering, community structure, or degree heterogeneity) that can inform more nuanced price-advertising decisions.

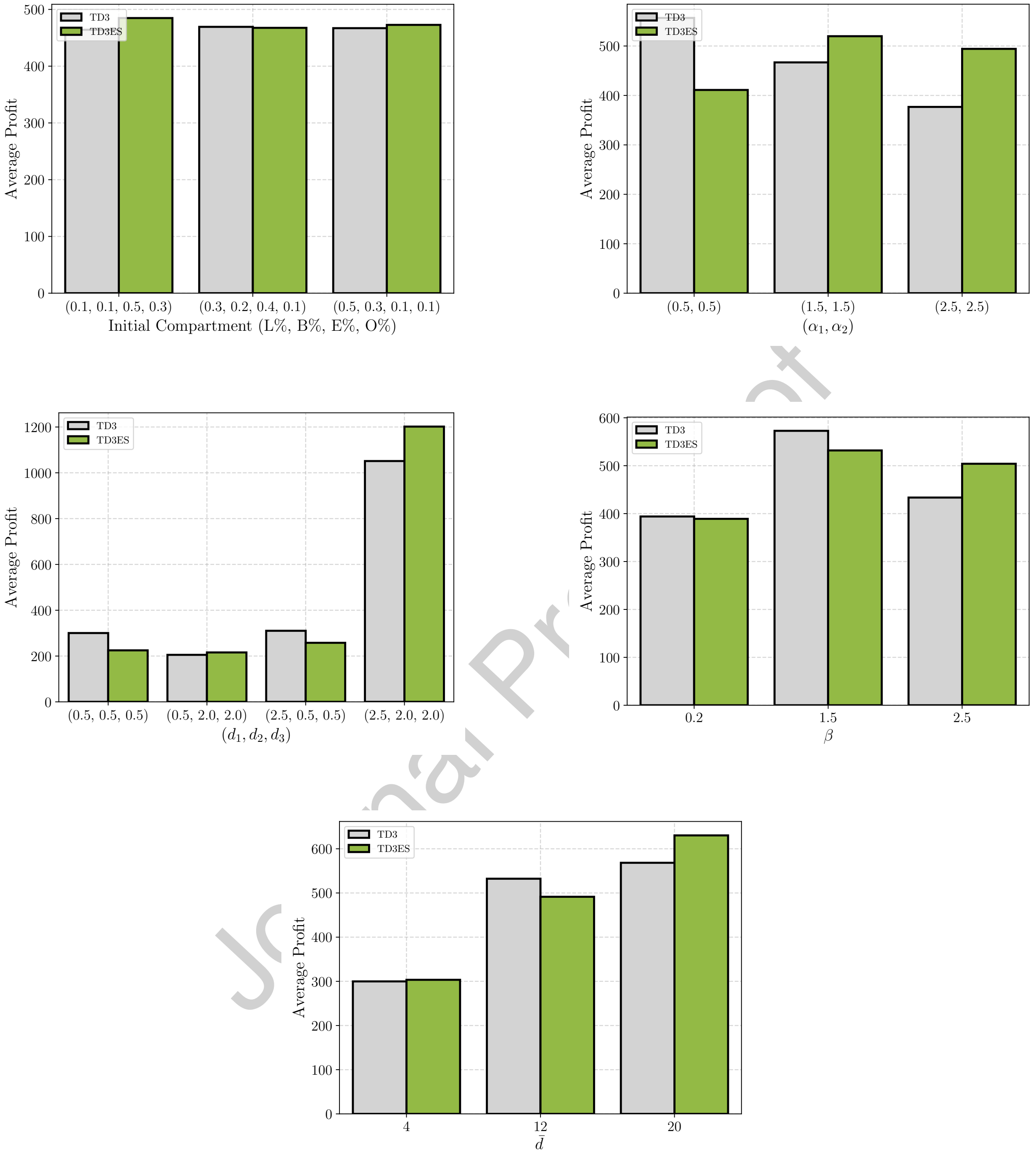

Figure 9: Average profit for Erdos-Renyi test problems across various parameter values.

6.2. *Test Problems with Exponentially Node Degree Distribution*

Table A.4 in Appendix A summarises the profits earned by the baseline TD3 agent and the autoencoder-enhanced TD3ES agent when the underlying consumer network follows an exponential degree distribution. Figure 10 complements these figures by plotting each agent's mean profit as the key problem parameters vary.

A Wilcoxon signed-rank test confirms that TD3ES achieves a statistically significant advantage over TD3 (p-value ≤ 0.001). Visual inspection of Figure 10 corroborates this result: TD3ES consistently surpasses TD3 across nearly the entire parameter space.

The performance gap reflects the pronounced heterogeneity of exponential networks. Such networks contain a small number of highly connected hubs alongside many sparsely connected peripheral nodes, creating diverse local contexts that a graph autoencoder can exploit. By embedding these topological nuances, especially hub dominance, long-tailed degree variability, and implicit community structure, TD3ES constructs a far richer state representation than the aggregate counts available to the plain TD3 agent. This additional information enables more accurate forecasts of diffusion dynamics and, in turn, more profitable price and advertising actions. Hence, the empirical gains reported in Table A.4 and Figure 10 should be interpreted as evidence that the representational power of TD3 becomes most valuable when the environment's network exhibits substantial structural diversity.

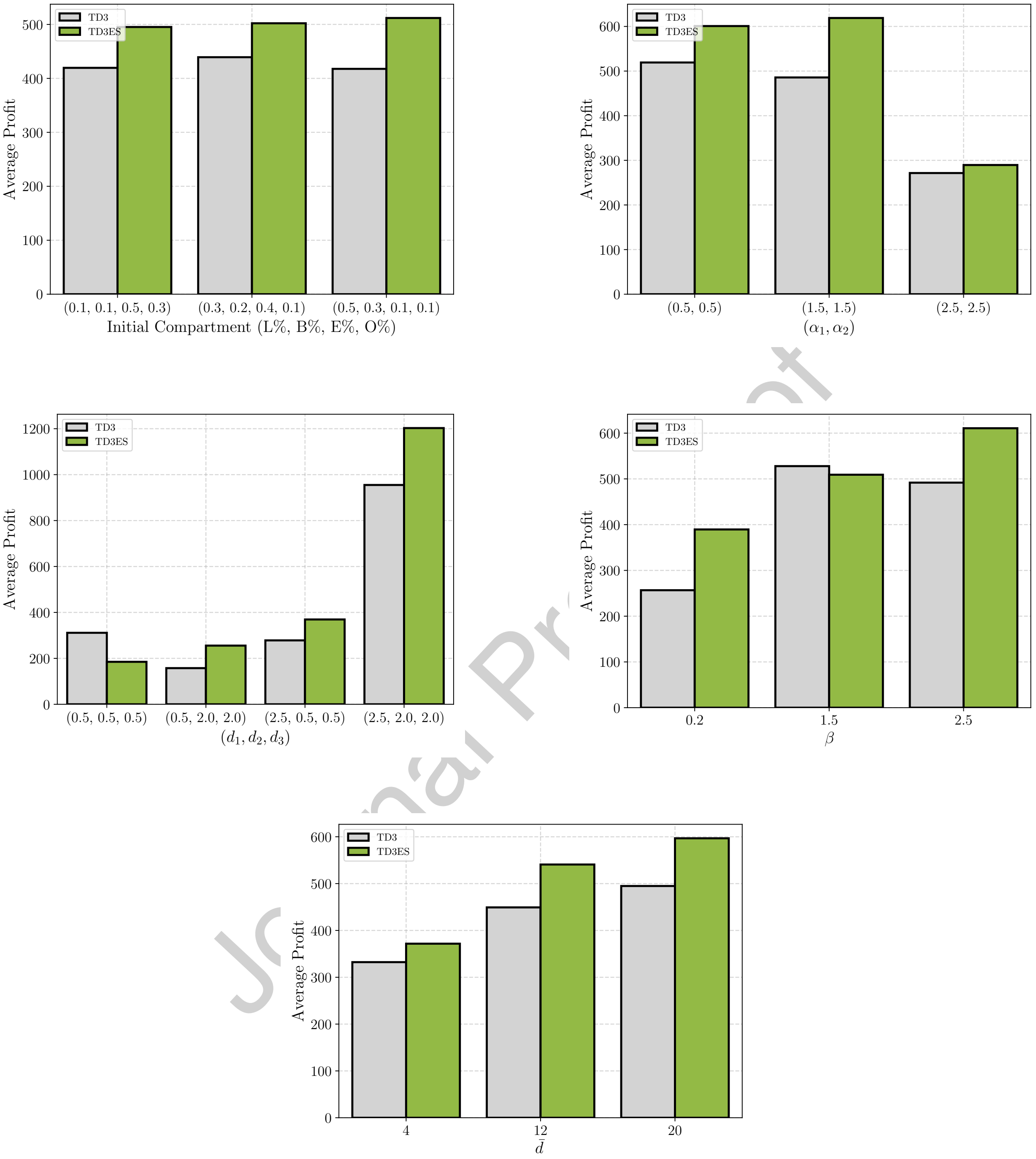

Figure 10: Average profit for test problems with exponentially distributed node degrees across various parameter values.

