LEAP: Learning Evolutionary Advantage Policies for Fixation Maximization via Deep Reinforcement Learning

Smita Ghosh*
sghosh3@scu.edu
Santa Clara University
Santa Clara, CA, United States

Shivvrat Arya*
shivvrat.arya@njit.edu
New Jersey Institute of Technology
Newark, NJ, United States

Abstract

Diffusion processes on networks are central to social network analysis and data mining, where influence maximization provides a canonical framework for identifying influential nodes. However, many real-world diffusion processes are competitive and reversible, making classical irreversible influence maximization models inadequate. The positional Moran process captures such dynamics by modeling stochastic competition with position-dependent fitness and motivates the fixation maximization problem: selecting an initial active set that maximizes the probability of eventual population-wide dominance. Existing methods rely on static structural heuristics or simulation-intensive greedy search. These approaches either ignore long-horizon evolutionary effects or scale poorly because they require repeated stochastic simulations. This limitation raises a fundamental question: *Can we learn scalable active node selection policies that capture the evolutionary dynamics governing fixation?* We answer this question with LEAP, a deep reinforcement learning framework for amortized fixation maximization under the positional Moran process. LEAP formulates active node selection as a Markov decision problem and learns fixation-aware representations that combine graph structure with mutant invasion and stability signals. Its fixation-modulated message passing mechanism models selection pressure through learned fitness contrasts, while fixation-weighted pooling constructs a global evolutionary representation. Experiments on real-world networks show that LEAP matches the solution quality of simulation-based greedy methods with approximately five orders of magnitude faster inference and outperforms all baselines in at least two-thirds of experimental settings. These results show that deep reinforcement learning can optimize long-horizon competitive diffusion when equipped with fixation-aware mechanisms that capture evolutionary dynamics.

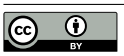
CCS Concepts

• **Human-centered computing** → **Social network analysis**; **Social networks**; • **Networks** → **Online social networks**.

Keywords

Evolutionary dynamics, positional Moran process, fixation maximization, social network analysis, deep reinforcement learning

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CIKM '26, Rome, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840796

ACM Reference Format:
Smita Ghosh and Shivvrat Arya. 2026. LEAP: Learning Evolutionary Advantage Policies for Fixation Maximization via Deep Reinforcement Learning. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799682.3840796

1 Introduction

The spread of behaviors, information, pathogens, traits, and innovations over networks is a central problem in data mining, social network analysis, and network science. Many such phenomena are modeled as stochastic diffusion or invasion processes, where entities propagate through repeated interactions constrained by network structure. Understanding and controlling these dynamics supports applications ranging from viral marketing and platform design to public health intervention and evolutionary control.

Within this landscape, *influence maximization* (IM) has emerged as a canonical optimization problem for network intervention. Given a network and a diffusion model, IM selects a small seed set whose initial activation maximizes expected influence spread [7, 20, 39, 50, 84]. This formulation has led to submodularity-based algorithms, greedy approximations, and sampling-based estimators with both theoretical guarantees and practical scalability. However, classical IM relies on an *irreversibility* assumption: once a node becomes activated, it remains active. This assumption breaks down in competitive settings, where hashtags, products, standards, pathogens, or traits can replace one another over time. In such systems, early exposure does not ensure long-run dominance, and subsequent competition can erase short-term gains.

This limitation motivates diffusion models that explicitly capture replacement and long-horizon competition. Competitive dynamics arise in population genetics, where mutations or engineered gene drives compete in natural populations [19, 33]; in epidemiology, where pathogens or vaccines spread through contact networks [49, 74, 78]; and in technological and social systems, where incompatible standards, products, or narratives diffuse under structural and behavioral constraints [48]. Across these domains, the relevant objective is not maximizing short-term exposure, but achieving long-run dominance under persistent competition.

The Moran process models competitive invasion through repeated stochastic replacement events between a mutant trait and residents [6, 28, 62]. Unlike cascade-based diffusion, it studies *fixation probability*: the probability that the mutant eventually takes over the population [61]. This perspective links network topology to long-run evolutionary outcomes, including the design of *amplifiers* that systematically increase fixation probability [36, 38, 72, 81, 94]. In spatially heterogeneous systems, selective advantage depends on where a trait originates and propagates, such as through dense

communities or strategic hubs. To model this setting, Brendborg et al. [16] introduced the *positional Moran process*, where a fitness advantage (δ) applies only to a designated node subset. This formulation shifts the intervention objective from selecting initial adopters to allocating positional advantage: given a graph G, select k nodes that maximize fixation probability.

This fixation maximization (FM) problem is difficult because the objective is non-submodular, highly stochastic, and computationally intractable to evaluate exactly. Simulation-based greedy methods require repeated objective evaluations and lack bounded approximation guarantees outside restrictive regimes [16, 34]. Topology-driven heuristics avoid this cost but ignore evolutionary dynamics, while learning-based methods for related problems such as IM do not model reversible competition, stochastic replacement, or long-horizon fixation. These limitations expose a central question: *how can we learn to allocate positional advantage when the objective is non-submodular, gradients are unavailable, dynamics are stochastic, and exact optimization is infeasible?*

We address this question with *fixation-aware policy learning*, a deep reinforcement learning (DRL) approach realized by LEAP for fixation maximization in the positional Moran process. The key benefit is amortization: during training, LEAP uses offline simulations on small synthetic graphs, where Monte Carlo fixation estimates are tractable, to learn a reusable policy for selecting advantaged nodes. At inference time, the learned policy selects nodes on unseen networks through a forward pass, avoiding repeated Monte Carlo evaluation on large target graphs. We formulate node selection as a Markov Decision Process, where states encode the partially selected advantage set and actions select additional nodes. Each selection reshapes the evolutionary landscape and changes the gains of subsequent choices, so the agent learns from simulation-based feedback to estimate long-horizon fixation gains under stochastic competition, positional advantage, and delayed rewards.

This formulation directly captures the sequential and history-dependent structure of fixation maximization: a node's value depends not only on its local topology but also on previously selected nodes and their induced evolutionary effects. Fixation maximization further introduces structural asymmetries absent from classical IM. Early suboptimal selections can suppress future spread, while distinct selected sets may achieve similar fixation probabilities through different evolutionary trajectories [16]. Fixation rewards can also exhibit high variance, complicating policy optimization. LEAP addresses these challenges through ECE, a fixation-aware graph representation; a fixation-aware aggregation module; and curriculum-driven training that stabilizes learning across increasingly complex topological and dynamical regimes.

Our contributions are summarized as follows:

- We introduce LEAP, a deep reinforcement learning framework for fixation maximization under the positional Moran process. To the best of our knowledge, LEAP is the first DRL formulation for this setting, casting active node selection as a sequential decision problem that learns transferable fixation policies from Moran-induced rewards and amortizes selection through a single forward-pass inference.
- We propose ECE, a fixation-modulated graph representation that integrates structural, evolutionary, and spectral properties with mutant invasion and stability signals, which prior work has shown to be informative for the task. We further introduce a fixation-aware aggregation module that refines these representations through fixation-modulated message passing and fixation-weighted pooling to obtain selection-pressure-aware node and graph representations for effective active node selection.
- We develop a phased training curriculum that stabilizes learning under high-variance fixation rewards by progressively increasing topological complexity.
- Extensive experiments on real-world networks show that LEAP consistently outperforms heuristic, process-aware greedy, and learning-based baselines across selection regimes and budgets, maintains low variance under stochastic fixation dynamics, achieves more than $10^5\times$ faster inference on average than the simulation-driven greedy baseline, and scales to large networks where repeated fixation-score estimation becomes computationally prohibitive.

2 Related Work

Evolutionary Dynamics on Graphs. The Moran process [73] is the canonical model of stochastic competition in finite populations, characterizing birth–death dynamics driven by relative fitness. In evolutionary graph theory, network topology critically shapes these dynamics; in particular, amplifiers systematically increase the fixation probability of advantageous mutants [1, 36, 38, 61]. Most prior work estimates fixation probabilities on static graphs via algorithmic approximations or Monte Carlo methods under uniform fitness assumptions [22, 29, 30]. These methods evaluate evolutionary outcomes rather than optimize them. The positional Moran process [16] introduces heterogeneous node fitness, yielding the *Fixation Maximization* (FM) problem: selecting an active set that maximizes eventual fixation probability. In general regimes, FM is analytically intractable due to stochastic absorption and high-variance outcomes. Although closed-form results exist in extreme selection limits, scalable optimization for arbitrary networks remains limited.

Influence Maximization. Influence Maximization (IM) selects seed nodes to maximize expected spread under stochastic diffusion models such as Independent Cascade and Linear Threshold [50]. Their submodularity enables greedy $(1 - 1/e)$ approximations and large-scale algorithms [14, 59, 91]. Recent DRL-based IM methods learn seeding policies for non-submodular or partially observable settings [60, 96], with extensions to group influence [37], rumor containment [56], competitive diffusion [27], and trust-aware propagation [55]. Despite structural similarities, FM and IM differ fundamentally: IM assumes progressive, irreversible diffusion, whereas the Moran process induces reversible stochastic replacement before absorption. Thus, FM depends on long-run stochastic fixation rather than monotone cascade spread, making classical IM guarantees inapplicable.