6.3. TD3ES vs. Static Policy

To measure the value of a sequential decision-making framework such as TD3ES, this section constructs a static policy as a benchmark. This policy assumes that the firm commits to a fixed price p and a fixed advertising intensity q for the entire decision horizon, rather than adapting actions over time. The performance of such a static strategy is then compared against the proposed TD3ES approach.

Let $x(t) = (l(t), b(t), e(t), o(t))$ denote the fractions of consumers in the leaving (L), buyer (B), exposed (E), and owner (O) states at time t. Given a fixed control pair (p, q), the system dynamics follow the mean–field approximation of Section 4.

$$\dot{x}(t) = F(x(t); p, q) \tag{16}$$

where the transition from exposed (E) to owner (O) occurs at rate

$$f(p) = d_1 e^{-d_2 p}$$

and the competing transition $E \to L$ occurs at rate

$$f(\frac{1}{p}) = d_1 e^{-d_2/p}$$

At equilibrium, the compartment fractions (l^*, b^*, e^*, o^*) satisfy

$$F\big((l^*, b^*, e^*, o^*); p, q\big) = 0.$$

During each decision interval of length Δt, the expected fraction of new owners generated from the exposed compartment is

$$\Delta O(p, q) \;=\; e^* \Big(1 - e^{-(f(p)+f(\frac{1}{p}))\Delta t}\Big) \cdot \frac{f(p)}{f(p) + f(\frac{1}{p})} \tag{17}$$

The corresponding expected per-capita profit in equilibrium is then

$$\pi(p, q) \;=\; p \cdot \Delta O(p, q) \;-\; c\, q\, \Delta t \tag{18}$$

The static optimal control (p^*, q^*) is defined as the solution to

$$(p^*, q^*) = \underset{p \in [p_{\min}, p_{\max}],\; q \in [q_{\min}, q_{\max}]}{\arg\max} \pi(p, q) \tag{19}$$

In practice, these optimal static controls are obtained via a grid search over (p, q). This provides a principled, tractable baseline grounded in the equilibrium analysis of the mean–field model. Although (p^*, q^*) is computed under deterministic mean–field equilibrium conditions, its actual performance in a stochastic networked environment may deviate due to randomness in adoption trajectories. To quantify this effect, the static pair (p^*, q^*) is deployed in the original GPU–accelerated stochastic simulator using the same unrolled interaction loop as in the TD3ES experiments. The simulator is then run for the full horizon of T steps, and the realized discounted profit is recorded in exactly the same manner as for the TD3ES agent.

Figure 11 presents the profit distributions of TD3ES and the static policy across test problems with Erdős–Rényi and exponentially distributed node degrees. The results indicate that TD3ES consistently outperforms the static policy, underscoring how a sequential decision-making framework can enhance traditional approaches to pricing and advertising intensity determination.

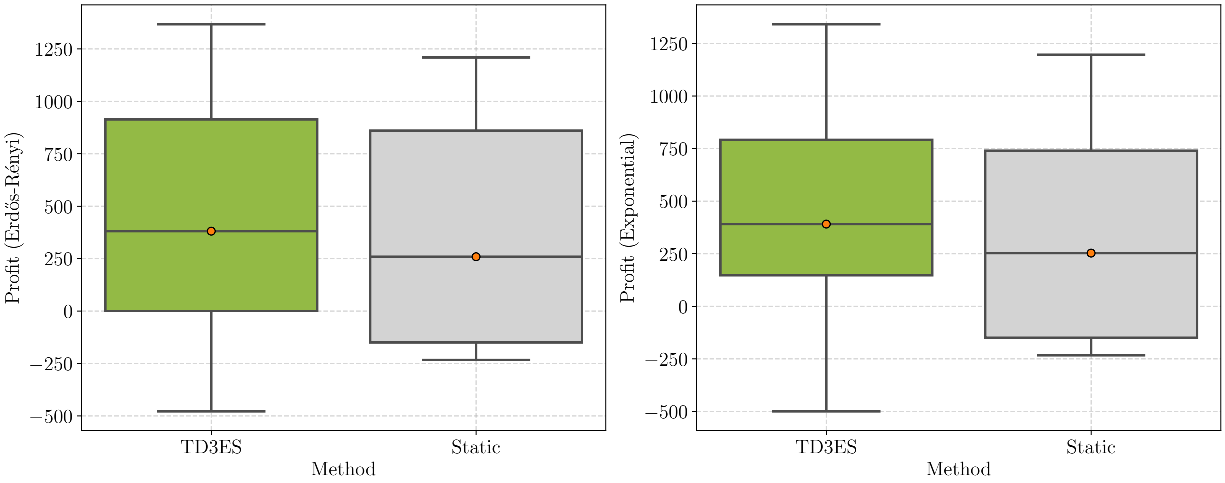

Figure 11: Profit distributions of TD3ES and the static policy across test problems with Erdős–Rényi and exponentially distributed node degrees.

7. Insights and Findings

This section examines key findings concerning TD3ES. Specifically, it analyzes (i) the TD3ES' output convergence when jointly setting price and advertising intensity, and (ii) the network's behavior in determining its actions.

7.1. Convergent Behavior of TD3ES in Joint Pricing and Advertising Decisions

When the TD3ES controller is deployed on a consumer network, the state of that network evolves in direct response to the price and advertising intensity selected by the controller. Simultaneously, TD3ES continually refines those two control variables by observing the network state in real time, thereby creating a closed-loop feedback system. The coupled dynamics between network composition and control actions almost invariably converge to a fixed point at which neither the network state nor the control policy exhibits meaningful change.

Figure 12 documents this convergence for two structurally distinct graphs - an Erdős-Rényi random network and a network with an exponential degree distribution - while keeping all other problem parameters identical. The figure traces (i) the temporal profiles of the price and advertising intensity chosen by TD3ES, (ii) the proportion of individuals residing in each compartment of the CDM Markov model, and (iii) the basic reproduction number, $\mathcal{R}_0$, whose value is modulated jointly by price and advertising intensity. In both topologies the three sets of trajectories stabilize, indicating that TD3ES has identified a stationary policy and that the network has settled into equilibrium.

A salient feature of this equilibrium in the networks of Figure 12 is the persistence of small yet non-zero fractions of nodes in the exposed (E) and owner (O) compartments. Such residual activity is theoretically expected whenever $\mathcal{R}_0 \geq 1$. Comparable convergence behavior has been observed across a broad spectrum of consumer network configurations, underscoring the robustness of TD3ES in steering complex diffusion processes toward stable operational regimes.