Learning for Graph Combinatorial Optimization. DRL has become a standard framework for graph combinatorial optimization, combining graph neural networks with RL to learn constructive or local-improvement policies for problems such as Maximum Cut, Minimum Vertex Cover, and Traveling Salesman [51, 54, 66]. These methods encode structural information via message passing and primarily optimize deterministic or low-variance objectives, with

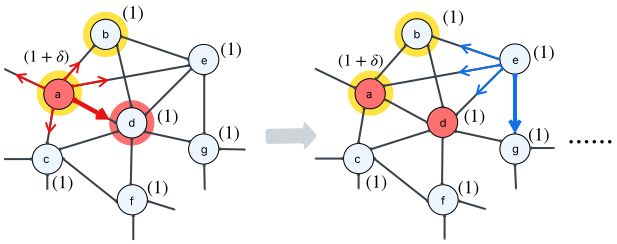


Figure 1: Positional Moran process dynamics with active nodes and mutant reproduction.

extensions to resource allocation and coverage tasks. Applying this paradigm to evolutionary dynamics is challenging: fixation rewards are highly variable due to stochastic absorption, leading to unstable gradients and slow convergence, while optimization depends on long-term equilibrium behavior. We address these challenges by coupling fixation-aware representations with simulation-driven reinforcement learning and curriculum design, enabling efficient policy learning under high-variance evolutionary feedback.

3 Preliminaries

Structured Populations and Graphs. A structured population can be modeled as a finite, connected graph $G = (V, E)$ with $|V| = n$ vertices, where each vertex represents an individual and edges encode potential reproduction or interaction opportunities. For a vertex v, let $N(v) = \{u \mid (u, v) \in E\}$ denote its neighbors and $d(v) = |N(v)|$ its degree. The graph is equivalently represented by an adjacency matrix $A \in \{0, 1\}^{n\times n}$, where $A_{uv} = 1$ if $(u, v) \in E$ and 0 otherwise. Each edge is also associated with a weight w_{uv}, representing the interaction probability between nodes u and v. The population comprises of two entities: mutants and residents, with all entities of the same type considered indistinguishable.

Active node sets and fitness. Positional fitness restricts the mutant advantage to a subset of vertices $S \subseteq V$, called the active node set [16]. Given a mutant set $X \subseteq V$, the fitness of node u is

$$f_{X,S}(u) = \begin{cases} 1 + \delta, & u \in X \cap S, \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

where $\delta \geq 0$ is the mutant fitness advantage. Fitness measures reproductive potential, with resident fitness normalized to 1 and δ capturing the relative competitive benefit of mutation.

The Positional Moran Process. The positional Moran process [16] extends the classical Moran model by allowing fitness to depend on node position. At $t = 0$, a single mutant is placed uniformly at random: $\mathbb{P}[X_0 = \{u\}] = 1/n$ for all $u \in V$. At each discrete time step, the population evolves from X_t to X_{t+1} by executing two stochastic events in succession.

(1) **Birth:** Select u with probability $\frac{f_{X_t,S}(u)}{\sum_{v\in V} f_{X_t,S}(v)}$.
(2) **Death:** Sample $v \in N(u)$ proportional to $w_{u,v}$, then replace the entity at v with a copy of that at u.

If u is a mutant and v a resident, the mutant spreads ($X_{t+1} = X_t \cup \{v\}$); if u is a resident and v a mutant, the mutant is replaced ($X_{t+1} = X_t \setminus \{v\}$); otherwise, $X_{t+1} = X_t$. The process forms a finite Markov chain that almost surely reaches a homogeneous state: $\lim_{t\to\infty} \mathbb{P}[X_t \in \{\emptyset, V\}] = 1$. Mutants fixate if $X_\tau = V$ for some $\tau \geq 0$; otherwise, they go extinct.

Example 1. *Figure 1 illustrates the positional Moran process. Nodes represent individuals in a structured population, with mutants (red) and residents (blue). Yellow rings denote active nodes where mutants receive a fitness advantage* $(1 + \delta)$*. At* $t = 0$ *(left), a mutant at node* a *is active and reproduces with probability proportional to its fitness, replacing a neighboring resident (node* d*). As the process evolves (right), both mutants and residents reproduce, allowing residents to reconvert mutants, as indicated by blue arrows. This stochastic reproduction and replacement continues until mutants either fixate or go extinct.*

The *fixation probability* for an initial mutant set $A \subseteq V$ is

$$fp(G^S, \delta, A) = \mathbb{P}[\exists \tau \geq 0 : X_\tau = V \mid X_0 = A], \tag{2}$$

and, for a uniformly random initial mutant,

$$fp(G^S, \delta) = \frac{1}{n} \sum_{u\in V} fp(G^S, \delta, \{u\}). \tag{3}$$

Because the underlying Markov chain is finite with transition probabilities rational in δ, $fp(G^S, \delta)$ is continuous and infinitely differentiable for all $\delta \geq 0$. In the neutral case ($\delta = 0$), $fp(G^S, 0) = 1/n$ for any graph G [18]. For $\delta > 0$, exact computation of $fp(G^S, \delta)$ remains open even when $S = V$. However, for undirected graphs with $S = V$, fully polynomial randomized approximation schemes (FPRAS) exist [5, 22, 29] due to polynomial absorption times, enabling efficient Monte Carlo estimation.

Fixation Maximization via Node Activation. We study Fixation Maximization (FM) under the positional Moran process and denote it by $\mathrm{FM}(G, \delta, k)$. Given a graph $G = (V, E)$, mutant fitness advantage $\delta \geq 0$, and budget k, FM seeks an active set $S \subseteq V$ with $|S| \leq k$ that maximizes the fixation probability $fp(G^S, \delta)$:

$$S^* = \arg\max_{S\subseteq V,\, |S|\leq k} fp(G^S, \delta). \tag{4}$$

Since $fp(G^S, \delta)$ is monotone in S, the constraint is tight ($|S| = k$) [16]. The decision variant asks whether some S satisfies $fp(G^S, \delta) \geq p$ for a threshold $p \in [0, 1]$. Even on simple graphs, activation patterns can yield substantially different fixation probabilities depending on δ and k [16].

Selection regimes. Brendborg et al. [16] analyzed two limiting regimes. Under *strong selection* ($\delta \to \infty$), reproduction becomes nearly deterministic and fixation probability converges to

$$fp^\infty(G^S) = \lim_{\delta\to\infty} fp(G^S, \delta),$$

yielding $\mathrm{FM}^\infty(G, k)$, which maximizes $fp^\infty(G^S)$. Under *weak selection* ($\delta \to 0^+$), fixation probability admits the first-order approximation

$$fp(G^S, \delta) \approx \tfrac{1}{n} + \delta\, fp'(G^S, 0),$$

where $fp'(G^S, 0) = \frac{d}{d\delta} fp(G^S, \delta)\big|_{\delta=0}$. Maximizing this term defines $\mathrm{FM}^0(G, k)$, which admits a closed-form solution. In contrast, intermediate selection regimes remain analytically intractable.

4 Fixation-Aware Representation Learning

Effective sequential mutant placement requires node representations that capture both graph structure and the evolutionary mechanisms that govern fixation. We introduce a fixation-aware representation learning framework that conditions policy learning on complementary structural, dynamical, and game-theoretic signals. The framework has two stages. First, we construct Evolutionary Conditioning Embeddings (ECE) that organize fixation-relevant information into interpretable feature blocks: topology, centrality, selection-limit proxies, fixation heuristics, evolutionary dynamics, game-theoretic indicators, and global structural geometry. Second, we aggregate these embeddings with a fixation-modulated message-passing architecture that produces node-level representations, a graph-level summary, and node-wise Q-values for sequential action selection. We now describe the embedding construction and the aggregation architecture used to transform these signals into fixation-aware Q-value predictions.

4.1 Evolutionary Conditioning Embeddings

Let $G = (V, E)$ denote the population graph with $|V| = n$. For each feature block m, we construct a matrix $\mathbf{F}^{(m)} \in \mathbb{R}^{n \times d_m}$, where row i contains the features associated with node i. We normalize each block separately and concatenate the resulting representations. This design treats fixation as a multi-mechanism process rather than reducing the graph to a single descriptor.

Degree-based. Degree captures local connectivity, which influences mutant propagation [6, 58, 61, 70]. For each node i, we use $\mathbf{f}_i^{\text{deg}} = [\widehat{d}_i,\ \log(1+d(i)),\ d(i)/\max_j d_j,\ \sqrt{d(i)}]$, where $d(i)$ denotes the degree of node i, and $\widehat{d}_i = \frac{d(i)-\mu_d}{\sigma_d+\varepsilon}$ is the normalized degree, with μ_d and σ_d denoting the empirical mean and standard deviation of node degrees. These transformations preserve connectivity information while distinguishing moderate hubs from extreme hubs.

Centrality. Centrality features measure how strongly a node connects and influences different regions of the graph [52, 53]. We combine three normalized measures, $\mathbf{f}_i^{\text{cent}} = [\widehat{\text{bet}}_i,\ \widehat{\text{eig}}_i,\ \widehat{\text{pr}}_i]$, where betweenness centrality captures shortest-path control [15], eigenvector centrality measures influence through well-connected neighbors [98], and PageRank reflects steady-state visitation under neutral drift [77]. Together, these features identify structurally influential positions for Fixation Maximization (FM).

Selection-limit proxies. We use neutral-drift quantities to approximate fixation-relevant node properties in the weak- and strong-selection limits. Under weak selection ($\delta \to 0$), we estimate each node's reproductive value [65] as $\mathbf{f}_i^{\text{weak}} = [\widehat{\text{er}}_i,\ \widehat{\text{ct}}_i,\ \widehat{\text{abs}}_i]$, where er_i denotes effective resistance, capturing connectivity-driven genetic drift and spread potential [31]; ct_i denotes commute time, capturing the expected neutral round trip [21]; and abs_i denotes the absorption probability for walks initialized elsewhere [29]. These proxies summarize reachability and persistence under weak selection [16]. Under strong selection ($\delta \to \infty$), we estimate the probability that a neutral mutant reaches each node before extinction [61, 80, 83]. We simulate T neutral random walks with per-step extinction probability ε_{ext} and maximum length $L_{\max}$, and compute $\mathbf{f}_i^{\text{strong}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{i \in \text{Vis}(\omega^{(t)})\}$, where $\text{Vis}(\omega^{(t)})$ contains vertices visited before termination or truncation. Large $\mathbf{f}_i^{\text{strong}}$ indicates frequent reachability under neutral drift, serving as an early-reachability proxy for strong-selection fixation [4, 61, 80].