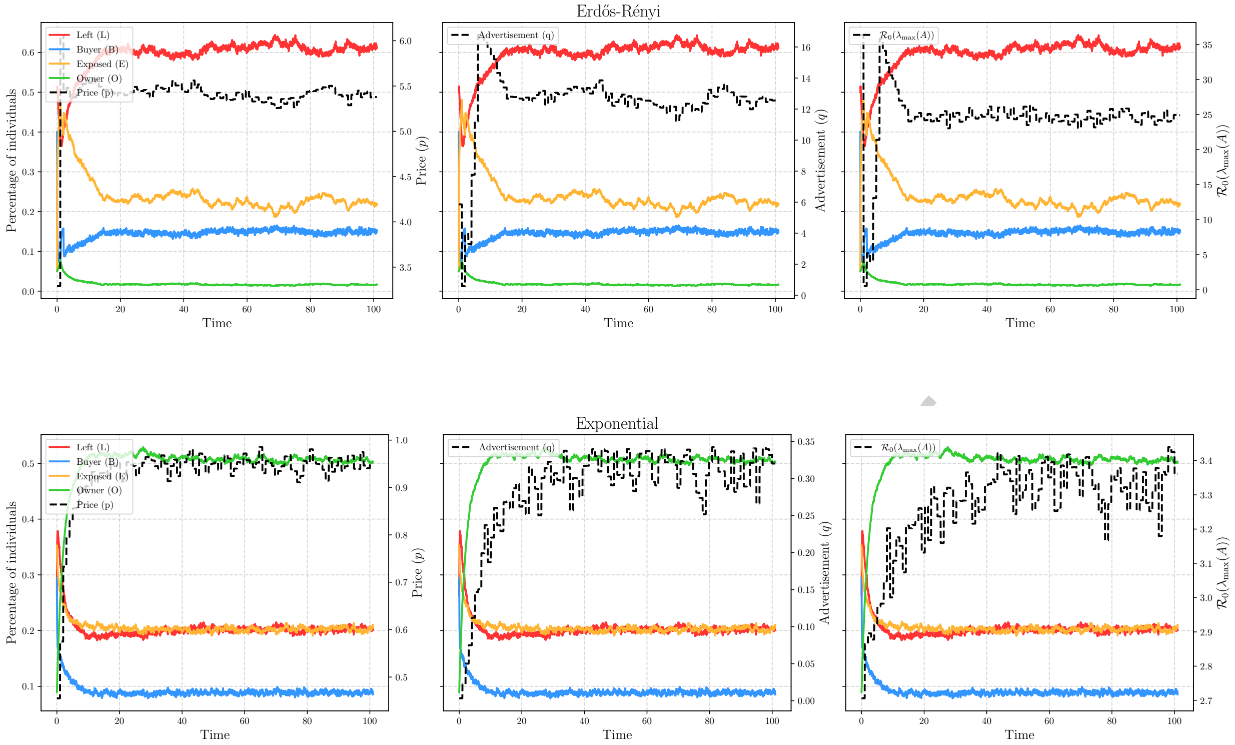

Figure 12: Price p, advertisement intensity q, and reproduction number $\mathcal{R}_0$ for two randomly generated Erdős-Rényi and exponentially distributed node degrees networks with parameters $N = 10,000, \alpha_1 = 2.86, \alpha_2 = 1.38, \beta = 1.63, \delta = 0.5, d_1 = 0.97, d_2 = 0.56, d_3 = 2.15, \bar{d} = 15$.

7.2. *Explaining TD3ES Decision-Making Behavior*

Section 5 introduced TD3ES as an RL framework that encodes the consumer network by reconstructing its node-level feature matrix, $\boldsymbol{X}$. Through this autoencoder, the algorithm obtains a compact latent representation that serves as its state description. Hence, identifying the factors that influence the reconstruction process is essential for understanding how TD3ES selects its actions.

To this end, the present work examines the element-wise reconstruction error, $\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|$, as an interpretability signal. For every node i, a reconstruction error score , $\|X_i - \hat{X}_i\|$ is computed, yielding an "error map" over the graph. Nodes that exhibit smaller errors are considered more faithfully captured by the embedding and therefore exert greater influence on the latent vector. Each row of $\boldsymbol{X}$ comprises five features: four one-hot indicators for the CDM Markov model compartments and the node's degree, normalized to $[0, 1]$.

Figure 13 compares reconstruction-error profiles for two synthetic topologies: an Erdős-Rényi graph and a graph whose degrees follow an exponential distribution. The horizontal axis reports node degree, while color denotes compartment membership. Several salient patterns emerge:

- **Topology-dependent error structure.** The global shape of the error distribution differs markedly between the two graph families, indicating that the autoencoder adapts to macroscopic degree statistics when forming its latent space.
- **Compartment separation.** In both networks, nodes cluster by compartment, confirming that the encoder is sensitive to compartment labels embedded in the feature matrix. Within each compartment, however, the ordering of errors is governed by network structure.

- **Erdős-Rényi graphs: preference for typical degrees.** In the Erdős-Rényi case, nodes whose degree lies near the ensemble average attain the lowest reconstruction error. Because such nodes dominate the training data, the autoencoder learns to represent them most accurately, causing TD3ES to weight their latent coordinates more heavily during policy evaluation.
- **Exponential graphs: dominance of high-degree nodes in the majority compartment.** Exponential graphs possess a long-tailed degree distribution. Here, high-degree nodes belonging to the largest compartment receive the smallest errors, whereas equally high-degree nodes in minority compartments incur substantially larger errors. Consequently, the embedding is biased toward hubs that are also members of the majority compartment.

These findings imply a topology-specific attention mechanism in TD3ES. In Erdős-Rényi environments, the policy is most influenced by nodes of average degree across all compartments. In contrast, when the underlying network is exponentially distributed, the policy focuses on hub nodes that reside in the most populous compartment. This behavioral shift, driven by the autoencoder's reconstruction dynamics, guides the agent's subsequent price and advertising decisions.

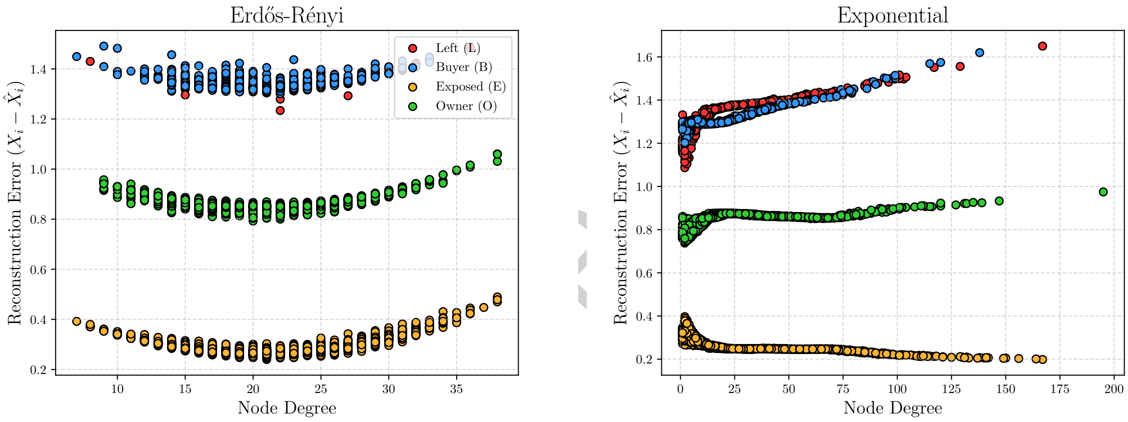

Figure 13: Node feature reconstruction error for two randomly generated Erdős-Rényi and exponentially distributed node degrees networks with parameters $N = 10,000, \alpha_1 = 3.10, \alpha_2 = 2.76, \beta = 2.57, \delta = 2.71, d_1 = 2.91, d_2 = 3.59, d_3 = 3.44, \bar{d} = 21$.

Figure 14 illustrates how the node-level reconstruction error evolves in a synthetically generated network whose degree sequence follows an exponential distribution, while prices are adjusted over time by the TD3ES controller. The three panels in the upper row correspond, from left to right, to the initial time step, the midpoint of the decision horizon, and the terminal step.

At the outset of the process most vertices occupy the L and B compartments. Because the majority of information needed for an accurate summary of the graph resides in those two groups, TD3ES assigns them greater importance, leading to noticeably lower reconstruction errors. In tandem, the algorithm sets a comparatively low price, reflecting its objective to stimulate adoption in the most populous compartments.