Fixation-specific heuristics. This block aggregates interpretable fixation-oriented features, $\mathbf{f}_i^{\text{fix}} = [\widehat{\text{amp}}_i,\ \widehat{\text{bar}}_i,\ \widehat{\text{prop}}_i,\ \widehat{\text{drift}}_i]$ [61, 67]. Here, amp_i measures mutant amplification potential through degree-weighted neighbor strength and local expansion [9, 61]; bar_i estimates fixation barriers from distances and clustering that can trap mutants [42, 46, 57]; prop_i estimates selective-advantage propagation by combining influence with core-periphery position [13, 52]; and drift_i estimates how local structure dissipates neutral lineages [17, 46, 57]. These features identify nodes that initiate, sustain, or obstruct mutant spread [61, 67].

Evolutionary dynamics. This block summarizes process-level evolutionary behavior through $\mathbf{f}_i^{\text{evo}} = [\widehat{\text{bd}}_i,\ \widehat{\text{diff}}_i,\ \widehat{\text{inv}}_i,\ \widehat{\text{ftime}}_i]$. The feature bd_i combines out-degree with neighbor quality to quantify reproductive advantage under birth–death dynamics [61]; diff_i captures short-step diffusion of fitness effects through local random walks [2]; inv_i integrates local dominance with network closeness to estimate mutant invasion potential [71]; and ftime_i uses eccentricity and local mixing-time surrogates to approximate time to global fixation [44, 57, 93]. These signals characterize both the speed and robustness of evolutionary spread.

Game-theoretic indicators. This block encodes strategic stability and competitive resilience under frequency-dependent fitness, treating fixation as an evolutionary game on the graph [3, 90]: $\mathbf{f}_i^{\text{game}} = [\widehat{\text{nash}}_i,\ \widehat{\text{ess}}_i,\ \widehat{\text{flow}}_i]$. Here, nash_i quantifies the stability of placing a mutant at node i under neighbors' strategic responses, combining local dominance with global connectivity [3, 92]. The feature ess_i estimates resistance to invasion by alternative strategies and rewards nodes that remain advantageous after selection [95]. The feature flow_i measures how effectively selective advantage propagates through one- and two-hop neighborhoods, approximating payoff diffusion and resource accessibility [75, 76].

Structural embeddings. To capture global geometry beyond handcrafted statistics, we compute spectral features from the k smallest nontrivial eigenvectors of the normalized Laplacian [10, 25, 87] and multi-step random-walk signatures following Node2Vec [40, 82]. We concatenate these features and reduce them by PCA [47] to d_{struc} dimensions: $\mathbf{S} \in \mathbb{R}^{n \times p} \xrightarrow{\text{PCA}} \tilde{\mathbf{S}} \in \mathbb{R}^{n \times d_{\text{struc}}}$. Spectral embeddings encode global connectivity and community structure [26, 87], whereas random-walk features capture higher-order reachability and diffusion patterns [40, 82]. These representations complement scalar graph statistics and support generalization across graph topologies.

Normalization and Final Embedding. We normalize each scalar feature x as $\hat{x} = \frac{x-\mu}{\sigma+\varepsilon}$, where μ and σ are the empirical mean and standard deviation computed within each graph and feature block, and $\varepsilon > 0$ ensures numerical stability. We then concatenate all normalized blocks $\mathbf{Z} = \left[\mathbf{F}^{\text{deg}}, \mathbf{F}^{\text{cent}}, \mathbf{F}^{\text{weak}}, \mathbf{F}^{\text{strong}}, \mathbf{F}^{\text{fix}}, \mathbf{F}^{\text{evo}}, \mathbf{F}^{\text{game}}, \tilde{\mathbf{S}}\right] \in \mathbb{R}^{n \times d}$.

The resulting evolutionary conditioning embedding unifies local connectivity, global influence, selection-limit behavior, fixation heuristics, evolutionary dynamics, game-theoretic stability, and spectral/random-walk geometry for fixation-aware policy learning.

4.2 Fixation-Aware Embedding Aggregation

Given the evolutionary conditioning embeddings $\mathbf{Z}$, we use Learning Evolutionary Advantage Policies (LEAP) to produce evolution-informed

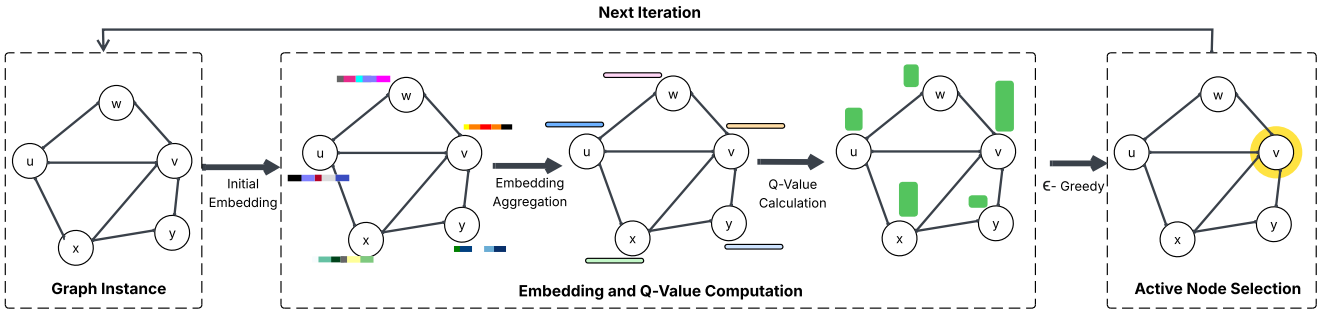


Figure 2: Overview of Learning Evolutionary Advantage Policies (LEAP): node embeddings are aggregated to predict Q-values, and the top-scoring node is selected iteratively under a k-budget.

representations for action selection. As shown in Figure 2, the model maps $\mathbf{Z}$ to node-level embeddings $\mathbf{H} \in \mathbb{R}^{n\times h}$, a graph-level summary $\mathbf{g} \in \mathbb{R}^h$, and node-wise Q-values q_i. The architecture integrates evolutionary priors through fixation-modulated message passing and fixation-weighted pooling.

4.2.1 Input Projection. We first project $\mathbf{Z}$ to a width h latent space:

$$\mathbf{H}^{(0)} = \phi\big(\mathbf{Z}\mathbf{W}_{\text{in}} + \mathbf{1}\mathbf{b}_{\text{in}}^\top\big), \tag{5}$$

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d\times h}$ and $\mathbf{b}_{\text{in}} \in \mathbb{R}^h$ are trainable parameters, and $\phi(x) = \max(0, x)$.

4.2.2 Fixation-Modulated Message Passing. We stack L message-passing layers Φ_ℓ, each coupling node features with a learned fitness landscape and selection-gated propagation controlled by $\alpha > 0$. At layer ℓ, node states are $\mathbf{H}^{(\ell)} = [\mathbf{h}_1^{(\ell)}; \ldots; \mathbf{h}_n^{(\ell)}]$. Given weighted edges $\mathcal{E} = \{(u, v, w_{uv})\}$,

$$\big(\mathbf{H}^{(\ell+1)}, \boldsymbol{\pi}^{(\ell)}\big) = \Phi_\ell\big(\mathbf{H}^{(\ell)}, \mathcal{E}; \alpha\big), \tag{6}$$

where $\boldsymbol{\pi}^{(\ell)} \in [0, 1]^n$ are auxiliary fixation scores.

Each layer contains three components:

Selection potential and birth transform. Each node i receives two learned quantities: a scalar fitness potential defining a latent selection order and a birth-transformed representation for node updates:

$$f_i^{(\ell)} = \sigma\big(\mathbf{v}_f^\top \mathbf{h}_i^{(\ell)} + c_f\big), \qquad \mathbf{b}_i^{(\ell)} = \phi\big(\mathbf{W}_b\, \mathbf{h}_i^{(\ell)}\big). \tag{7}$$

Selection-pressure propagation. For each edge $(j \to i)$, the message depends on source and target features, the edge weight, a learned gate, and the learned fitness contrast between nodes:

$$\begin{aligned} \mathbf{s}_{j\to i}^{(\ell)} &= \mathbf{W}_s[\mathbf{h}_j^{(\ell)} \| \mathbf{h}_i^{(\ell)}], \quad g_{j\to i}^{(\ell)} = \sigma\big(\mathbf{v}_g^\top \mathbf{s}_{j\to i}^{(\ell)} + c_g\big), \\ \mathbf{a}_i^{(\ell)} &= \sum_{j\in\mathcal{N}(i)} w_{ji}\, g_{j\to i}^{(\ell)}\, \sigma\big(\alpha(f_j^{(\ell)} - f_i^{(\ell)})\big)\, \mathbf{s}_{j\to i}^{(\ell)}. \end{aligned} \tag{8}$$

The term $\sigma(\alpha(f_j^{(\ell)} - f_i^{(\ell)}))$ increases the contribution of neighbors with higher latent fitness potential and suppresses messages from lower-potential neighbors. This mechanism aligns message passing with selection pressure rather than purely structural adjacency.

Amplification and update. Let $\rho_i = d_i/\max_k d_k$. Structural amplification combines self, neighbor, and density channels, followed by dropout for regularization.

$$\begin{aligned} \mathbf{u}_i^{(\ell)} &= \phi\big(\mathbf{W}_a \mathbf{h}_i^{(\ell)}\big), \qquad \tilde{\mathbf{h}}_i^{(\ell)} = \gamma_0 \mathbf{u}_i^{(\ell)} + \gamma_1 \mathbf{a}_i^{(\ell)} + \gamma_2 \rho_i \mathbf{u}_i^{(\ell)}, \\ \mathbf{h}_i^{(\ell+1)} &= \mathrm{LN}\big(\mathbf{b}_i^{(\ell)} + \tilde{\mathbf{h}}_i^{(\ell)}\big). \end{aligned} \tag{9}$$

Each layer also outputs fixation relevance scores

$$\pi_i^{(\ell)} = \sigma\big(\mathbf{v}_p^\top \mathbf{h}_i^{(\ell+1)} + c_p\big), \qquad \bar{\pi}_i = \frac{1}{L}\sum_{\ell=0}^{L-1} \pi_i^{(\ell)}. \tag{10}$$

We use $\sigma(x) = 1/(1 + e^{-x})$ and LN for layer normalization over the h features, with concatenation denoted by $[\cdot\|\cdot]$. The trainable parameters and their dimensions are

$$\begin{aligned} &\mathbf{v}_f, \mathbf{v}_g, \mathbf{v}_p \in \mathbb{R}^h,\ c_f, c_g, c_p \in \mathbb{R},\ \mathbf{W}_b, \mathbf{W}_a, \mathbf{W}_t \in \mathbb{R}^{h\times h}, \\ &\mathbf{W}_s \in \mathbb{R}^{h\times 2h},\ \gamma_0, \gamma_1, \gamma_2 \in \mathbb{R}. \end{aligned}$$

4.2.3 Graph-Level Fixation Pooling. Given final node states $\mathbf{H} = \mathbf{H}^{(L)}$ and averaged scores $\bar{\boldsymbol{\pi}}$, we compute a fixation-aligned graph representation:

$$\mathbf{g} = \frac{\sum_i \bar{\pi}_i\, \mathbf{W}_t \mathbf{h}_i}{\sum_i \bar{\pi}_i + \varepsilon}, \tag{11}$$

with trainable $\mathbf{W}_t \in \mathbb{R}^{h\times h}$. This pooling assigns higher weight to nodes predicted to be fixation-relevant, producing a graph-level summary tailored to fixation-aware action selection.

Summary. The proposed fixation-aware representation learning pipeline combines two complementary stages. First, the ECE encode node-level structural, dynamical, game-theoretic, and fixation-specific signals into a representation $\mathbf{Z}$. Second, LEAP transforms $\mathbf{Z}$ through fixation-modulated message passing and fixation-weighted pooling, embedding evolutionary priors at the edge, node, and graph levels: (i) edge-level selection pressure via fitness contrast (Eq. 8), (ii) node-level amplification via degree-normalized updates (Eq. 9), and (iii) graph-level fixation-weighted pooling (Eq. 11). Together, these components produce compact fixation-aware representations for sequential node selection under a budget.

5 Learning Node Selection via Q-Learning

Fixation maximization poses three coupled challenges: the objective is non-submodular, node-selection decisions are combinatorial and therefore non-differentiable, and exact optimization becomes

intractable on large graphs. We address these challenges by formulating node selection as a finite-horizon Markov decision process (MDP), where an agent sequentially selects nodes under a cardinality budget. The agent receives fixation-aware marginal rewards and learns an action-value function Q_θ, parameterized by LEAP (Section 4), over evolutionary graph embeddings. This formulation converts a difficult combinatorial optimization problem into a sequential decision-making task that can be trained using value-based reinforcement learning.

5.1 MDP Formulation

Let $G = (V, E)$ be the input graph, $\delta \geq 0$ the mutant fitness advantage, and k the node-activation budget. The goal is to find an active set $S \subseteq V$, with $|S| \leq k$, that maximizes $fp(G^S, \delta)$. We model this optimization problem as a finite-horizon MDP $(\Omega, \mathcal{A}, \mathcal{T}, r, \gamma)$, with horizon $M = k$ and discount factor $\gamma \in [0, 1]$.

State. The state space is $\Omega = \{0, 1\}^{|V|}$. At decision step m, the state $\mathbf{s}_m \in \Omega$ encodes the current active set $S_m \subseteq V$:

$$\mathbf{s}_{m,i} = 1 \Leftrightarrow i \in S_m. \tag{12}$$

Each episode starts from $S_0 = \emptyset$, equivalently $\mathbf{s}_0 = \mathbf{0}$, and terminates when $|S_m| = k$.

Actions. At decision step m, the feasible actions are the inactive nodes:

$$\mathcal{A}(\mathbf{s}_m) = \{i \in V : \mathbf{s}_{m,i} = 0\}. \tag{13}$$

Selecting action $a_m \in \mathcal{A}(\mathbf{s}_m)$ activates node a_m.

Transition. The transition is deterministic:

$$S_{m+1} = S_m \cup \{a_m\}, \qquad \mathbf{s}_{m+1} = \mathbf{s}_m \vee \mathbf{e}_{a_m}, \tag{14}$$

where $\mathbf{e}_{a_m} \in \{0, 1\}^{|V|}$ is the one-hot vector for node a_m, and $\vee$ denotes elementwise logical OR.

Reward. The reward measures the marginal improvement in fixation probability from activating one additional node. For a partial active set S_m, the corresponding positional Moran process is defined on G^{S_m}, where active nodes receive the fitness advantage specified in Sec. 3. We estimate the fixation probability $fp(G^{S_m}, \delta)$ (Eq. (3)) using N_{trials} Monte Carlo trials:

$$\widehat{fp}(G^{S_m}, \delta) = \frac{1}{N_{\text{trials}}} \sum_{\ell=1}^{N_{\text{trials}}} \mathbb{I}[\text{fixation occurs in trial } \ell]. \tag{15}$$

Each trial samples the initial mutant uniformly at random, consistent with the definition of $fp(G^{S_m}, \delta)$. The immediate reward is the estimated marginal gain:

$$r_m = \widehat{fp}(G^{S_{m+1}}, \delta) - \widehat{fp}(G^{S_m}, \delta). \tag{16}$$

Objective. The policy π maximizes the expected discounted return:

$$\max_\pi \ \mathbb{E}_\pi \left[\sum_{m=0}^{M-1} \gamma^m r(\mathbf{s}_m, a_m) \right], \quad a_m \sim \pi(\cdot \mid \mathbf{s}_m), \quad \mathbf{s}_{m+1} = \mathcal{T}(\mathbf{s}_m, a_m).$$

When $\gamma = 1$ and rewards are unclipped, the return telescopes to

$$\widehat{fp}(G^{S_M}, \delta) - \widehat{fp}(G^{S_0}, \delta).$$

Since $S_0 = \emptyset$, maximizing this return is equivalent to maximizing the final estimated fixation probability $\widehat{fp}(G^{S_M}, \delta)$. For $\gamma < 1$ or clipped rewards, the training objective becomes a shaped approximation of the original fixation maximization objective.

5.2 State-Conditioned Action-Value Estimation

LEAP estimates $Q_\theta(\mathbf{s}_m, a; G)$ for each feasible node action. Given the graph representation $\mathbf{Z}$ and activation state $\mathbf{s}_m$, the model produces node embeddings $\mathbf{h}_i^{(m)}$ and a pooled graph summary $\mathbf{g}^{(m)}$. For each candidate node i, the Q-head computes

$$q_i = Q_\theta(\mathbf{s}_m, i; G) = \text{MLP}\left([\mathbf{h}_i^{(m)} \| \mathbf{g}^{(m)}] \right), \tag{17}$$

where the MLP is a two-layer network with ReLU activation and dropout. Before selecting an action, we mask active nodes and choose

$$a_m^* = \arg\max_{i \in V \setminus S_m} Q_\theta(\mathbf{s}_m, i; G). \tag{18}$$

This masking prevents repeated activations and enforces the cardinality constraint over the active set.

5.3 Training with n-Step Double Q-Learning

We train Q_θ using n-step Double Deep Q-learning (DDQN) [43, 69]. The method maintains an online network Q_θ and a target network Q_{θ^-}. For a sampled transition sequence beginning at $(\mathbf{s}_m, a_m)$, with rewards $r_m, \ldots, r_{m+n-1}$ and resulting state $\mathbf{s}_{m+n}$, the target is

$$y_m = \sum_{\ell=0}^{n-1} \gamma^\ell r_{m+\ell} + \gamma^n Q_{\theta^-} \left(\mathbf{s}_{m+n}, \arg\max_{a' \in \mathcal{A}(\mathbf{s}_{m+n})} Q_\theta(\mathbf{s}_{m+n}, a'; G); G \right).$$

The feasible-action mask in the inner maximization prevents the target from bootstrapping on already active nodes. We minimize the mean-squared temporal-difference error over a minibatch of size B:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \left(Q_\theta(\mathbf{s}_{m,i}, a_{m,i}; G) - y_{m,i} \right)^2. \tag{19}$$

We update the target network using an exponential moving average, $\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^-$, and use an ϵ-greedy policy for exploration.

5.4 Curriculum for Fixation-Aware Q-Learning

Directly training on fixation maximization can produce high-variance TD targets because fixation rewards are estimated from stochastic simulations and depend strongly on graph topology. We address this challenge by using curriculum learning [11] to structure the training distribution over graphs while optimizing a single action-value function across all phases.

Let $\{P_c\}_{c=1}^3$ denote phase-specific graph distributions over $\mathcal{G}$, ordered by increasing structural complexity. The curriculum exposes the agent first to fixation patterns driven by local connectivity, then to mesoscopic community structure, and finally to global motifs with long-range dependencies and bottleneck effects. Phase 1 samples Erdős–Rényi graphs [32] and Barabási–Albert preferential attachment networks [8], where local neighborhoods, edge density, and degree heterogeneity largely determine fixation behavior, enabling the agent to learn basic activation preferences such as selecting nodes that influence many reproduction events. Phase 2 introduces Stochastic Block Models [45], whose modular structure requires reasoning about fixation within and across communities, including inter-community bridges, community-size imbalance, and interactions between local advantage and global spread. Phase 3 uses motif-rich topologies, including barbell, star, and Watts–Strogatz small-world graphs [97], which contain hubs,

shortcuts, peripheral leaves, and structural bottlenecks where fixation can depend on non-local propagation paths rather than degree alone. Thus, the curriculum progressively increases credit-assignment difficulty while training a single fixation-aware value function.