As time advances, the state distribution of nodes drifts: membership in the E and O compartments rises, while the share of L and B nodes diminishes. TD3ES responds by reallocating representational capacity toward the newly dominant compartments, a shift visible in the declining reconstruction errors assigned to E and O nodes and the increase for L and B nodes. By approximately the midpoint of the horizon, the system reaches a quasi-equilibrium in which the spatial pattern of reconstruction error stabilizes; subsequent changes are marginal and largely reflect small stochastic fluctuations rather than substantive strategic revisions.

Taken together, the evidence demonstrates that TD3ES integrates both global topology and the evolving compartment size distribution when embedding the graph. The adaptive focus on the currently predominant compartment enables the encoder–decoder pair to maintain a compact yet informative latent space, ensuring that pricing decisions remain aligned with the most influential portions of the network throughout the diffusion process.

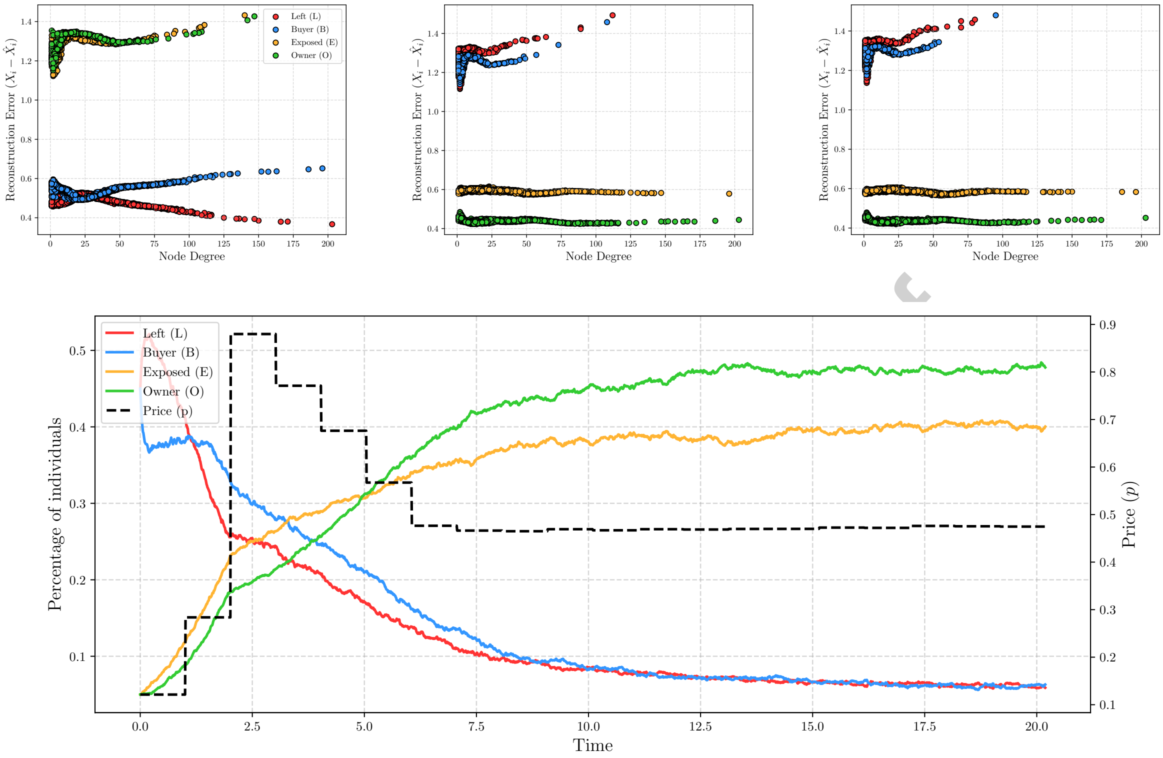

Figure 14: Node feature reconstruction error for a randomly generated network with exponentially distributed node degrees with parameters $N = 10,000, \alpha_1 = 3.37, \alpha_2 = 2.98, \beta = 0.16, \delta = 3.73, d_1 = 3.24, d_2 = 1.88, d_3 = 2.33, \bar{d} = 19$. Node construction error changes as TD3ES sets the price and network evolves in time.

8. Conclusion

This study has proposed, analyzed, and computationally validated an integrated framework for jointly steering price and advertising decisions in networked consumer markets. A stochastic compartmental model grounded in the consumer decision-making model (CDM) paradigm was formulated, and a deterministic mean-field approximation enabled tractable equilibrium and stability analysis. In particular, a trade-free equilibrium (TFE) was derived in closed form, and a reproduction number threshold condition established precise parameter regions under which markets self-extinguish or sustain adoption.

Building on these analytical foundations, the research introduced a twin delayed deep deterministic policy gradient with encoded state (TD3ES), an RL controller augmented with a graph convolutional autoencoder. The autoencoder compresses high-dimensional, dynamically evolving purchase networks into a compact latent representation, thereby allowing the actor-critic architecture to remain scalable while remaining sensitive to topological heterogeneity. A custom GPU-accelerated stochastic simulator further ensured that policy learning remained computationally feasible on realistically sized graphs.

Extensive numerical experiments on both homogeneous Erdős-Rényi and heterogeneous exponential networks confirmed that TD3ES reliably converges to profit-maximizing joint policies and, on structurally diverse graphs, significantly outperforms a TD3 baseline lacking structural awareness. Error attribution analyses showed that the autoencoder automatically shifts attention toward the currently dominant CDM

Markov model compartment and, in heavy-tailed graphs, toward influential hubs, thereby explaining the controller's superior profitability under realistic heterogeneity.

In summary, the evaluation of methods, based on total profit as the key performance metric, reveals that TD3ES provides statistically significant improvements over TD3 in exponential network structures ($p \leq 0.001$). This advantage stems from the algorithm's ability to identify and exploit highly connected nodes, which play a disproportionate role in driving adoption. While TD3 performs adequately in more homogeneous networks, the evidence indicates that TD3ES is especially valuable when decision makers operate in environments characterized by heavy-tailed degree distributions, such as social or consumer networks.

Beyond methodological contributions, several managerial insights emerge. First, optimal pricing and advertising are path-dependent: they must react not only to aggregate adoption levels but also to evolving network composition. Second, ignoring network structure can leave substantial revenue unrealized, especially in markets where influence is concentrated among a minority of highly connected consumers. Third, GPU-enabled, structure-aware RL now renders such adaptive strategies operationally viable at scale.

Important limitations also warrant acknowledgment. This study relies on controlled numerical experiments rather than real-world datasets, which may constrain the external validity of the findings. Although the experimental settings were carefully designed to reflect empirically observed patterns (e.g., exponential degree distributions in social networks), additional validation on real marketing datasets is crucial for confirming the robustness and applicability of the results. Moreover, the current model assumes a single monopolistic seller, a static network topology, and complete observability of consumer states. Another limitation is that the reinforcement learning component is instantiated only with the TD3 backbone in order to provide a controlled ablation between encoder-augmented and non-augmented variants. While this isolates the contribution of the proposed graph-based state representation, it also means that the study does not report results across other RL families, such as SAC or PPO. Future research could therefore extend the analysis by benchmarking multiple RL backbones, each with and without the encoder, to demonstrate the generalizability of the representation. Beyond this, future studies may relax further assumptions by incorporating competing firms, dynamically rewiring social ties, partial observability, or richer consumer heterogeneity. Empirical calibration on real datasets would also be valuable for testing the theoretical thresholds and for quantifying welfare implications in practical contexts.

In sum, the present work demonstrates that coupling network-aware RL with rigorous diffusion modeling furnishes a powerful and interpretable tool for dynamic marketing control. The approach unifies analytical insight with scalable computation and thus charts a promising avenue for data-driven management of influence, pricing, and promotion in increasingly connected consumer ecosystems.

References

[1] A. Stankevich, Explaining the consumer decision-making process: Critical literature review, Journal of international business research and marketing 2 (6) (2017).

[2] D. Ge, Value pricing in presence of network effects, Journal of Product & Brand Management 11 (3) (2002) 174–185.

[3] D. P. Kumar, K. V. Raju, The role of advertising in consumer decision making, IOSR Journal of Business and Management 14 (4) (2013) 37–45.