At curriculum phase c, we sample training graphs from the cumulative mixture

$$P_{1:c} = \sum_{j=1}^{c} \lambda_j P_j, \qquad \sum_{j=1}^{c} \lambda_j = 1, \qquad \lambda_c > \lambda_{c-1} > \cdots > \lambda_1. \quad (20)$$

The mixture weights are scheduled linearly during each phase transition. Training proceeds sequentially with warm-start initialization: $\theta^{(c)} \leftarrow \theta^{(c-1)}$ for $c > 1$, where $\theta^{(c)}$ denotes the model parameters at the end of curriculum phase c. This strategy preserves exposure to earlier graph families while gradually increasing the frequency of structurally complex topologies.

5.5 Test-Time Policy Execution

At test time, the trained LEAP greedily constructs an active set of size k on a given graph $G = (V, E)$. Starting from $S_0 = \emptyset$, equivalently $\mathbf{s}_0 = \mathbf{0}$, the model computes node-wise Q-values at each decision step, masks active nodes, and selects

$$a_m^* = \arg\max_{i \in V \setminus S_m} Q_\theta(\mathbf{s}_m, i; G). \quad (21)$$

It then updates the active set (S_m) and state vector ($\mathbf{s}_m$) and repeats this procedure until $|S_m| = k$ or no feasible action remains. Unlike greedy active node selection, test-time execution avoids evaluating exact marginal gains for every candidate at each step. Instead, the learned Q-function amortizes this computation by directly estimating candidate utilities from the current leader-selection state.

6 Experiments

We evaluate Learning Evolutionary Advantage Policies (LEAP) for fixation maximization under the positional Moran process. Our experiments address three questions: (i) whether LEAP improves fixation probability relative to structural, process-aware, and learning-based baselines; (ii) whether it amortizes expensive fixation-probability estimation into efficient test-time node selection; and (iii) which architectural components and feature groups contribute most to performance.

6.1 Experimental Setup

Training Graphs. We train LEAP using the three-phase curriculum in Section 5.4, which progresses from simple random graphs to structured and motif-rich topologies. The curriculum contains 270 synthetic graphs: 80 in Phase 1, 40 in Phase 2, and 150 in Phase 3. For each phase, we uniformly subsample $N_t = 50$ graphs and train sequentially, warm-starting each phase from the preceding one.

Evaluation graphs. We evaluate all methods on 20 real-world networks [85], which we refer to as the quality-evaluation benchmark. These datasets span diverse domains and structural regimes, including neural, biological, ecological, organizational, communication, web, and transportation networks: mouse-visual-cortex-1, firm-hi-tech, karate, stmarks, dolphins, everglades, sandi-auths, ENZYMES8, ENZYMES123, bn-macaque-rhesus-brain-2, ENZYMES118, mangwet, ENZYMES297, ENZYMES295, ENZYMES296, email-Enron-only, bn-mouse-visual-cortex-2, webkb-wisc, DD244, and USAir97. They range from 29 to 332 nodes and from 44 to 2,125 edges. This scale permits reliable fixation-score estimation through repeated positional Moran simulations while preserving substantial variation in graph size and topology.

We evaluate scalability on 8 larger real-world networks used only for runtime analysis: Anybeat [35], Gplus [68], Epinions [41], EmailEU [85], GemsecRO [86], Brightkite [24], DBLP2010 [85], and Delicious [99]. These graphs contain 12,645 to 536,108 nodes and 39,182 to 1,365,961 edges. We exclude them from fixation-quality comparisons because estimating fixation probability requires repeated positional Moran simulations until absorption, which is computationally prohibitive at this scale [16]. This separation isolates solution quality on graphs where fixation can be measured from deployment cost on graphs where direct simulation-based evaluation is impractical.

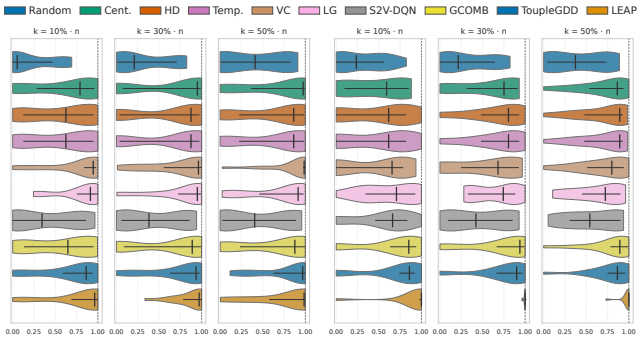
Heuristic and process-aware baselines. We compare against six heuristic and process-aware baselines from Brendborg et al. [16]: Random, High Degree (HD), Centrality (Cent.), Temperature (Temp.), Vertex Cover (VC), and Lazy Greedy (LG)[1]. These baselines cover static graph heuristics and simulation-driven optimization under the positional Moran process. We use the implementation details, hyperparameters, and simulation budgets of Brendborg et al. [16].

Learning-based baselines. We also compare against three learning-based node-selection methods originally developed for influence maximization: S2V-DQN [51], GCOMB [66], and ToupleGDD [23]. Because these methods target irreversible diffusion rather than fixation dynamics, we adapt them to fixation maximization by retaining objective-independent components, including their original state features and architectures, and replacing task-specific rewards and feasible-action constraints as described in Section 5.1.

Training configuration. We train LEAP with n-step Double Deep Q-Learning [43, 69], experience replay, and EMA target updates with $\tau = 0.01$. The agent uses $L = 3$ fixation-aware message-passing layers, hidden width $h = 128$, ReLU activations, and dropout $p = 0.1$. We use $n = 5$-step returns, AdamW with learning rate 10^{-4} and weight decay 10^{-5} [64], gradient clipping at $\|\nabla_\theta\|_2 \le 3$, and mini-batches of $B = 128$ transitions sampled from a replay buffer of size 10^5. Exploration follows a linearly decaying ϵ-greedy schedule from 0.9 to 0.05 [88]. We select hyperparameters using held-out synthetic graphs from the curriculum and reserve real-world graphs exclusively for final evaluation. We train for $E_t = 500$ epochs in Phase 1 and increase the budget by 250 epochs in each subsequent phase. We train one LEAP instance per evolutionary regime and reuse it across all activation budgets. All experiments use PyTorch [79] on a workstation with an NVIDIA A40 GPU and an Intel(R) Xeon(R) Silver 4314 CPU.

Evaluation Protocol. We estimate fixation probabilities by Monte Carlo simulation of the positional Moran process defined in Section 3. We evaluate the strong-selection objective $\mathrm{FM}^\infty(G, k)$ and

[1] We exclude weak selection because Brendborg et al. [16] showed that, as $\delta \to 0$, fixation maximization admits a closed-form approximation computable in $O(n^{2\omega})$ time, where ω is the matrix multiplication exponent.



(a) Performance for FM^{∞} **(b) Performance for FM^{δ}**

Figure 3: Average normalized fixation probability for each method.

an intermediate-selection setting $\mathsf{FM}^{\delta}(G,k)$ with $\delta = 0.33$, following prior work [12, 16, 17, 63, 89]. For each graph, budget $k \in \{10\%, 30\%, 50\%\}$, regime, and method, we run 10,000 independent trials and report the empirical fixation probability, i.e., the fraction of trials ending in fixation.

Metrics. For solution quality, we first estimate the fixation probability in Eq. (3) over 120 configurations, corresponding to 20 graphs, two evolutionary regimes, and three activation budgets. Based on these estimates, we report two quality metrics: (i) **Normalized average fixation probability** measures each method's fixation probability normalized by the maximum fixation probability achieved on the same graph-budget pair; and (ii) **Win rate** measures the pairwise fraction of configurations in which one method achieves a higher fixation probability than another, assigning 0.5 points for ties. We also report wall-clock node-selection time on the 20 quality-evaluation graphs and the 8 large runtime graphs. For learning-based methods, we separately report offline training time to distinguish one-time training cost from amortized inference cost.

6.2 Fixation Quality

We first compare LEAP with structural, process-aware, and learning-based baselines on the 20 quality-evaluation graphs. Because raw fixation probabilities vary across topologies, regimes, and budgets, we follow Brendborg et al. [16] and normalize each score by the best fixation probability achieved on the same configuration.

Figures 3a and 3b summarize normalized fixation probabilities under FM^{∞} and FM^{δ}, respectively. Figure 4 reports budget-dependent behavior on representative networks, with FM^{∞} in the top row and FM^{δ} in the bottom row.

Q1. *How do baselines and* LEAP *perform under strong selection?* Under strong selection (Fig. 3a), fixation dynamics depend heavily on graph connectivity and high-degree regions. The process-aware LG baseline performs well, consistent with its constant-factor approximation guarantee in this regime [16]. Several structural heuristics achieve competitive scores on some graphs, but their wider lower tails indicate topology-sensitive failures. The adapted learning-based baselines usually improve over simple heuristics in several cases, but they do not consistently dominate structural or process-aware methods.

LEAP achieves consistently high normalized fixation probabilities across all activation budgets and remains close to the best method even when it does not rank first. Its thinner lower tails indicate fewer severe failures, including in the low-budget regime where selecting a small set of nodes is most constrained. Across $k \in \{10\%, 30\%, 50\%\}$, LEAP matches or surpasses LG in solution quality while avoiding repeated Monte Carlo simulations at inference time. The per-graph results in Figure 4 show the same trend: LEAP remains among the strongest methods across representative networks, selection regimes, and activation budgets.