[4] C. Chih-Chung, C. Chang, L. W.-C. Lin, et al., The effect of advertisement frequency on the advertisement attitude-the controlled effects of brand image and spokesperson's credibility, Procedia-Social and behavioral sciences 57 (2012) 352–359.

[5] B. J. Ali, G. Anwar, Marketing strategy: Pricing strategies and its influence on consumer purchasing decision, Ali, BJ, & Anwar, G.(2021). Marketing Strategy: Pricing strategies and its influence on consumer purchasing decision. International journal of Rural Development, Environment and Health Research 5 (2) (2021) 26–39.

[6] M. Ameri, E. Honka, Y. Xie, Word of mouth, observed adoptions, and anime-watching decisions: The role of the personal vs. the community network, Marketing Science (2019).

[7] M. L. Katz, C. Shapiro, Network externalities, competition, and compatibility, The American Economic Review 75 (3) (1985) 424–440.
URL http://www.jstor.org/stable/1814809

[8] A. Banerji, B. Dutta, Local network externalities and market segmentation, International Journal of Industrial Organization 27 (5) (2009) 605 – 614. doi:https://doi.org/10.1016/j.ijindorg.2009.02.001.
URL http://www.sciencedirect.com/science/article/pii/S0167718709000228

[9] J. Meyners, C. Barrot, J. U. Becker, A. V. Bodapati, Reward-scrounging in customer referral programs, International Journal of Research in Marketing 34 (2) (2017) 382–398.

[10] R. Bapna, A. Umyarov, Do your online friends make you pay? a randomized field experiment on peer influence in online social networks, Management Science 61 (8) (2015) 1902–1920.

[11] A. Ajorlou, A. Jadbabaie, A. Kakhbod, Dynamic pricing in social networks: The word-of-mouth effect, Management Science 64 (2) (2016) 971–979.

[12] P. Van Mieghem, J. Omic, R. Kooij, Virus spread in networks, IEEE/ACM Transactions on Networking (TON) 17 (1) (2009) 1–14.

[13] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[14] J. Farrell, G. Saloner, Standardization, compatibility, and innovation, The RAND Journal of Economics 16 (1) (1985) 70–83.
URL http://www.jstor.org/stable/2555589

[15] O. Candogan, K. Bimpikis, A. Ozdaglar, Optimal pricing in networks with externalities, Operations Research 60 (4) (2012) 883–905.

[16] P. Saaskilahti, Monopoly pricing of social goods, International Journal of the Economics of Business 22 (3) (2015) 429–448. doi:10.1080/13571516.2015.1008731.
URL https://doi.org/10.1080/13571516.2015.1008731

[17] Z. Cao, X. Chen, X. Hu, C. Wang, Approximation algorithms for pricing with negative network externalities, Journal of Combinatorial Optimization 33 (2) (2017) 681–712.

[18] A. Gramstad, Nonlinear pricing with local network effects, Available at SSRN 2597638 (2016).

[19] A. Jadbabaie, A. Kakhbod, Optimal contracting in networks, Journal of Economic Theory 183 (2019) 1094–1153. doi:https://doi.org/10.1016/j.jet.2019.07.017.
URL https://www.sciencedirect.com/science/article/pii/S002205311930081X

[20] G. De Giorgi, A. Frederiksen, L. Pistaferri, Consumption network effects, The Review of Economic Studies 87 (1) (2020) 130–163.

[21] S. Raghavan, R. Zhang, A branch-and-cut approach for the weighted target set selection problem on social networks, INFORMS Journal on Optimization 1 (4) (2019) 304–322.

[22] R. Zhang, X. Wang, S. Pei, Targeted influence maximization in complex networks, Physica D: Nonlinear Phenomena 446 (2023) 133677.

[23] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, G. J. Pappas, Optimal resource allocation for network protection against spreading processes, IEEE Transactions on Control of Network Systems 1 (1) (2014) 99–108.

[24] N. J. Watkins, C. Nowzari, G. J. Pappas, Robust economic model predictive control of continuous-time epidemic processes, IEEE Transactions on Automatic Control 65 (3) (2019) 1116–1131.

[25] M. C. Cohen, P. Harsha, Designing price incentives in a network with social interactions, Manufacturing & Service Operations Management 22 (2) (2020) 292–309.

[26] F. Bloch, N. Quérou, Pricing in social networks, Games and economic behavior 80 (2013) 243–261.

[27] F. D. Sahneh, C. M. Scoglio, Optimal information dissemination in epidemic networks, in: 2012 ieee 51st IEEE conference on decision and control (cdc), IEEE, 2012, pp. 1657–1662.

[28] R.-C. Bayer, M. Chan, Network externalities, demand inertia and dynamic pricing in an experimental oligopoly market, Economic Record 83 (263) (2007) 405–415.

[29] S. Alizamir, N. Chen, S.-H. Kim, V. Manshadi, Impact of network structure on new service pricing, Mathematics of Operations Research 47 (3) (2022) 1999–2033.

[30] M. Newman, Networks, 2nd Edition, Oxford University Press, 2018.

[31] I. Z. Kiss, J. C. Miller, P. L. Simon, et al., Mathematics of epidemics on networks, Vol. 598, Springer, 2017.

[32] F. D. Sahneh, A. Vajdi, H. Shakeri, F. Fan, C. Scoglio, Gemfsim: A stochastic simulator for the generalized epidemic modeling framework, Journal of computational science 22 (2017) 36–44.

[33] M. H. Samaei, F. D. Sahneh, C. Scoglio, Fastgemf: Scalable high-speed simulation of stochastic spreading processes over complex multilayer networks, IEEE Access (2025).

[34] M. Fey, J. E. Lenssen, Fast graph representation learning with pytorch geometric, arXiv preprint arXiv:1903.02428 (2019).

[35] M. R. Mendonça, A. M. Barreto, A. Ziviani, Efficient information diffusion in time-varying graphs through deep reinforcement learning, World Wide Web 25 (6) (2022) 2535–2560.

[36] L. Ling, W. U. Mondal, S. V. Ukkusuri, Cooperating graph neural networks with deep reinforcement learning for vaccine prioritization, IEEE Journal of Biomedical and Health Informatics (2024).

[37] Y. Chen, X. M. Chen, A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction, Transportation Research Part C: Emerging Technologies 143 (2022) 103820.

[38] X. Zhao, C. Gu, H. Zhang, X. Yang, X. Liu, J. Tang, H. Liu, Dear: Deep reinforcement learning for online advertising impression in recommender systems, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 750–758.

[39] V. Singh, B. Nanavati, A. K. Kar, A. Gupta, How to maximize clicks for display advertisement in digital marketing? a reinforcement learning approach, Information Systems Frontiers 25 (4) (2023) 1621–1638.

[40] S. Tian, S. Mo, L. Wang, Z. Peng, Deep reinforcement learning-based approach to tackle topic-aware influence maximization, Data Science and Engineering 5 (2020) 1–11.

[41] L. Ma, Z. Shao, X. Li, Q. Lin, J. Li, V. C. Leung, A. K. Nandi, Influence maximization in complex networks by using evolutionary deep reinforcement learning, IEEE Transactions on Emerging Topics in Computational Intelligence 7 (4) (2022) 995–1009.

[42] D. Vakratsas, F. M. Feinberg, F. M. Bass, G. Kalyanaram, The shape of advertising response functions revisited: A model

of dynamic probabilistic thresholds, Marketing Science 23 (1) (2004) 109–119.

[43] S. Cowan, Third-degree price discrimination and consumer surplus, The Journal of Industrial Economics 60 (2) (2012) 333–345.

[44] M. Mrázová, J. P. Neary, Io for exports (s), International Journal of Industrial Organization 70 (2020) 102561.

[45] Y. Wan, T. Kober, M. Densing, Nonlinear inverse demand curves in electricity market modeling, Energy Economics (2022) 105809.

[46] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, Journal of computational physics 22 (4) (1976) 403–434.

[47] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, The journal of physical chemistry 81 (25) (1977) 2340–2361.

[48] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International conference on machine learning, PMLR, 2018, pp. 1587–1596.

[49] L. Li, T. J. Walsh, M. L. Littman, Towards a unified theory of state abstraction for mdps., AI&M 1 (2) (2006) 3.

Appendix A. Additional Tables

Table A.3: Erdős-Rényi network results

				$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
4	(0.5, 0.5)	(0.5, 0.5, 0.5)	0.2	-14.98	**-5.64**	-14.49	**-4.74**	**-11.11**	-86.79
			1.5	-275.93	**-3.72**	-272.17	**-4.01**	-269.90	**-0.72**
			2.5	-120.35	**0.15**	-118.93	**0.15**	-127.11	**0.03**
		(0.5, 2.0, 2.0)	0.2	259.15	**262.92**	254.52	**256.69**	-0.03	**258.22**
			1.5	**391.76**	325.15	**391.30**	322.61	**392.00**	322.38
			2.5	**398.86**	0.18	**399.89**	68.74	**395.24**	21.58
		(2.5, 0.5, 0.5)	0.2	**80.59**	-476.77	**83.93**	-478.15	**68.73**	-476.18
			1.5	-81.71	**0.22**	-79.70	**-0.19**	-76.85	**0.08**
			2.5	-462.31	**0.52**	-451.55	**0.22**	-416.41	**3.22**
		(2.5, 2.0, 2.0)	0.2	**1127.46**	115.04	**1125.52**	120.86	**1124.78**	169.81
			1.5	1256.90	**1271.72**	1252.96	**1269.68**	1256.37	**1272.51**
			2.5	1211.59	**1228.03**	1217.27	**1258.96**	1212.86	**1224.53**
	(1.5, 1.5)	(0.5, 0.5, 0.5)	0.2	-12.76	**-1.68**	-10.48	**-7.03**	-11.82	**-2.40**
			1.5	0.03	**0.77**	-0.07	**0.42**	**7.74**	0.27
			2.5	**0.77**	0.58	**2.76**	-0.67	**0.41**	0.04
		(0.5, 2.0, 2.0)	0.2	0.70	**354.86**	0.55	**353.12**	0.04	**353.64**
			1.5	**1.98**	0.03	0.12	**8.73**	0.04	**2.74**
			2.5	**380.72**	0.00	**379.10**	0.00	**379.93**	0.13
		(2.5, 0.5, 0.5)	0.2	**585.04**	0.59	**583.42**	0.47	**582.71**	0.31
			1.5	-11.99	**0.60**	-11.09	**0.29**	-11.28	**0.19**
			2.5	-492.41	**0.20**	-491.47	**-0.18**	-493.60	**5.97**
		(2.5, 2.0, 2.0)	0.2	804.36	**1008.47**	840.93	**922.70**	823.82	**976.20**
			1.5	1050.68	**1276.35**	1047.65	**1278.79**	1059.35	**1265.52**
			2.5	1046.26	**1282.84**	1033.99	**1292.56**	1048.96	**1282.95**
	(2.5, 2.5)	(0.5, 0.5, 0.5)	0.2	**379.10**	-20.06	**380.03**	-24.24	**376.89**	-20.37
			1.5	**317.30**	0.09	**319.11**	0.10	**315.66**	-0.10
			2.5	0.89	**2.39**	0.34	**0.98**	**0.18**	0.03
		(0.5, 2.0, 2.0)	0.2	-0.00	**249.41**	-0.00	**255.14**	-0.00	**253.27**
			1.5	0.00	**392.32**	0.00	**389.79**	0.00	**398.73**
			2.5	**361.17**	0.00	**361.31**	0.00	**362.10**	0.00
		(2.5, 0.5, 0.5)	0.2	**593.78**	53.24	**593.22**	10.81	**594.02**	507.89

Continued on next page

Table A.3 – *Continued from previous page*

				$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
			1.5	**418.17**	388.31	**475.37**	-31.83	**428.46**	389.45
			2.5	**-11.14**	-52.50	**-11.31**	-68.99	**-10.38**	-168.42
		(2.5, 2.0, 2.0)	0.2	-4.24	**790.51**	-4.01	**773.86**	-4.10	**774.13**
			1.5	857.44	**1269.52**	858.35	**1271.09**	862.35	**1271.73**
			2.5	780.62	**1272.86**	781.94	**1273.04**	788.71	**1276.08**
12	(0.5, 0.5)	(0.5, 0.5, 0.5)	0.2	-16.86	**384.20**	-15.60	**335.52**	-13.64	**379.98**
			1.5	**668.63**	-0.91	**654.25**	2.63	**681.86**	-0.23
			2.5	**740.76**	1.04	**747.08**	13.85	**745.35**	11.89
		(0.5, 2.0, 2.0)	0.2	291.88	**388.09**	287.49	**378.58**	285.62	**382.09**
			1.5	414.68	**414.92**	414.54	**415.11**	**413.97**	408.41
			2.5	**416.85**	287.47	**416.94**	252.29	**416.68**	248.31
		(2.5, 0.5, 0.5)	0.2	**683.56**	-447.29	**686.59**	-457.61	**684.05**	-455.55
			1.5	**406.82**	-0.04	**408.58**	-0.60	**407.98**	1.04
			2.5	**359.08**	191.13	**359.50**	0.48	**359.41**	229.87
		(2.5, 2.0, 2.0)	0.2	**1316.79**	1232.72	**1317.60**	1244.95	**1315.26**	1126.42
			1.5	**1362.64**	1326.17	**1362.38**	1334.68	**1361.50**	1319.79
			2.5	**1321.38**	1220.18	**1315.85**	1204.28	**1318.52**	1209.59
	(1.5, 1.5)	(0.5, 0.5, 0.5)	0.2	-4.98	**478.06**	-4.83	**463.21**	-5.84	**485.71**
			1.5	699.41	**796.10**	698.00	**795.61**	699.17	**792.95**
			2.5	627.19	**801.23**	633.39	**795.31**	629.12	**785.18**
		(0.5, 2.0, 2.0)	0.2	1.67	**388.20**	1.37	**386.84**	0.08	**385.53**
			1.5	1.03	**5.74**	0.06	**4.24**	0.11	**3.83**
			2.5	**403.48**	0.00	**403.19**	0.01	**403.39**	3.83
		(2.5, 0.5, 0.5)	0.2	**608.85**	0.73	**617.79**	3.23	**609.75**	0.07
			1.5	**774.88**	30.47	**750.58**	0.00	**771.04**	12.36
			2.5	-486.95	**534.01**	-483.85	**250.90**	-481.27	**-1.86**
		(2.5, 2.0, 2.0)	0.2	**1231.54**	1223.47	**1232.57**	1224.79	**1237.35**	1180.37
			1.5	1308.76	**1348.39**	1315.79	**1350.27**	1312.74	**1347.49**
			2.5	1170.41	**1357.46**	1166.40	**1352.00**	1163.83	**1354.61**
	(2.5, 2.5)	(0.5, 0.5, 0.5)	0.2	**420.03**	-284.69	**427.44**	-181.86	**415.57**	-341.09
			1.5	**387.89**	0.28	**401.38**	0.72	**375.02**	0.03
			2.5	7.61	**33.77**	0.73	**28.00**	**0.43**	0.08
		(0.5, 2.0, 2.0)	0.2	-0.00	**245.83**	-0.00	**252.99**	-0.00	**245.21**
			1.5	0.00	**340.13**	0.00	**340.96**	0.00	**365.21**
			2.5	**391.62**	0.00	**388.64**	0.09	**391.67**	0.00
		(2.5, 0.5, 0.5)	0.2	622.00	**725.75**	638.80	**820.49**	635.13	**802.15**
			1.5	598.98	**652.10**	**780.59**	549.03	**778.44**	612.53
			2.5	-14.38	**815.90**	-14.54	**818.63**	-12.37	**817.84**
		(2.5, 2.0, 2.0)	0.2	-6.29	**1025.23**	-5.97	**924.81**	14.56	**921.12**
			1.5	1223.39	**1326.89**	1219.84	**1324.38**	1223.63	**1326.77**
			2.5	1093.99	**1349.40**	1097.54	**1343.66**	1099.00	**1342.65**
20	(0.5, 0.5)	(0.5, 0.5, 0.5)	0.2	-18.13	**399.70**	-19.10	**354.69**	-14.74	**420.88**
			1.5	842.84	**842.89**	**842.93**	839.59	**848.03**	818.85
			2.5	**800.43**	638.43	**807.17**	788.09	**806.41**	711.23
		(0.5, 2.0, 2.0)	0.2	296.24	**406.80**	302.64	**410.40**	293.74	**402.06**
			1.5	**414.01**	402.93	**411.65**	377.55	**411.92**	386.88