Q2. *How do baselines and* LEAP *perform under intermediate selection?* Under intermediate selection, fixation outcomes become more stochastic, and strong-selection approximation guarantees no longer apply directly. Structural heuristics such as HD, Cent., Temp., and VC degrade more substantially, suggesting that static connectivity-based signals are insufficient when stochastic replacement dynamics become more prominent. LG also shows higher variance across graphs and budgets, indicating sensitivity to selection intensity. The adapted learning-based baselines exhibit mixed performance. S2V-DQN generally attains lower normalized fixation probabilities with broad dispersion. GCOMB and ToupleGDD remain competitive in aggregate, but their violin plots show broader lower tails, especially at smaller budgets.

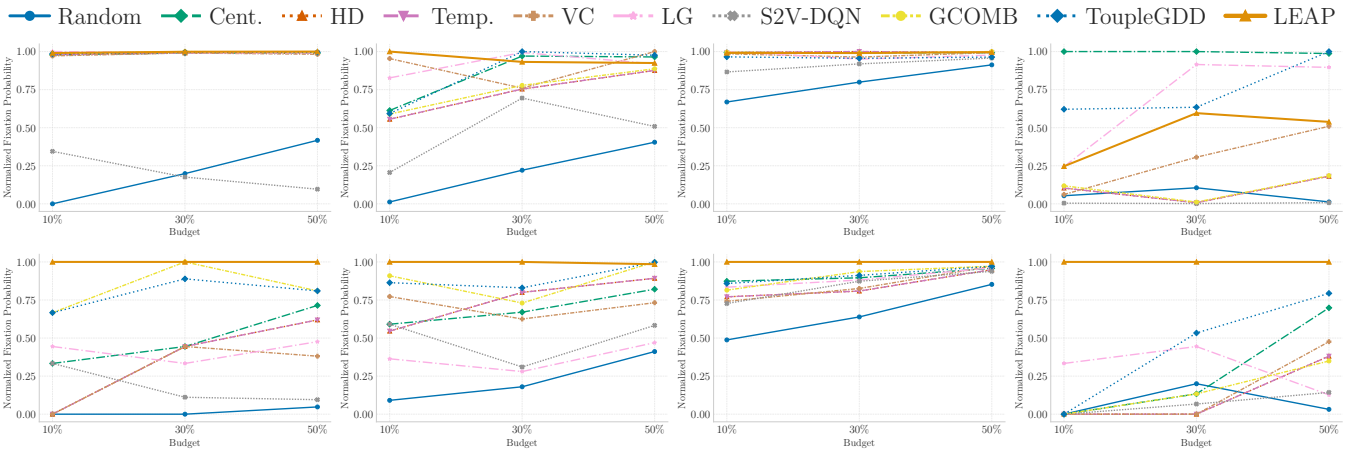
In contrast, LEAP shows the same stability observed under strong selection, maintaining high normalized fixation probabilities with lower variance across budgets. This consistency is notable because intermediate selection introduces stronger stochasticity, making fixation outcomes less directly determined by static graph structure. Together, the aggregate and per-graph results indicate that fixation-aware representations better capture stochastic replacement dynamics than generic structural features or architectures designed for irreversible influence maximization.

Pairwise win rates. Figure 5 reports pairwise win rates over the 120 quality-evaluation configurations. Each cell for an ordered method pair (i, j) gives the score of method i against method j, assigning 1 point for a higher fixation probability and 0.5 for a tie. Symmetric entries therefore sum to 120. This metric tests whether improvements are consistent across graph families, activation budgets, and selection regimes rather than concentrated on a small subset of favorable cases.

LEAP obtains the highest cumulative pairwise score. It scores above 90 out of 120 against most structural heuristics, including HD, Cent., and VC. It also outperforms the adapted learning-based baselines, scoring 106, 90, and 84 against S2V-DQN, GCOMB, and ToupleGDD, respectively. Against LG, LEAP scores 80 out of 120, indicating higher fixation probability on most configurations while avoiding repeated test-time simulations. These results support the main empirical claim that fixation-aware learning provides a favorable quality-efficiency tradeoff across heterogeneous graphs, budgets, and selection regimes.

6.3 Runtime and Amortized Inference

We evaluate runtime on the same 20 real-world networks used for fixation-quality comparisons and on 8 additional larger networks



(a) bn-macaque (b) ENZYMES (c) eco-mangwet (d) enron

Figure 4: Per-graph normalized fixation probability versus budget. Top: $\mathtt{FM}^{\infty}$, Bottom: $\mathtt{FM}^{\delta}$.

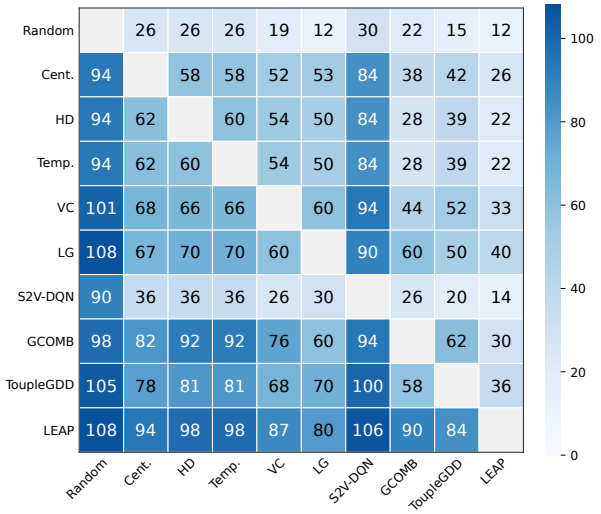
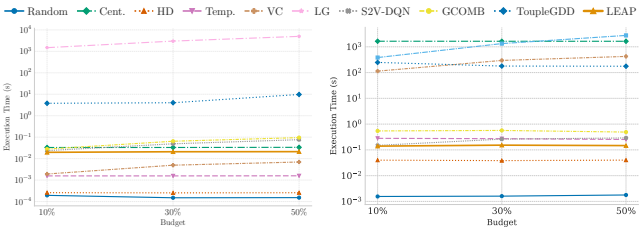


Figure 5: Pairwise win rates of baselines versus LEAP. Each cell shows the row method's total score against the column; darker blue indicates better performance.



(a) Small Networks (b) Large Networks

Figure 6: Average runtime for each method.

where fixation-score evaluation is computationally prohibitive. We report average node-selection time per run in seconds, which measures deployment cost after any required training. On the small networks, we include LG; on the large networks, LG does not complete and is therefore excluded.

On the small graphs (Figure 6a), lightweight heuristics are fastest: Random and HD average 0.0002 s and 0.0005 s per run, respectively. Temp., VC, and Cent. average 0.0032 s, 0.0356 s, and 0.2706 s, respectively. The learning-based baselines also infer efficiently: S2V-DQN and GCOMB average 0.0481 s and 0.0665 s, while ToupleGDD averages 5.0861 s. In contrast, LG averages approximately 2500 s because it repeatedly simulates the Moran process to estimate marginal gains. LEAP averages 0.0226 s, making it faster than all learning-based baselines in this setting and several orders of magnitude faster than LG.

On the large graphs (Figure 6b), Random, HD, and Temp. remain fast, averaging 0.002 s, 0.039 s, and 0.268 s, respectively. More expensive graph-analytic heuristics scale poorly: VC averages 289.9 s, and Cent. averages 1641.4 s. Among learning-based baselines, S2V-DQN and GCOMB average 0.234 s and 0.534 s, respectively, whereas ToupleGDD increases to 200.9 s. LEAP averages 0.146 s, outperforming all learning-based baselines in runtime and remaining orders of magnitude faster than VC, Cent., and ToupleGDD.

For all learning-based methods, training is a one-time offline cost amortized at inference time. LEAP requires 8,229 s per regime, compared with 2,030 s for GCOMB, 3,888 s for S2V-DQN, and 3,906 s for ToupleGDD. This higher training cost reflects the fixation-aware embedding and aggregation modules and curriculum learning. Once trained, all learning-based methods select nodes without repeated fixation-probability estimation. However, only LEAP combines fixation-specific architecture and embeddings, yielding the strongest observed quality-efficiency tradeoff for the positional Moran process.

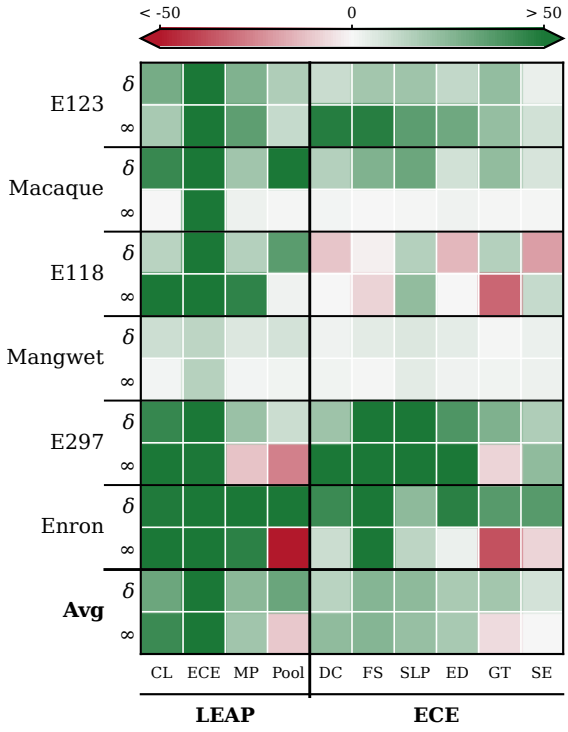


Figure 7: Ablations. Cells show relative gaps: green denotes degradation after removal, red improvement. Rows list datasets under FM^{δ} and FM^{∞}, plus averages. Columns group LEAP- and ECE-level ablations.

6.4 Component and Feature Ablations

Figure 7 summarizes component and feature ablations. Each cell reports Gap% = 100(LEAP − abl.)/LEAP, where positive values indicate degradation after removing a component. The left block ablates curriculum learning (CL; Section 5.4), evolutionary conditioning embeddings (ECE; Section 4.1), fixation-modulated message passing (MP; Section 4.2.2), and graph-level fixation pooling (Pool; Section 4.2.3). The right block ablates ECE feature groups: degree and centrality features (DC), fixation-specific heuristics (FS), selection-limit proxies (SLP), evolutionary dynamics features (ED), game-theoretic indicators (GT), and structural embeddings (SE). These ablations test whether the gains arise from fixation-aware design choices rather than a single module or generic graph representation.