Continued on next page

Table A.3 – *Continued from previous page*

$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
				TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
			2.5	**417.11**	397.68	**417.20**	397.13	**416.88**	252.00
		(2.5, 0.5, 0.5)	0.2	**706.33**	-417.24	**713.45**	-441.30	**706.61**	-414.37
			1.5	**462.80**	1.27	**464.15**	-0.46	**466.08**	18.41
			2.5	423.98	**690.13**	422.21	**723.96**	422.67	**693.40**
		(2.5, 2.0, 2.0)	0.2	**1334.46**	1309.92	**1335.57**	1303.23	**1336.09**	1297.04
			1.5	**1366.65**	1326.96	**1366.67**	1326.94	**1365.46**	1325.10
			2.5	**1320.84**	1202.02	**1317.04**	1190.58	**1317.20**	1194.17
	(1.5, 1.5)	(0.5, 0.5, 0.5)	0.2	-3.66	**544.10**	-2.82	**492.85**	-3.59	**519.50**
			1.5	**880.31**	832.55	**880.18**	842.91	**878.81**	853.13
			2.5	**931.40**	772.63	**933.06**	770.93	**932.56**	742.15
		(0.5, 2.0, 2.0)	0.2	10.06	**391.53**	8.90	**389.85**	0.11	**388.98**
			1.5	0.74	**23.63**	0.06	**4.18**	0.26	**3.99**
			2.5	**402.96**	0.00	**402.83**	0.05	**402.37**	2.91
		(2.5, 0.5, 0.5)	0.2	608.90	**860.12**	**604.47**	-0.18	609.09	**742.02**
			1.5	**929.13**	670.94	**930.59**	659.88	**923.38**	634.92
			2.5	-478.29	**481.73**	-477.23	**538.90**	-479.40	**528.12**
		(2.5, 2.0, 2.0)	0.2	**1315.99**	1193.04	**1315.89**	1212.38	**1315.60**	1276.23
			1.5	1344.19	**1363.35**	1343.55	**1362.83**	1347.38	**1363.89**
			2.5	1159.45	**1361.05**	1154.78	**1358.78**	1176.08	**1358.70**
	(2.5, 2.5)	(0.5, 0.5, 0.5)	0.2	**422.45**	-422.58	**423.49**	-424.50	**420.65**	-446.19
			1.5	**416.97**	0.37	**436.86**	3.92	**428.60**	0.04
			2.5	10.21	**453.82**	7.91	**462.49**	**4.01**	0.11
		(0.5, 2.0, 2.0)	0.2	-0.00	**235.54**	-0.00	**236.43**	-0.00	**183.02**
			1.5	0.00	**376.16**	0.00	**383.11**	0.00	**377.52**
			2.5	**386.57**	0.01	**385.39**	49.70	**383.53**	0.02
		(2.5, 0.5, 0.5)	0.2	620.67	**1026.28**	610.95	**929.52**	614.76	**979.80**
			1.5	627.01	**1056.08**	868.47	**1057.45**	856.70	**1021.90**
			2.5	-16.83	**911.46**	-17.55	**906.50**	-14.58	**906.95**
		(2.5, 2.0, 2.0)	0.2	-5.50	**1149.40**	-5.24	**1207.92**	14.91	**1240.94**
			1.5	1246.92	**1329.45**	1257.03	**1328.65**	1257.29	**1328.99**
			2.5	1103.50	**1366.39**	1101.32	**1367.57**	1104.73	**1364.84**

Table A.4: Exponential network results

$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
				TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
4	(0.5, 0.5)	(0.5, 0.5, 0.5)	0.20	-13.24	**-0.68**	-12.90	**-1.22**	-13.17	**-0.07**
			1.50	0.44	**321.93**	8.91	**347.35**	24.71	**344.11**
			2.50	0.21	**277.45**	14.38	**241.29**	**403.73**	149.87
		(0.5, 2.0, 2.0)	0.20	158.64	**271.40**	150.96	**262.13**	-185.64	**267.53**
			1.50	0.74	**316.41**	-0.02	**320.61**	0.01	**312.57**
			2.50	-0.00	**393.47**	-0.00	**398.86**	0.00	**396.70**
		(2.5, 0.5, 0.5)	0.20	**0.48**	-145.28	**0.33**	-155.93	**0.15**	-194.25

Continued on next page

Table A.4 – *Continued from previous page*

				$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
			1.50	**517.83**	191.63	**520.72**	241.69	**519.97**	202.94
			2.50	**469.80**	183.13	**469.91**	197.61	**471.77**	180.49
		(2.5, 2.0, 2.0)	0.20	**1159.82**	1133.29	1166.46	**1173.25**	**1160.54**	1150.20
			1.50	1262.84	**1272.85**	1261.56	**1272.62**	1254.35	**1264.84**
			2.50	1224.21	**1242.65**	1223.09	**1248.24**	1218.59	**1241.20**
	(1.5, 1.5)	(0.5, 0.5, 0.5)	0.20	-55.08	**-28.99**	-38.55	**-29.44**	**-18.98**	-26.72
			1.50	**61.34**	7.79	**78.83**	8.13	**73.54**	17.95
			2.50	328.04	**381.56**	**320.67**	44.43	318.15	**389.13**
		(0.5, 2.0, 2.0)	0.20	-70.50	**133.83**	-78.82	**88.19**	-76.63	**123.77**
			1.50	**390.51**	331.39	**392.80**	346.18	**390.81**	324.86
			2.50	**403.15**	248.19	**402.50**	247.42	**401.76**	237.75
		(2.5, 0.5, 0.5)	0.20	0.44	**393.28**	0.31	**381.73**	0.13	**386.98**
			1.50	481.96	**526.84**	481.35	**519.62**	489.37	**521.39**
			2.50	13.28	**185.66**	-10.46	**140.70**	-0.40	**489.10**
		(2.5, 2.0, 2.0)	0.20	1140.94	**1243.98**	1150.31	**1254.98**	-1.62	**1249.75**
			1.50	1199.41	**1267.66**	1197.05	**1266.94**	1207.79	**1271.91**
			2.50	1235.83	**1239.49**	1239.03	**1239.86**	**1234.81**	1231.06
	(2.5, 2.5)	(0.5, 0.5, 0.5)	0.20	**-102.44**	-499.27	**-101.06**	-499.30	**-100.96**	-499.32
			1.50	**-18.93**	-427.56	**8.57**	-428.79	**61.63**	-428.70
			2.50	**273.28**	-100.32	**274.08**	-75.85	**275.92**	98.23
		(0.5, 2.0, 2.0)	0.20	**90.32**	0.01	**84.50**	0.01	**73.99**	0.35
			1.50	**381.63**	0.10	**379.42**	4.48	**374.71**	3.20
			2.50	0.22	**317.37**	0.03	**317.63**	0.01	**317.12**
		(2.5, 0.5, 0.5)	0.20	**-17.62**	-31.22	**-17.60**	-29.20	**0.44**	-24.09
			1.50	**248.09**	-172.48	**251.53**	-163.58	**249.04**	-164.86
			2.50	**323.42**	-69.15	**342.59**	-72.28	**312.08**	-66.20
		(2.5, 2.0, 2.0)	0.20	-31.13	**691.12**	-30.69	**646.32**	-29.45	**667.07**
			1.50	272.40	**1099.18**	272.57	**1117.73**	272.92	**1124.73**
			2.50	930.82	**1169.82**	932.27	**1171.84**	930.83	**1177.26**
12	(0.5, 0.5)	(0.5, 0.5, 0.5)	0.20	**506.51**	440.85	**522.97**	462.39	**527.64**	471.16
			1.50	3.47	**533.31**	**585.46**	519.43	**582.97**	512.74
			2.50	11.45	**284.93**	**646.42**	290.70	**646.54**	283.73
		(0.5, 2.0, 2.0)	0.20	192.32	**344.17**	189.02	**342.60**	187.94	**346.88**
			1.50	1.34	**359.99**	-0.02	**360.20**	0.01	**343.34**
			2.50	-0.00	**402.20**	-0.00	**408.20**	0.00	**398.98**
		(2.5, 0.5, 0.5)	0.20	0.65	**211.95**	0.49	**165.16**	0.36	**188.40**
			1.50	656.35	**664.06**	657.59	**666.00**	**654.75**	581.84
			2.50	493.53	**691.48**	496.46	**693.67**	498.22	**675.73**
		(2.5, 2.0, 2.0)	0.20	**1247.07**	1239.22	**1249.09**	1240.99	**1242.85**	1233.17
			1.50	1289.28	**1326.33**	1288.98	**1327.30**	1284.20	**1321.58**
			2.50	1243.68	**1276.91**	1238.84	**1300.33**	1238.07	**1283.64**
	(1.5, 1.5)	(0.5, 0.5, 0.5)	0.20	**502.31**	-61.49	**497.52**	-66.11	**497.67**	-57.27
			1.50	566.84	**676.87**	565.15	**665.91**	569.22	**685.45**
			2.50	406.50	**681.92**	401.33	**669.94**	401.04	**671.96**
		(0.5, 2.0, 2.0)	0.20	1.59	**247.67**	0.85	**249.15**	1.83	**248.75**
			1.50	**405.25**	359.20	**405.26**	359.06	**405.33**	364.43