At the model level, removing ECE causes the largest and most consistent degradation across datasets and regimes, identifying ECE as the most important component among those tested. Curriculum learning also contributes substantially, especially under FM^{∞}. Fixation-modulated message passing provides a moderate but stable gain. Pooling is more regime-dependent: it improves performance under FM^{δ} but yields negative gaps on a small number of datasets under FM^{∞}.

The feature-group ablations show that fixation-specific signals contribute more reliably than generic structural embeddings. Under FM^{δ}, fixation-specific heuristics (FS) and selection-limit proxies (SLP) have the strongest average effects. Under FM^{∞}, degree and centrality features (DC), fixation-specific heuristics (FS), selection-limit proxies (SLP), and evolutionary dynamics features (ED) remain broadly useful. In contrast, structural embeddings (SE) have smaller average contributions and occasionally yield negative gaps. This pattern suggests that generic structural representations, such as Node2Vec-style embeddings, are less reliable for fixation maximization than conditioning signals tailored to the Moran process.

Summary. The results suggest that transfer from synthetic training graphs to real-world networks improves when the policy encodes fixation dynamics rather than relying only on static graph structure. Structural heuristics identify locally important nodes but do not model mutant survival, stochastic replacement, or long-horizon fixation probability, which makes them sensitive to graph topology and selection regime. Simulation-based methods such as LG directly estimate fixation gains but incur high test-time cost and remain myopic. Learning-based IM baselines amortize node selection, but their architectures target progressive irreversible diffusion rather than competitive and reversible Moran dynamics. This mismatch is consistent with their higher variance and weaker pairwise win rates, especially under FM^{δ}. In contrast, LEAP combines a fixation-aware value function with dynamics-specific embeddings, message passing, pooling, and curriculum training. The ablations indicate that these components, especially ECE, contribute to its accuracy, while the runtime results show that the learned policy preserves low inference cost after offline training.

7 Conclusion and Future Work

This paper introduced Learning Evolutionary Advantage Policies (LEAP), a deep reinforcement learning framework for fixation maximization under the *positional Moran process*. To our knowledge, LEAP is the first method to formulate fixation maximization as sequential node activation and learn fixation-aware policies for this setting. It integrates fixation-aware graph embeddings, fixation-modulated message passing, fixation-weighted pooling, and curriculum learning to model how network topology shapes stochastic fixation dynamics. Experiments on real-world networks show that LEAP transfers reliably from synthetic training graphs across selection regimes and activation budgets. It improves fixation probability over structural, process-aware, and learning-based baselines while achieving orders-of-magnitude faster test-time inference than simulation-driven greedy optimization. Ablations further show that fixation-specific representations and curriculum learning drive these gains.

Although LEAP performs fast inference after training, fixation-probability estimation still requires repeated stochastic simulations, limiting solution-quality evaluation and comparison with simulation-driven greedy methods. Future work should develop faster fixation-probability estimators or approximation schemes for large-scale validation, and extend the framework to dynamic and co-evolving networks, competitive multi-agent fixation, and tighter connections between reinforcement learning and amplification theory.

8 GenAI Usage Disclosure

We used generative AI tools solely for language editing, such as grammar, punctuation, and clarity improvements, comparable to conventional writing assistants such as Grammarly. These tools were not used to generate original content, ideas, analyses, or claims. All substantive content was authored, reviewed, and verified by the authors.

References

[1] Ben Adlam, Krishnendu Chatterjee, and Martin A Nowak. 2015. Amplifiers of selection. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471, 2181 (2015), 20150114.

[2] Benjamin Allen, Gabor Lippner, Yu-Ting Chen, Babak Fotouhi, Naghmeh Momeni, Shing-Tung Yau, and Martin A Nowak. 2017. Evolutionary dynamics on any population structure. *Nature* 544, 7649 (2017), 227–230.

[3] Benjamin Allen and Martin A Nowak. 2014. Games on graphs. *EMS surveys in mathematical sciences* 1, 1 (2014), 113–151.

[4] Philipp M Altrock, Arne Traulsen, and Martin A Nowak. 2017. Evolutionary games on cycles with strong selection. *Physical Review E* 95, 2 (2017), 022407.

[5] Leslie Ann Goldberg, John Lapinskas, and David Richerby. 2020. Phase transitions of the Moran process and algorithmic consequences. *Random Structures & Algorithms* 56, 3 (2020), 597–647.

[6] Tibor Antal, Sidney Redner, and Vishal Sood. 2006. Evolutionary dynamics on degree-heterogeneous graphs. *Physical review letters* 96, 18 (2006), 188104.

[7] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*. 519–528.

[8] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[9] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. 2004. The architecture of complex weighted networks. *Proceedings of the national academy of sciences* 101, 11 (2004), 3747–3752.

[10] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15, 6 (2003), 1373–1396.

[11] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. doi:10.1145/1553374.1553380

[12] Jnanajyoti Bhaumik and Naoki Masuda. 2024. Constant-selection evolutionary dynamics on weighted networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 480, 2296 (Aug. 2024), 20240223. doi:10.1098/rspa.2024.0223

[13] Stephen P Borgatti and Martin G Everett. 2000. Models of core/periphery structures. *Social networks* 21, 4 (2000), 375–395.

[14] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 946–957.

[15] Ulrik Brandes. 2001. A faster algorithm for betweenness centrality*. *The Journal of Mathematical Sociology* 25, 2 (June 2001), 163–177. doi:10.1080/0022250X.2001.9990249

[16] Joachim Brendborg, Panagiotis Karras, Andreas Pavlogiannis, Asger Ullersted Rasmussen, and Josef Tkadlec. 2022. Fixation maximization in the positional moran process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9304–9312.

[17] David A. Brewster, Martin A. Nowak, and Josef Tkadlec. 2024. Fixation times on directed graphs. *PLOS Computational Biology* 20, 7 (July 2024), e1012299. doi:10.1371/journal.pcbi.1012299

[18] Mark Broom, C Hadjichrysanthou, J Rychtář, and BT Stadler. 2010. Two results on evolutionary processes on general non-directed graphs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 466, 2121 (2010), 2795–2798.

[19] Austin Burt. 2003. Site-specific selfish genes as tools for the control and genetic engineering of natural populations. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270, 1518 (2003), 921–928.

[20] Damon Centola. 2010. The spread of behavior in an online social network experiment. *science* 329, 5996 (2010), 1194–1197.

[21] Ashok K Chandra, Prabhakar Raghavan, Walter L Ruzzo, and Roman Smolensky. 1989. The electrical resistance of a graph captures its commute and cover times. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. 574–586.

[22] Krishnendu Chatterjee, Rasmus Ibsen-Jensen, and Martin A Nowak. 2017. Faster monte-carlo algorithms for fixation probability of the moran process on undirected graphs. *arXiv preprint arXiv:1706.06931* (2017).

[23] Tiantian Chen, Siwen Yan, Jianxiong Guo, and Weili Wu. 2023. ToupleGDD: A fine-designed solution of influence maximization by deep reinforcement learning. *IEEE Transactions on Computational Social Systems* 11, 2 (2023), 2210–2221.

[24] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, California, USA) *(KDD '11)*. Association for Computing Machinery, New York, NY, USA, 1082–1090. doi:10.1145/2020408.2020579

[25] Fan RK Chung. 1996. Lectures on spectral graph theory. *CBMS Lectures, Fresno* 6, 92 (1996), 17–21.

[26] Fan RK Chung. 1997. *Spectral graph theory*. Vol. 92. American Mathematical Soc.

[27] Tzu-Yuan Chung, Khurshed Ali, and Chih-Yu Wang. 2019. Deep reinforcement learning-based approach to tackle competitive influence maximization. In *Proc. of MLG workshop*.

[28] PETER CLIFFORD and AIDAN SUDBURY. 1973. A model for spatial conflict. *Biometrika* 60, 3 (Dec. 1973), 581–588. doi:10.1093/biomet/60.3.581

[29] Josep Díaz, Leslie Ann Goldberg, George B Mertzios, David Richerby, Maria Serna, and Paul G Spirakis. 2014. Approximating fixation probabilities in the generalized moran process. *Algorithmica* 69, 1 (2014), 78–91.

[30] Josep Díaz, Leslie Ann Goldberg, David Richerby, and Maria Serna. 2016. Absorption time of the Moran process. *Random Structures & Algorithms* 49, 1 (2016), 137–159.

[31] Peter G Doyle and J Laurie Snell. 1984. *Random walks and electric networks*. Vol. 22. American Mathematical Soc.

[32] P. Erdös and A. Rényi. 2006. *On the evolution of random graphs*. Princeton University Press, Princeton, 38–82. doi:doi:10.1515/9781400841356.38

[33] Kevin M Esvelt, Andrea L Smidler, Flaminia Catteruccia, and George M Church. 2014. Concerning RNA-guided gene drives for the alteration of wild populations. *elife* 3 (2014), e03401.

[34] Eyal Even-Dar and Asaf Shapira. 2011. A note on maximizing the spread of influence in social networks. *Inform. Process. Lett.* 111, 4 (Jan. 2011), 184–187. doi:10.1016/j.ipl.2010.11.015

[35] Michael Fire, Rami Puzis, and Yuval Elovici. 2012. Link prediction in highly fractional data sets. In *Handbook of computational approaches to counterterrorism*. Springer, 283–300.

[36] Andreas Galanis, Andreas Göbel, Leslie Ann Goldberg, John Lapinskas, and David Richerby. 2017. Amplifiers for the Moran process. *Journal of the ACM (JACM)* 64, 1 (2017), 1–90.

[37] Smita Ghosh, Tiantian Chen, and Weili Wu. 2024. Enhanced group influence maximization in social networks using deep reinforcement learning. *IEEE Transactions on Computational Social Systems* (2024).

[38] George Giakkoupis. 2016. Amplifiers and suppressors of selection for the moran process on undirected graphs. *arXiv preprint arXiv:1611.01585* (2016).

[39] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J Watts. 2016. The structural virality of online diffusion. *Management science* 62, 1 (2016), 180–196.

[40] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[41] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web* (New York, NY, USA) *(WWW '04)*. Association for Computing Machinery, New York, NY, USA, 403–412. doi:10.1145/988672.988727

[42] Mahdi Hajihashemi and Keivan Aghababaei Samani. 2021. Path to fixation of evolutionary processes in graph-structured populations. *The European Physical Journal B* 94, 2 (2021), 51.

[43] Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double Q-Learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2094–2100.

[44] Laura Hindersin and Arne Traulsen. 2014. Counterintuitive properties of the fixation time in network-structured populations. *Journal of The Royal Society Interface* 11, 99 (2014), 20140606.

[45] Paul Holland, Kathryn B. Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks* 5 (1983), 109–137.

[46] Yuni Iwamasa and Naoki Masuda. 2014. Networks maximizing the consensus time of voter models. *Physical Review E* 90, 1 (2014), 012816.

[47] Ian Jolliffe. 2011. Principal component analysis. In *International encyclopedia of statistical science*. Springer, 1094–1096.

[48] Michael L Katz and Carl Shapiro. 1985. Network externalities, competition, and compatibility. *The American economic review* 75, 3 (1985), 424–440.

[49] Matt J Keeling and Ken TD Eames. 2005. Networks and epidemic models. *Journal of the royal society interface* 2, 4 (2005), 295–307.

[50] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.

[51] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. 2017. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems* 30 (2017).

[52] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. 2010. Identification of influential

spreaders in complex networks. *Nature physics* 6, 11 (2010), 888–893.
[53] Konstantin Klemm, M Ángeles Serrano, Víctor M Eguíluz, and Maxi San Miguel. 2012. A measure of individual role in collective dynamics. *Scientific reports* 2, 1 (2012), 292.
[54] Wouter Kool, Herke Van Hoof, and Max Welling. 2018. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475* (2018).
[55] ANIKÓ KOPACZ. 2024. Competitive influence maximization in trust-based social networks with deep q-learning. *Studia Universitatis Babes-Bolyai, Informatica* 69, 1 (2024).
[56] Gouri Kundu, Smita Ghosh, and Sankhayan Choudhury. 2024. Rumor Containment in Hypergraph Representation of Social Networks: A Deep Reinforcement Learning-Based Solution. *IEEE Transactions on Computational Social Systems* (2024).
[57] Yang Ping Kuo and Oana Carja. 2024. Evolutionary graph theory beyond pairwise interactions: higher-order network motifs shape times to fixation in structured populations. *PLOS Computational Biology* 20, 3 (2024), e1011905.
[58] Yang Ping Kuo, César Nombela-Arrieta, and Oana Carja. 2024. A theory of evolutionary dynamics on any complex population structure reveals stem cell niche architecture as a spatial suppressor of selection. *Nature Communications* 15, 1 (2024), 4666.
[59] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 420–429.
[60] Hui Li, Mengting Xu, Sourav S Bhowmick, Joty Shafiq Rayhan, Changsheng Sun, and Jiangtao Cui. 2022. PIANO: Influence maximization meets deep reinforcement learning. *IEEE Transactions on Computational Social Systems* 10, 3 (2022), 1288–1300.
[61] Erez Lieberman, Christoph Hauert, and Martin A. Nowak. 2005. Evolutionary dynamics on graphs. *Nature* 433, 7023 (Jan. 2005), 312–316. doi:10.1038/nature03204
[62] Thomas M. Liggett. 1985. *Interacting Particle Systems*. Springer Berlin Heidelberg. doi:10.1007/b138374
[63] Ruodan Liu and Naoki Masuda. 2023. Fixation dynamics on hypergraphs. *PLOS Computational Biology* 19, 9 (Sept. 2023), e1011494. doi:10.1371/journal.pcbi.1011494
[64] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
[65] Wes Maciejewski. 2014. Reproductive value in graph-structured populations. *Journal of Theoretical Biology* 340 (2014), 285–293.
[66] Sahil Manchanda, AKASH MITTAL, Anuj Dhawan, Sourav Medya, Sayan Ranu, and Ambuj Singh. 2020. GCOMB: Learning Budget-constrained Combinatorial Algorithms over Billion-sized Graphs. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 20000–20011.
[67] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. 2017. Random walks and diffusion on networks. *Physics reports* 716 (2017), 1–58.
[68] Julian McAuley and Jure Leskovec. 2012. Learning to discover social circles in ego networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada) *(NIPS'12)*. Curran Associates Inc., Red Hook, NY, USA, 539–547.
[69] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
[70] Javad Mohamadichamgavi and Jacek Miękisz. 2024. Effect of the degree of an initial mutant in Moran processes in structured populations. *Physical Review E* 109, 4 (2024), 044406.
[71] Marius Möller, Laura Hindersin, and Arne Traulsen. 2019. Exploring and mapping the universe of evolutionary graphs identifies structural properties affecting fixation probability and time. *Communications biology* 2, 1 (2019), 137.
[72] T. Monk, P. Green, and M. Paulin. 2014. Martingales and fixation probabilities of evolutionary graphs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 470, 2165 (May 2014), 20130730. doi:10.1098/rspa.2013.0730
[73] Patrick Alfred Pierce Moran. 1958. Random processes in genetics. In *Mathematical proceedings of the cambridge philosophical society*, Vol. 54. Cambridge University Press, 60–71.
[74] Mark EJ Newman. 2002. Spread of epidemic disease on networks. *Physical review E* 66, 1 (2002), 016128.
[75] Martin A Nowak and Robert M May. 1992. Evolutionary games and spatial chaos. *nature* 359, 6398 (1992), 826–829.
[76] Hisashi Ohtsuki, Christoph Hauert, Erez Lieberman, and Martin A Nowak. 2006. A simple rule for the evolution of cooperation on graphs and social networks. *Nature* 441, 7092 (2006), 502–505.
[77] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab.
[78] Romualdo Pastor-Satorras and Alessandro Vespignani. 2001. Epidemic spreading in scale-free networks. *Physical review letters* 86, 14 (2001), 3200.
[79] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 8024–8035.
[80] Zaheerabbas Patwa and Lindi M Wahl. 2008. The fixation probability of beneficial mutations. *Journal of The Royal Society Interface* 5, 28 (2008), 1279–1289.
[81] Andreas Pavlogiannis, Josef Tkadlec, Krishnendu Chatterjee, and Martin A. Nowak. 2018. Construction of arbitrarily strong amplifiers of natural selection using evolutionary graph theory. *Communications Biology* 1, 1 (June 2018), 71. doi:10.1038/s42003-018-0078-7
[82] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
[83] Sidney Redner. 2001. *A guide to first-passage processes*. Cambridge university press.
[84] Everett M Rogers, Arvind Singhal, and Margaret M Quinlan. 2014. Diffusion of innovations. In *An integrated approach to communication theory and research*. Routledge, 432–448.
[85] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. http://networkrepository.com
[86] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. 2020. GEMSEC: graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (Vancouver, British Columbia, Canada). Association for Computing Machinery, New York, NY, USA, 65–72. doi:10.1145/3341161.3342890
[87] Daniel A Spielman. 2007. Spectral graph theory and its applications. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 29–38.
[88] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[89] Jakub Svoboda, Soham Joshi, Josef Tkadlec, and Krishnendu Chatterjee. 2024. Amplifiers of selection for the Moran process with both Birth-death and death-Birth updating. *PLOS Computational Biology* 20, 3 (March 2024), e1012008. doi:10.1371/journal.pcbi.1012008
[90] György Szabó and Gabor Fath. 2007. Evolutionary games on graphs. *Physics reports* 446, 4-6 (2007), 97–216.
[91] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 75–86.
[92] Peter D Taylor and Leo B Jonker. 1978. Evolutionary stable strategies and game dynamics. *Mathematical biosciences* 40, 1-2 (1978), 145–156.
[93] Josef Tkadlec, Andreas Pavlogiannis, Krishnendu Chatterjee, and Martin A Nowak. 2019. Population structure determines the tradeoff between fixation probability and fixation time. *Communications biology* 2, 1 (2019), 138.
[94] Josef Tkadlec, Andreas Pavlogiannis, Krishnendu Chatterjee, and Martin A. Nowak. 2021. Fast and strong amplifiers of natural selection. *Nature Communications* 12, 1 (June 2021), 4009. doi:10.1038/s41467-021-24271-w
[95] Jeremy Van Cleve. 2023. Evolutionarily stable strategy analysis and its links to demography and genetics through invasion fitness. *Philosophical transactions of the Royal Society B* 378, 1876 (2023), 20210496.
[96] Jingwen Wang, Zhoulin Cao, Chunzhi Xie, Yanli Li, Jia Liu, and Zhisheng Gao. 2025. DGN: influence maximization based on deep reinforcement learning. *The Journal of Supercomputing* 81, 1 (2025), 130.
[97] Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684 (1998), 440–442. doi:10.1038/30918
[98] Teh-Hsing Wei. 1952. *The Algebraic Foundations of Ranking Theory*. Doctoral thesis. University of Cambridge. https://www.repository.cam.ac.uk/handle/1810/250988
[99] Reza Zafarani and Huan Liu. 2014. Users Joining Multiple Sites: Distributions and Patterns. *Proceedings of the International AAAI Conference on Web and Social Media* 8, 1 (May 2014), 635–638. doi:10.1609/icwsm.v8i1.14564