Continued on next page

Table A.4 – *Continued from previous page*

				$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
			2.50	**412.99**	258.41	**413.14**	249.84	**412.60**	223.40
		(2.5, 0.5, 0.5)	0.20	0.53	**522.51**	0.43	**511.78**	0.28	**493.00**
			1.50	490.87	**686.97**	505.39	**688.43**	496.05	**689.21**
			2.50	-9.56	**713.50**	5.21	**713.04**	9.43	**718.32**
		(2.5, 2.0, 2.0)	0.20	1226.86	**1262.44**	1230.44	**1271.88**	-1.68	**1253.09**
			1.50	1299.18	**1326.27**	1300.58	**1327.09**	1301.08	**1327.23**
			2.50	1301.16	**1318.16**	1302.45	**1316.09**	1299.05	**1319.25**
	(2.5, 2.5)	(0.5, 0.5, 0.5)	0.20	**-178.25**	-499.40	**-173.46**	-499.48	**-181.36**	-499.49
			1.50	**337.18**	-425.62	**339.13**	-425.41	**341.16**	-424.42
			2.50	356.39	**667.82**	358.34	**679.94**	357.84	**667.44**
		(0.5, 2.0, 2.0)	0.20	**86.12**	0.01	**85.55**	0.01	**90.18**	0.05
			1.50	**403.63**	95.30	**401.41**	64.13	**402.81**	116.85
			2.50	0.64	**353.44**	0.04	**347.68**	0.01	**346.21**
		(2.5, 0.5, 0.5)	0.20	**-16.66**	-37.05	-16.21	**720.83**	626.12	**730.84**
			1.50	**345.19**	-189.19	**345.85**	-175.91	**343.37**	-175.47
			2.50	**367.71**	-83.69	**318.13**	-76.30	**333.80**	327.71
		(2.5, 2.0, 2.0)	0.20	-31.22	**838.93**	-30.27	**767.39**	-29.61	**827.34**
			1.50	614.47	**1202.51**	617.88	**1211.09**	619.34	**1194.34**
			2.50	1026.20	**1224.65**	1022.12	**1223.53**	1022.72	**1230.24**
20	(0.5, 0.5)	(0.5, 0.5, 0.5)	0.20	626.40	**641.00**	621.32	**641.76**	626.78	**636.69**
			1.50	**653.06**	577.63	**660.74**	588.75	**656.53**	585.06
			2.50	**724.98**	355.60	**742.26**	361.09	**741.78**	370.85
		(0.5, 2.0, 2.0)	0.20	200.27	**366.38**	199.78	**367.35**	195.54	**366.28**
			1.50	1.61	**383.47**	-0.01	**374.86**	0.01	**364.60**
			2.50	-0.00	**394.35**	-0.00	**392.95**	0.00	**403.93**
		(2.5, 0.5, 0.5)	0.20	2.68	**353.70**	291.97	**354.94**	295.88	**359.45**
			1.50	696.35	**726.67**	697.11	**730.31**	696.08	**734.39**
			2.50	501.67	**771.09**	506.92	**753.97**	506.52	**751.47**
		(2.5, 2.0, 2.0)	0.20	**1266.23**	1253.06	**1268.58**	1261.23	**1262.45**	1246.66
			1.50	1300.61	**1340.66**	1293.78	**1340.67**	1289.77	**1337.98**
			2.50	1242.48	**1296.51**	1242.87	**1296.20**	1239.05	**1298.38**
	(1.5, 1.5)	(0.5, 0.5, 0.5)	0.20	**589.43**	-84.59	**593.99**	-83.96	**609.54**	-58.22
			1.50	587.83	**722.35**	604.73	**761.22**	579.66	**700.57**
			2.50	418.33	**749.56**	420.19	**774.05**	422.63	**751.18**
		(0.5, 2.0, 2.0)	0.20	15.21	**278.81**	19.50	**303.56**	16.60	**290.53**
			1.50	**408.41**	369.45	**408.29**	368.17	**407.96**	368.40
			2.50	**414.49**	250.34	**414.82**	231.25	**413.62**	236.17
		(2.5, 0.5, 0.5)	0.20	0.73	**603.17**	1.03	**616.66**	235.91	**598.36**
			1.50	527.55	**784.41**	499.62	**752.26**	520.74	**778.70**
			2.50	3.42	**802.05**	3.65	**800.94**	-6.36	**788.59**
		(2.5, 2.0, 2.0)	0.20	1249.18	**1285.49**	1252.30	**1292.24**	-1.61	**1241.34**
			1.50	1325.60	**1341.28**	1320.63	**1336.88**	1321.92	**1339.07**
			2.50	1316.10	**1337.45**	1316.98	**1337.74**	1314.19	**1337.12**
	(2.5, 2.5)	(0.5, 0.5, 0.5)	0.20	**-171.16**	-499.48	**271.85**	-499.54	**-172.63**	-499.56
			1.50	**418.30**	-421.30	**422.05**	-421.24	**404.39**	-421.00
			2.50	372.02	**778.78**	371.29	**770.96**	367.93	**706.76**

Continued on next page

Table A.4 – *Continued from previous page*

				$(L\%, B\%, E\%, O\%)$					
				$(0.1, 0.1, 0.5, 0.3)$		$(0.3, 0.2, 0.4, 0.1)$		$(0.5, 0.3, 0.1, 0.1)$	
$\bar{d}$	(α_1, α_2)	(d_1, d_2, d_3)	β	TD3	TD3ES	TD3	TD3ES	TD3	TD3ES
		(0.5, 2.0, 2.0)	0.20	**78.90**	0.00	**85.65**	0.01	**86.56**	0.04
			1.50	**406.85**	107.28	**407.70**	116.04	**404.62**	89.78
			2.50	0.93	**368.32**	0.06	**365.90**	0.02	**363.08**
		(2.5, 0.5, 0.5)	0.20	-17.26	**898.16**	-18.03	**905.22**	828.98	**900.51**
			1.50	**368.01**	-176.24	**364.84**	-189.27	**373.67**	-174.37
			2.50	**279.05**	-77.43	351.40	**479.77**	301.75	**515.63**
		(2.5, 2.0, 2.0)	0.20	-31.28	**885.19**	-29.71	**719.19**	-29.50	**857.46**
			1.50	660.40	**1203.78**	664.02	**1203.02**	663.95	**1236.20**
			2.50	1048.32	**1236.86**	1059.74	**1212.65**	1055.38	**1227.85**

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: