

# Random Walk Learning and the Pac-Man Attack

Xingran Chen, Parimal Parag, Rohit Bhagat, Zonghong Liu, and Salim El Rouayheb

**Abstract**—Random walk (RW)-based algorithms have long been popular in distributed systems due to low overheads and scalability, with recent growing applications in decentralized learning. However, their reliance on local interactions makes them inherently vulnerable to malicious behavior. In this work, we investigate an adversarial threat that we term the “Pac-Man” attack, in which a malicious node probabilistically terminates any RW that visits it. This stealthy behavior gradually eliminates active RWs from the network, effectively halting the decentralized operators without triggering failure alarms. To counter this threat, we propose the AVERAGE CROSSING (AC) algorithm—a fully decentralized mechanism for duplicating RWs to prevent RW extinction in the presence of Pac-Man. Our theoretical analysis establishes that (i) the RW population remains almost surely bounded under AC and (ii) RW-based stochastic gradient descent remains convergent under AC, even in the presence of Pac-Man, with a quantifiable deviation from the true optimum. Furthermore, they uncover a phase transition in the extinction probability as a function of the duplication threshold, which we explain through theoretical analysis of a simplified variant of AC. Our extensive empirical results on both synthetic and public benchmark datasets validate our theoretical findings.

## I. INTRODUCTION

Decentralized algorithms are becoming increasingly important in modern large-scale networked applications. These algorithms enable a network of nodes/agents, each with access only to local data and a limited local view of the connection graph, to collaborate in solving global computational tasks without centralized coordination. Among the most widely studied approaches are random-walk (RW)-based algorithms [1], [2] and gossip-based algorithms [3].

While gossip-based methods are powerful, their repeated local broadcasts can lead to large communication overhead in large-scale systems [3]. This motivates the study of RW-based algorithms, which offer a communication-efficient alternative for large networks due to their simplicity and scalability [1], [2]. As a result, RW-based methods have been successfully applied across a wide range of domains [4]–[8]. A particularly compelling application is decentralized machine learning, where RWs are used to aggregate learning on data distributed across networked agents [9], [10].

Despite their advantages, RW learning algorithms are susceptible to security threats [11], [12]. We focus in this work on a particular threat in which certain nodes behave maliciously by terminating, or “killing”, any random walk that reaches

them. We refer to this as the *Pac-Man attack*, evoking the arcade character known for devouring everything in its path.

Recent work by [13] introduced a novel approach based on *self-regulating* RWs, aimed at enhancing the resilience of RW-based algorithms in a decentralized manner. The main algorithm proposed in this work is called DECAFORK, which operates by maintaining a history of RW visit times at each node. The work in [13] provides theoretical guarantees for catastrophic failure scenarios under idealized assumptions and approximations. The followup work in [14] includes numerical simulations evaluating the DECAFORK’s behavior in the presence of malicious nodes.

In this work, to counter the Pac-Man attack, we propose a new decentralized duplication mechanism, termed the AVERAGE CROSSING (AC) algorithm (see Algorithm 1). In the AC algorithm, each benign node independently decides whether to duplicate an incoming RW based solely on local timing information—specifically, the time interval between two successive visits from any RW. If this interval exceeds a predetermined threshold, the node infers that at least one RW may have been killed by the Pac-Man and, with a given probability, duplicates the currently visiting RW. In the following, we summarize the main contributions of this article.

(i) *Novel theoretical framework and analytical results*: We develop a novel and rigorous theoretical framework to analyze the evolution of the number of active RWs under AC, explicitly accounting for the strong interdependence among RWs. Within this framework, we establish that the RW population under AC is almost surely bounded (Theorem 1).

Beyond boundedness, we further reveal a phase transition behavior of the RW population with respect to the duplication threshold. Specifically, for appropriately chosen thresholds, the RW population persists with positive probability, whereas overly large thresholds lead to almost sure extinction. To theoretically characterize this phenomenon, we introduce a simplified and analytically tractable variant of the AC algorithm, under which we rigorously establish the existence of a phase transition (see Proposition 1).

(ii) *Integration with decentralized learning and proof of convergence*: The AC mechanism can be seamlessly integrated into random walk stochastic gradient descent (RW-SGD) [9], [15], [16]. We prove that RW-SGD converges under the AC mechanism in the presence of an adversary (see Theorem 2). We further characterize the bias induced by premature RW termination, showing how adversarial behavior skews the optimizer and leads to a bounded deviation from the true global optimum (Proposition 2).

Xingran Chen, Rohit Bhagat, Zonghong Liu, and Salim El Rouayheb are with Department of Electrical and Computer Engineering, Rutgers University, Piscataway Township, NJ 08854, USA (E-mail: xingranc@iecc.org, {rb1395, zl304, sye8}@scarletmail.rutgers.edu).

Parimal Parag is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, Karnataka 560012, India (E-mail: parimal@iisc.ac.in).

## II. PROBLEM FORMULATION

### A. Graph and Random Walks

These agents are modeled as nodes in a *connected* undirected graph, where each agent possesses its own local data and can communicate only with neighboring agents.<sup>1</sup> Without loss of generality, we assume that the Pac-Man agent is indexed as node 1. The communication topology graph is constructed as a finite graph  $\mathcal{G} \triangleq (\mathcal{V}, E)$ , where the set of agents/nodes is  $\mathcal{V} \triangleq [N]$ , and  $E \subseteq \binom{\mathcal{V}}{2}$  denotes the set of edges. In addition, let  $\mathcal{B} \triangleq \mathcal{V} \setminus \{1\}$  denote the set of benign nodes.

In this system, computation is carried out via RWs on the graph. Each RW is determined by a fixed and identical transition probability matrix  $P$ . Moreover, each RW carries a token message (which can be viewed as a global task) that is processed and passed along the network. At each time step, only the node currently holding a token performs local computation and updates the message. After computation, the current node forwards the token to a randomly selected neighbors based on the transition probability matrix  $P$ . This process continues until a predefined stopping criterion is satisfied. When multiple RWs are present, each is uniquely identifiable (e.g., via an index  $j$ ) to track their individual progress through the network.

### B. Threat Model: The Pac-Man Attack

We are interested in ensuring the resiliency of RW-based algorithms—that is, their ability to prevent total extinction and continue operating effectively even when a malicious node attempts to terminate the RWs. Specifically, we focus on a threat we refer to as the *Pac-Man attack*: a malicious node, termed *Pac-Man*, terminates all incoming RWs without performing the required computation or forwarding the result to a neighbor<sup>2</sup>. This setting already captures the core difficulty of the problem: even one such adversary is sufficient to cause eventual extinction of all RWs with probability one, thereby completely halting the global task. The Pac-Man is particularly dangerous due to its ability to remain hidden through a deceptive behavior.

- (i) The Pac-Man node can reply positively to all network-level fault-tolerance checks and controls, for example re-transmissions and timeouts [17]–[21], making it difficult to distinguish it from benign nodes despite its malicious actions.
- (ii) To avoid detection, the Pac-Man node terminates incoming RWs independently with a probability  $\zeta \in (0, 1]$ , referred to as the termination probability. This randomized behavior allows the Pac-Man to conceal itself

among benign nodes. If the Pac-Man node terminates all incoming RWs, it would never propagate RWs to its neighbors, making it easily identifiable as malicious since no RWs would be observed from that node over a long time horizon. In contrast, by terminating incoming RWs only with a positive probability, the Pac-Man can intermittently forward RWs, thereby blending in with normal nodes.

One commonly used fault-tolerance technique is the introduction of redundancy. However, static redundancy alone is *ineffective* in this adversarial setup. Simply starting with multiple RWs does not guarantee their survival: for any small termination probability  $\zeta > 0$ , all RWs will eventually be terminated with probability 1, causing the task to fail. Thus, redundancy cannot serve as a long-term solution.

### C. RW-based Stochastic Gradient Descent

We consider the following standard distributed optimization and learning problem:

$$\min_{\mathbf{x} \in \mathbb{R}^m} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathbb{R}^m} \mathbb{E}_{u \sim \pi} [f_u(\mathbf{x})], \quad (1)$$

where each node  $u$  possesses a local function  $f_u(\mathbf{x})$ ,  $\pi$  is a target sampling distribution, and  $m \in \mathbb{Z}_+$ . Here, we assume  $[\pi]_u > 0$  for  $u \in \mathcal{V}$ .

Under local communication constraints, solving problem (1) using RW-SGD algorithms has proven to be highly effective [22]–[24]. Given a target sampling distribution  $\pi$  and a connected graph  $\mathcal{G}$ , The Metropolis–Hastings algorithm [23] constructs a RW on  $\mathcal{G}$  with transition matrix  $P$ , for which  $\pi$  is the stationary distribution.<sup>3</sup> The RW-SGD is outlined as  $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \hat{g}_{v_t}(\mathbf{x}_t)$ , where  $v_t$  is the node visited by the random walk at time  $t$ , and  $\hat{g}_{v_t}(\mathbf{x}_t)$  denotes the gradient or sub-gradient at that node. Here, the sequence  $\{v_t\}_{t \in \mathbb{N}}$  evolves according to  $P$ .

### D. Designable Properties of the Decentralized Mechanism

The entire goal is to design a *decentralized mechanism* such that (i) the global task can be carried out via RW-based algorithms even in the presence of a Pac-Man node, and (ii) when the proposed decentralized mechanism is integrated with RW-SGD, it remains effective and convergent.<sup>4</sup>

At the beginning of time slot  $t$ , let  $\mathcal{Z}_t$  denote the set of indices of active RWs, and define the random variable  $Z_t \triangleq |\mathcal{Z}_t|$  as the total number of active RWs at that moment. For each RW  $j \in \mathcal{Z}_t$  at time  $t$ , we denote its location at time  $t$  by  $X_j(t) \in \mathcal{V}$ . At the initial time  $t = 0$ , we denote the number of initial RWs as  $Z_0 = z_0$ , where  $z_0$  is a predetermined scalar. To achieve our entire goal, the proposed algorithm must have the following desirable properties.

<sup>1</sup>Without loss of generality, we restrict our analysis to connected graphs. For disconnected graphs, the proposed theoretical framework can be applied separately to each connected component, and the analysis proceeds analogously.

<sup>2</sup>In this paper, we focus on the case of a single Pac-Man node to simplify the presentation of the algorithm and its theoretical analysis. The proposed framework naturally extends to multiple Pac-Man nodes; see Remark 2 in Appendix A for further discussion.

<sup>3</sup>The transition matrix  $P$  depends on both the target sampling distribution  $\pi$  and the graph  $\mathcal{G}$ . We re-parameterize it as  $P_{\pi, \mathcal{G}} \in \mathbb{R}^{N \times N}$ , and simply write  $P$  when there is no risk of confusion.

<sup>4</sup>In the absence of adversarial behavior, RW-SGD is known to converge to the global optimum [22]–[24]. However, the introduction of a Pac-Man node fundamentally alters the dynamics. Since this adversarial node probabilistically terminates incoming RWs, it is no longer clear whether RW-SGD remains convergent.

a) **No Blowup:** To ensure system stability, we must avoid uncontrolled growth in the number of RWs. The algorithm should keep the RW population bounded with probability one:

$$\Pr\left(\sup_t Z_t < \infty \mid Z_0 = z_0\right) = 1. \quad (2)$$

b) **Low Probability of Extinction:** To ensure the resiliency of the system, it is essential to sustain the RW population over time. The algorithm should keep that the probability that the Pac-Man attack eliminates all RWs is small. Let  $\delta$  be a small threshold, we have

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) < \delta. \quad (3)$$

c) **Convergence:** All active RWs under RW-SGD must converge to the same minimizer, in the following sense:

$$\lim_{t \rightarrow \infty} \mathbb{E}[\|\mathbf{x}_t^{(j)} - \mathbf{x}^*\|] = 0, \forall j \in \mathcal{Z}_t, \quad (4)$$

where  $\mathbf{x}^*$  denotes the convergent point attained by the RW-SGD in the absence of a Pac-Man node. If the limit in (4) exists, we further aim to characterize the approximation error with respect to the true optimizer  $\mathbf{x}^*$ , by bounding the deviation  $\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\|$ .

### III. AVERAGE CROSSING ALGORITHM

We present the AVERAGE CROSSING (AC) algorithm, a decentralized mechanism for adaptively duplicating RWs to meet the designable properties outlined in Section II-D. We outline the AC algorithm in Algorithm 1.

The core idea is as follows: each benign node  $u \in \mathcal{B}$  maintains a variable  $L_t^{(u)}$ , representing the most recent time prior to time  $t$  at which node  $u$  was visited by any RW. If node  $u$  has not been visited for too long, i.e., the interval  $t - L_t^{(u)}$  exceeds a predetermined threshold  $A_u$ , node  $u$  suspects that at least one RW may have been lost. Then, with probability  $q$ , it duplicates a new RW as an *identical* copy of the currently visiting one (see lines 5-6 in Algorithm 1). If multiple RWs arrive at node  $u$  at time  $t$  and the same condition met, node  $u$  selects one of the arriving RWs uniformly at random and duplicates it with probability  $q$ .

### IV. FUNDAMENTAL ANALYSIS

In this section, we present a rigorous theoretical analysis of the AC algorithm, focusing on the designable properties outlined in Section II-D. For brevity, the preliminaries—including notations, definitions, and assumptions—are deferred to Appendix A.

#### A. Population Boundedness

First, we establish that the number of active RWs remains almost surely bounded at all times, despite ongoing duplications. This property is crucial for ensuring that the algorithm is stable and does not flood the network with RWs.

**Theorem 1** (Boundedness). *On any finite graph  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$  of Definition 4 in Appendix A, with  $z_0 \geq 1$ ,  $A_u \geq 1$ ,  $q \leq 1$ , and*

---

#### Algorithm 1 Average Crossing (AC) Algorithm

---

```

1: Input: The graph  $\mathcal{G}$ , the thresholds  $\{A_u\}_{u \in [N]}$ , the initial
   recording  $L_0^{(u)} = 0$  for  $u \in [N]$ , the forking probability
    $q$ , and the initial location of RWs  $\mathcal{Z}_0$ .
2: for  $t \geq 0$  do
3:   for  $u \in \mathcal{B}$  do
4:     if  $u \in \cup_{j \in \mathcal{Z}_t} X_{t,j}$  then.
5:       if  $u \in [N]$  and  $t - L_t^{(u)} > A_u$  then
6:         With probability  $q$ , node  $u$  forks a new RW
         by uniformly copying one of the currently visiting RWs.
7:       end if
8:        $L_t^{(u)} \leftarrow t$ .
9:     end if
10:   end for
11: end for

```

---

$\zeta \in (0, 1]$ , the AC algorithm ensures that  $\limsup_{t \rightarrow \infty} Z_t < \infty$  almost surely.

*Proof.* The full proof is given in Appendix B.  $\square$

Theorem 1 implies that regardless of the thresholds  $\{A_u\}_{u \in \mathcal{B}}$  selected by the benign nodes, the RW population remains bounded over time almost surely.

#### B. Phase Transition in Extinction Behavior

Second, we study how the duplication threshold affects whether the RW population survives or dies out. Intuitively, a small threshold leads to frequent duplications, increasing the chance of survival, while a large threshold delays responses and may result in extinction. Numerical experiments (Fig. 2 in Section V and Fig 6 in Appendix G) support this intuition, showing a soft phase transition in extinction probability as the threshold decreases. In particular, by choosing the duplication threshold appropriately, the extinction probability can be reduced to a very small positive value.

However, analyzing the AC algorithm is challenging due to the implicit correlations in the duplication decisions across nodes. Upon each visit, node  $u$  decides whether to duplicate by evaluating  $t - L_t^{(u)}$ , which reflects the cumulative influence of active RWs' trajectories from the initial time to the current moment. As a result, a benign node's duplication behavior is indirectly affected by the entire set of active RWs. These hidden correlations significantly complicate the analysis.

To circumvent this difficulty and better understand the extinction behavior, we consider a simplified variant called the *Weak Version of AC* (W-AC) algorithm, formally defined in Algorithm 2 in Appendix C. The key difference is that, in W-AC, each RW decides autonomously when to duplicate based only on the time since its last visit to the current node, whereas in AC the duplication decision is made by the node.

**Proposition 1** (Soft Phase Transition). *Consider  $z_0$  initial uniform RWs on an almost fully connected graph from Definition 12 (in Appendix C). Then, under the W-AC algorithm defined*

in Algorithm 2, there exist two non-negative constants  $\bar{\alpha} \geq \underline{\alpha}$ , such that, for all  $z_0 \geq 1$ ,

$$\begin{cases} \Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) = 1 & \text{if } A \geq \bar{\alpha} \\ \Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) < 1 & \text{if } A \leq \underline{\alpha}. \end{cases}$$

*Proof.* The proof is given in Appendix C.  $\square$

### C. Convergence

Now, we study how the AC algorithm affects the convergence of RW-SGD in the presence of a Pac-Man node. It is important to note that each RW will eventually be terminated by the Pac-Man node with probability one. To address this, rather than relying on a single RW, we need to analyze the behavior of a *chain* of RWs. Specifically, whenever a RW duplicates, we refer to the original as the *parent* and the resulting RW as its *child*.

**Definition 1** (Chain of RWs). Start with an initial RW  $j_0 \in [z_0]$ . At each step  $s \geq 0$ , if  $j_s$  has at least one child, let  $j_{s+1}$  be a uniformly chosen child of  $j_s$ ; otherwise, set  $j_{s'} = -1$  for all  $s' \geq s$  to mark termination. We refer to  $\{j_s\}_{s \in \mathbb{N}}$  as a chain of RWs. We call the chain *infinite* if  $\lim_{s \rightarrow \infty} j_s > 0$ ; otherwise, we call it *finite*.

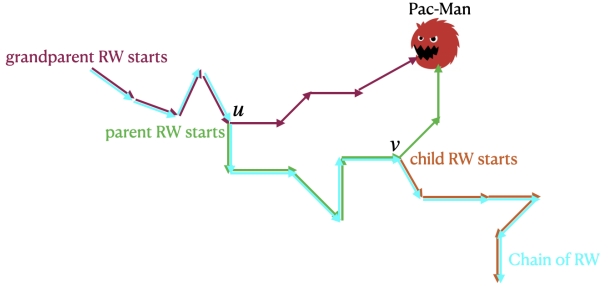


Fig. 1: Illustration of a chain of RWs. Node  $u$  duplicates a new RW (green) by copying the purple RW. Node  $v$  duplicates a new RW (orange) by copying the green RW. The blue trajectory shows how these RWs are connected over time, forming a chain.

We call a trajectory  $\{Z_t\}_{t \in \mathbb{N}}$  *surviving* if  $\inf_{t \in \mathbb{N}} Z_t > 0$ . Under a surviving trajectory  $\{Z_t\}_{t \in \mathbb{N}}$ , there must exist at least one infinite chain of RWs; otherwise, if all chains of RWs are finite, then there exists some  $t_0 \in \mathbb{Z}$  such that  $|Z_t| = 0$  for  $t \geq t_0$ , which contradicts the definition of surviving trajectories.

**Theorem 2.** Under the AC mechanism and along a surviving trajectory  $\{Z_t\}_{t \geq 0}$ , RW-SGD converges to the minimizer of the following problem:

$$\min_{\mathbf{x} \in \mathbb{R}^m} \tilde{f}(\mathbf{x}) = \min_{\mathbf{x} \in \mathbb{R}^m} \mathbb{E}_{u \sim \pi_{\text{chain}}^{(\zeta)}} [f_u(\mathbf{x})], \quad (5)$$

where

$$\pi_{\text{chain}}^{(\zeta)} = \begin{cases} [0, \nu^{(1)}] & \zeta = 1, \\ \nu^{(\zeta)} & \zeta \in (0, 1), \end{cases} \quad (6)$$

and  $\nu^{(\zeta)}$  is defined in Definition 6 in Appendix A.

*Proof.* The full proof is given in Appendix D.  $\square$

We treat a chain of RWs as a *single effective* RW. The corresponding effective transition probability matrix  $P_{\text{chain}}^{(\zeta)}$  is provided in Appendix E. This matrix will be used to quantify the deviation of the new minimizer from the true global optimum  $\mathbf{x}^*$ . In the standard RW-SGD algorithm, when the stepsize  $\eta_t$  decreases with time  $t$  and tends to 0, the algorithm converges to a deterministic point; if  $\eta_t$  remains constant, it converges to a random variable [22]–[24].

**Proposition 2** (Shift of Optima). Let  $P_{\text{chain}}^{(\zeta)}$  be defined in Appendix E, with spectral gap  $\gamma_{\text{chain}}^{(\zeta)}$ . Let  $\|\cdot\|_{TV}$  denote the total variation distance, and let  $\mathbf{x}_0$  be the starting point. Under the setting of Theorem 2:

(i) If the stepsize  $\eta_t \downarrow 0$ , then RW-SGD converges to the minimizer  $\tilde{\mathbf{x}}^*$  of (5). Moreover,

$$\frac{1}{L} \|\nabla f(\tilde{\mathbf{x}}^*)\| \leq \|\tilde{\mathbf{x}}^* - \mathbf{x}^*\| \leq \frac{1}{\mu} \|\nabla f(\tilde{\mathbf{x}}^*)\|.$$

(ii) If the stepsize  $\eta_t = \eta < \frac{1}{L}$ , then

$$\mathbb{E}[\|\tilde{\mathbf{x}}_T - \mathbf{x}^*\|^2] \leq 2(1 - \gamma\mu)^T \|\mathbf{x}_0 - \mathbf{x}^*\|^2 + \frac{\eta L \sigma^2}{\gamma_{\text{chain}}^{(\zeta)} \mu^2} + \frac{\|\tilde{\nu}^{(\zeta)} - \pi\|_{TV}^2 \sigma^2 L}{\mu^3}.$$

*Proof.* The proof is given in Appendix F.  $\square$

## V. SIMULATIONS

In the main text, we only present representative simulations, and additional simulation results are provided in Appendix G.

### A. Simulation Setup

a) *Graph Settings:* We consider a complete graph with 100 nodes, consisting of one Pac-Man node and 99 benign nodes. The target sampling distribution is uniform, i.e.,  $\pi = (\frac{1}{100}, \frac{1}{100}, \dots, \frac{1}{100})$ . We assume  $A_u = A$  for all  $u \in \mathcal{B}$ . We set both the duplication and termination probabilities to 1, i.e.,  $q = \zeta = 1$ . We also consider three other connected graphs, including regular, ring, and Erdős-Rényi graphs; details are provided in Appendix G-A.

b) *Learning Settings:* For the distributed learning problem defined in (1), we evaluate our approach on both synthetic and public benchmark datasets.

**Synthetic dataset.** We consider a decentralized linear regression task, where each node  $u$  minimizes a local mean squared error (MSE) loss of the form:  $f_u(\mathbf{x}) = (\mathbf{w}^T \mathbf{x} + b - y_u)^2$ , where  $\mathbf{x}$  is the input feature,  $y_u$  is the target label at node  $u$ ,  $\mathbf{w}$  is the weight vector, and  $b$  is the bias. We assume each node holds only a single data point. This local objective  $f_u(\mathbf{x})$  is strongly convex and  $L$ -smooth.

**Public benchmark dataset.** We use the MNIST handwritten digit dataset [25]. The dataset is evenly partitioned into 100 disjoint subsets, with each node is assigned a unique subset. Each node  $u$  minimizes the empirical cross-entropy loss over its local dataset  $\mathcal{D}_u$ :  $f_u(w) = \frac{1}{|\mathcal{D}_u|} \sum_{(\mathbf{x}, y) \in \mathcal{D}_u} \ell(w; \mathbf{x}, y)$ ,

where  $w$  denotes the model parameters,  $\ell(w; \mathbf{x}, y)$  is the cross-entropy loss function, and  $\mathcal{D}_u$  is the local subset assigned to node  $u$ . Both *i.i.d.* and non-*i.i.d.* data partitioning schemes for distributing data are considered; further details are provided in Appendix G-B.<sup>5</sup>

*c) Baselines:* To the best of our knowledge, the algorithm most closely related to our work is the DECAFORK algorithm [13], [14]. To ensure a fair comparison, DECAFORK method adopts the same random walk transition matrix  $P$ , as used in the AC algorithm, so their movement behavior remains identical.

Another baseline we consider is the classical GOSSIP-BASED SGD [3]. We incorporate the same adversarial setting as before. As shown in Fig. 4 in Appendix G-C, GOSSIP-BASED SGD exhibits significantly slow convergence in the Pac-Man setting. Consequently, we conclude that it is not a suitable baseline and exclude it from the remaining simulation results.

### B. Boundedness and Persistence

We begin by validating Theorem 1. In Fig. 2(a), the  $y$ -axis shows the number of RWs, averaged over 100 iterations, while the  $x$ -axis denotes the time steps. We first allow the RWs to traverse without the presence of the Pac-Man for a short phase ( $T_0 = 2000$ ). We then introduce the Pac-Man to run the AC algorithm over a longer time horizon ( $T = 50000$ ). The RW population process  $\{Z_t\}_{t \in \mathbb{Z}}$  remains bounded over time.

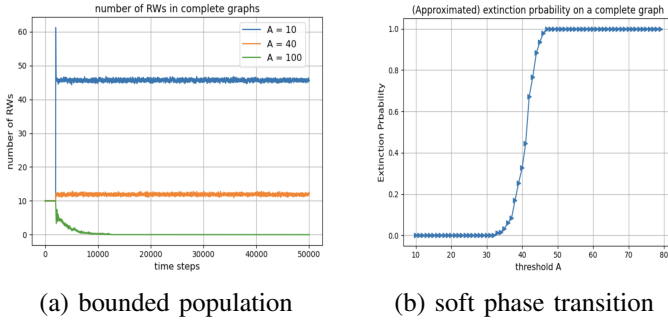


Fig. 2: Behaviors of  $\{Z_t\}_{t \geq 0}$  on the complete graph.

Then, we validate the persistence. Although Proposition 1 focuses exclusively on the W-AC algorithm, we extend our investigation by simulating the AC algorithm extinction behavior. Fig. 2(b) illustrates a soft phase transition in the extinction probability as a function of  $A$ . The  $y$ -axis shows the approximated extinction probability<sup>6</sup>, while the  $x$ -axis denotes the value of  $A$ . when  $A$  exceeds a critical value (which depends on the graph topology), extinction occurs with probability 1. Conversely, when  $A$  falls below this critical threshold, the extinction probability drops sharply and approaches zero for small values of  $A$ .

<sup>5</sup>Results under *i.i.d.* partitioning are provided in Appendix G-E, while the main text focuses on simulations under non-*i.i.d.* partitioning.

<sup>6</sup>We approximate the extinction probability by running a large number of simulations over a long time horizon and computing the ratio of runs in which the RW population goes extinct to the total number of runs.

### C. Convergence

Figs. 3(a) and (b) present the convergence performance of RW-SGD under the proposed AC algorithm, in comparison with the DECAFORK baseline, on the synthetic and public benchmark datasets, respectively. The  $y$ -axis represents the loss curves,<sup>7</sup> while the  $x$ -axis denotes the number of time steps. From the figure, both loss functions demonstrate effective convergence, which demonstrates Theorem 2. Notably, the loss curves under RW-SGD are nearly identical across AC and DECAFORK. This observation is expected, the distributed optimization (1) depends on the movement of active RWs, which is governed by the shared transition matrix  $P$ . Since the two algorithms differ only in how RWs are duplicated, not in how active RWs move, their convergence behavior remains similar.

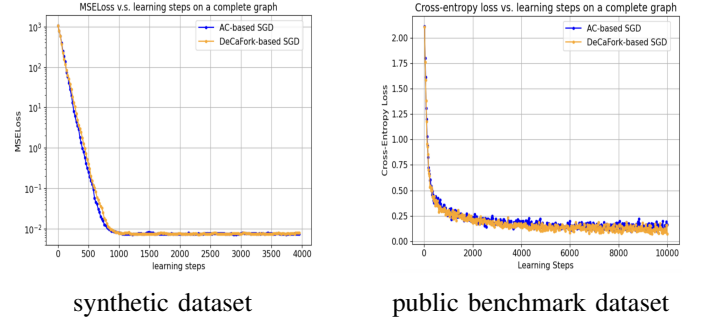


Fig. 3: Loss function v.s. learning steps on different graphs.

Finally, we evaluate the performance of the final models across different graphs using testing accuracies as the performance metric. Table I shows that the presence of a Pac-Man node degrades performance. This is because, in the Pac-Man setting, active RWs fail to access a representative sample of the entire dataset, leading to degraded performance. Further discussion is provided in Appendix G-E.

	Complete	Regular	Ring	Erdős-Rényi
A single SGD without the Pac-Man	0.9764	0.9719	0.9782	0.9751
AC	0.9613	0.9611	0.9617	0.9596
DeCaFork	0.9620	0.9606	0.9596	0.9621

TABLE I: Testing accuracies on different graphs under non-*i.i.d.* partitioning.

## VI. CONCLUSION

In this work, we investigated the robustness of RW-based SGD under the Pac-Man attack. This stealthy and highly disruptive threat can gradually degrade the decentralized operations without triggering any detectable failure signals. To address this challenge, we proposed the AC algorithm, a fully decentralized duplication mechanism based solely on local visitation intervals, and provided rigorous analysis. Extensive experiments on synthetic and real-world datasets validate our findings.

<sup>7</sup>In Fig. 3(a), the results are presented on a logarithmic scale.

## REFERENCES

- [1] L. Lovász, “Random walks on graphs,” *Combinatorics, Paul Erdős is eighty*, vol. 2, pp. 1–46, 1993.
- [2] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*. American Mathematical Society, second ed., 2017.
- [3] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, “A unified theory of decentralized SGD with changing topology and local updates,” in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 5381–5393, 2020.
- [4] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web,” technical report, Stanford InfoLab, 1999.
- [5] F. Fous, A. Pirotte, J. Renders, and M. Saerens, “Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation,” *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 355–368, 2007.
- [6] Y. Liu, S. Ji, and P. Mittal, “Smartwalk: Enhancing social network security via adaptive random walks,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 492–503, 2016.
- [7] L. Backstrom and J. Leskovec, “Supervised random walks: predicting and recommending links in social networks,” in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 635–644, 2011.
- [8] C. Zhang, D. Song, C. Huang, A. Swami, and N. Chawla, “Heterogeneous graph neural network,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 793–803, 2019.
- [9] B. Johansson, M. Rabi, and M. Johansson, “A simple peer-to-peer algorithm for distributed optimization in sensor networks,” in *46th IEEE Conference on Decision and Control*, pp. 4705–4710, 2007.
- [10] G. Ayache, V. Dassari, and S. E. Rouayheb, “Walk for learning: A random walk approach for federated learning from heterogeneous data,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 929–940, 2023.
- [11] Y. Lai, M. Waniek, L. Li, J. Wu, Y. Zhu, T. P. Michalak, T. Rahwan, and K. Zhou, “Coupled-space attacks against random-walk-based anomaly detection,” *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9315–9329, 2024.
- [12] X. Zhang, H. Xie, P. Yi, and J. Lui, “Enhancing sybil detection via social-activity networks: A random walk approach,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1213–1227, 2023.
- [13] M. Egger, R. Bitar, G. Ayache, A. Wachter-Zeh, and S. Rouayheb, “Self-duplicating random walks for resilient decentralized learning on graphs,” arXiv: 2407.11762, 2024.
- [14] M. Egger, R. Bitar, G. Ayache, A. Wachter-Zeh, and S. E. Rouayheb, “Self-regulating random walks for resilient decentralized learning on graphs,” *arXiv preprint arXiv:2407.11762*, 2024. Revised February 10, 2025.
- [15] T. Sun, Y. Sun, and W. Yin, “On markov chain gradient descent,” *Advances in neural information processing systems*, vol. 31, 2018.
- [16] X. Mao, K. Yuan, Y. Hu, Y. Gu, A. H. Sayed, and W. Yin, “Walkman: A communication-efficient random-walk algorithm for decentralized optimization,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2513–2528, 2020.
- [17] F. B. Schneider, “Implementing fault-tolerant services using the state machine approach: A tutorial,” *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, pp. 299–319, 1990.
- [18] L. Lamport, “The part-time parliament,” *ACM Transactions on Computer Systems (TOCS)*, vol. 16, no. 2, pp. 133–169, 1998.
- [19] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.
- [20] E. N. Elnozahy, L. Alvisi, Y. Wang, and D. B. Johnson, “A survey of rollback-recovery protocols in message-passing systems,” *ACM Computing Surveys (CSUR)*, vol. 34, no. 3, pp. 375–408, 2002.
- [21] T. D. Chandra and S. Toueg, “Unreliable failure detectors for reliable distributed systems,” *Journal of the ACM (JACM)*, vol. 43, no. 2, pp. 225–267, 1996.
- [22] M. Even, “Stochastic gradient descent under markovian sampling schemes,” in *Proceedings of the 40th International Conference on Machine Learning*, 2023.
- [23] B. Johansson, M. Rabi, and M. Johansson, “A randomized incremental subgradient method for distributed optimization in networked systems,” *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2010.
- [24] T. Sun, Y. Sun, and W. Yin, “On markov chain gradient descent,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 9918–9927, 2018.
- [25] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141 – 142, 2012.
- [26] R. Durrett, *Probability: Theory and Examples*. Thomson, Brooks Cole, 2019.
- [27] R. D. Foley and D. R. McDonald, “Yaglom limits can depend on the starting state,” *Journal of Applied Probability*, vol. 54, no. 3, pp. 726–734, 2017.
- [28] P. Collet, S. Martínez, and J. S. Martín, *Quasi-Stationary Distributions: Markov Chains, Diffusions and Dynamical Systems*. Probability and Its Applications, Springer, 2012.
- [29] P. Billingsley, *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons Inc., 3rd ed., 1995.
- [30] J. N. Darroch and E. Seneta, “On quasi-stationary distributions in absorbing discrete-time finite markov chains,” *Journal of Applied Probability*, vol. 2, no. 1, pp. 88–100, 1965.
- [31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [32] Z. Liu, S. Rouayheb, and M. Dwyer, “The entrapment problem in random walk decentralized learning,” in *IEEE ISIT*, 2024.
- [33] S. Hoory, N. Linial, and A. Wigderson, “Expander graphs and their applications,” *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.
- [34] T. H. Hsu, H. Qi, and M. Brown, “Measuring the effects of non-identical data distribution for federated visual classification,” 2019. arXiv:1909.06335.

APPENDIX A  
PRELIMINARIES: NOTATIONS, DEFINITIONS, AND  
ASSUMPTIONS

In this Appendix, we introduce the notation, definitions, and assumptions used in the system model and its transition dynamics.

**Definition 2** (Communication topologies and RWs). A communication topology is defined as a finite directed graph  $\mathcal{G} \triangleq (\mathcal{V}, E)$  with the set of nodes  $\mathcal{V} = [N]$  and the set of edges  $E \subseteq \binom{\mathcal{V}}{2}$ . Each RW  $X_j : \Omega \rightarrow \mathcal{V}^{\mathbb{Z}^+}$  on this graph is assumed to be *i.i.d.* and can be defined by the common transition probability matrix  $P : \mathcal{V} \rightarrow \mathcal{M}(\mathcal{V})$ , where the probability of transition from node  $u$  to node  $v$  in one time step at time  $t \in \mathbb{Z}_+$ , is

$$P_{uv} \triangleq \Pr(X_j(t+1) = v \mid X_j(t) = u).$$

We will call the RW on graph  $\mathcal{G}$  aperiodic if transition probability matrix  $P$  is aperiodic.

*Remark 1.* Without loss of generality, we assume that  $P_{uv} > 0$  for all nodes  $v$  connected to node  $u$  in graph  $\mathcal{G}$ . Therefore, transition matrix  $P$  is irreducible iff graph  $\mathcal{G}$  is connected.

**Definition 3** (Timing conventions). We define the timing conventions of the system as follows:

- (1) At the beginning of time slot  $t$ , let  $\mathcal{Z}_t$  denote the set of indices of active RWs. Each active RW  $j \in \mathcal{Z}_t$  is associated with a birth time  $\theta_j \geq 0$  and an initial location  $u_j \triangleq X_j(\theta_j) \in \mathcal{V}$ . Specifically, let  $Z_t = |\mathcal{Z}_t|$ .
- (2) At the end of time slot  $t$ , each active RW  $j \in \mathcal{Z}_t$  moves from its current location to a randomly selected neighbor in the next time slot. We denote by  $X_j(t+1)$  the location of RW  $j$  after this movement at time  $t+1$ .
- (3) Upon arrival at location  $X_j(t+1)$  at time  $t+1$ , termination operation at node 1 and creation operations at other nodes is performed (if applicable).

Under the model assumption, the Pac-Man node is fixed at location 1. We next present a formal definition of the system including the Pac-Man node.

**Definition 4.** Consider a communication topology  $\mathcal{G} = (\mathcal{V}, E)$ , as defined in Definition 2, where node 1 acts as the Pac-Man, such that if a RW visits node 1, it is sent to a death node  $0^8$  with a termination probability  $\zeta \in (0, 1]$ . That is, we augment graph  $\mathcal{G}$  to  $\mathcal{G}' = (\mathcal{V}', E')$  where  $\mathcal{V}' \triangleq \mathcal{V} \cup \{0\}$  and  $E' \triangleq E \cup \{(1, 0)\}$ . In the presence of this Pac-Man, the original RW transition probability matrix  $P$  now changes to  $P'$ , where for each state  $u, v \in \mathcal{V}'$

$$P'_{uv} \triangleq \begin{cases} P_{uv}, & u \in \mathcal{B}, v \in \mathcal{V}, \\ (1 - \zeta)P_{uv}, & u = 1, v \in \mathcal{V}, \\ \zeta, & u = 1, v = 0, \\ 1, & u = 0, v = 0 \\ 0, & u = 0, v \in \mathcal{V}. \end{cases} \quad (7)$$

<sup>8</sup>The death node is a virtual node used only for a clearer presentation.

Based on Definition 4, the transition matrix  $P'$  can be characterized by two cases:  $\zeta = 1$  and  $0 < \zeta < 1$ . When  $\zeta = 1$ , the Pac-Man node becomes an absorbing state, whereas when  $0 < \zeta < 1$ , the Pac-Man node is non-absorbing. In particular,

- (1) When  $\zeta = 1$ , the Pac-Man node eats every incoming RW, and the local data at the Pac-Man node cannot be utilized, making it an absorbing state. From Definition 4, the death node 0 is also absorbing. Therefore, we merge the Pac-Man node and the death node into a single absorbing one, which we continue to denote as node 1. The corresponding transition matrix  $P'$  becomes

$$P' = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{bmatrix} \triangleq \begin{bmatrix} 1 & \mathbf{0}_{1 \times (N-1)} \\ R^{(1)} & Q^{(1)} \end{bmatrix}. \quad (8)$$

We observe that  $Q^{(1)} \in [0, 1]^{\mathcal{B} \times \mathcal{B}}$  is sub-stochastic matrix where  $Q_{uv}^{(1)} = P_{uv}$  for each  $u, v \in \mathcal{B}$ , and  $R^{(1)} \in [0, 1]^{\mathcal{B} \times 1}$  column vector where  $R_u^{(1)} = P_{u1}$  for each benign node  $u \in \mathcal{B}$ .

- (2) When  $0 < \zeta < 1$ , the local data at the Pac-Man node cannot be reliably utilized, because any RW visiting it is terminated with probability strictly less than 1. In this case, only the death node 0 is absorbing. For clarity, we denote the location of the death node as 0. According to Definition 4 and denoting  $\bar{\zeta} \triangleq 1 - \zeta$ , the corresponding transition matrix  $P'$  becomes

$$P' = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \zeta & \bar{\zeta}P_{11} & \cdots & \bar{\zeta}P_{1N} \\ 0 & P_{21} & \cdots & P_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & P_{N1} & \cdots & P_{NN} \end{bmatrix} \triangleq \begin{bmatrix} 1 & \mathbf{0}_{1 \times N} \\ R^{(\zeta)} & Q^{(\zeta)} \end{bmatrix}. \quad (9)$$

We observe that

$$R^{(\zeta)} \triangleq \begin{bmatrix} \zeta \\ \mathbf{0}_{(N-1) \times 1} \end{bmatrix}, \quad Q^{(\zeta)} \triangleq \begin{bmatrix} \bar{\zeta}P_{11} & \bar{\zeta}S \\ R^{(1)} & Q^{(1)} \end{bmatrix}.$$

The row vector  $S \in [0, 1]^{1 \times \mathcal{B}}$  such that  $S_{1v} = P_{1v}$  for each benign node  $v \in \mathcal{B}$ .

*Remark 2.* If  $\zeta = 1$ , the analysis extends straightforwardly to the setting with multiple Pac-Man nodes. In this case, all Pac-Man nodes can be treated as a single “super node”, which acts as an absorbing state of the system. The resulting analysis is essentially identical to that of the single Pac-Man node case. If  $\zeta < 1$ , the analysis becomes more involved, since the transition probability to the absorbing state (i.e., the death node 0) depends on the identity of the Pac-Man node through (a) their corresponding termination probabilities and (b) their connectivity to the benign nodes. Nevertheless, the analysis can still be carried out within the theoretical framework proposed for the single Pac-Man node setting, although the analysis would be more tedious for a larger number of Pac-Man nodes.

**Definition 5** (Robustly Connected Graph). A graph  $\mathcal{G}$  is robustly connected if



- (i) every pair of benign nodes in  $\mathcal{B}$  is connected by a path that avoids the Pac-Man node, and
- (ii) the Markov chain corresponding to each active RW is aperiodic.

*Remark 3.* According to Definition 5, a robustly connected graph cannot be partitioned into two disjoint components by Pac-Man. In addition, in a robustly connected graph, the Markov chain corresponding to each active RW is irreducible and aperiodic.

**Definition 6.** For  $\zeta \in (0, 1]$ , let  $Q^{(\zeta)}$  be the matrix defined in (8) and (9). Let  $\alpha^{(\zeta)}$  denote the (unique) maximum eigenvalue of  $Q^{(\zeta)}$ <sup>9</sup>, and let  $\nu^{(\zeta)}$  denote the associated left normalized positive eigenvectors with unit sum.

**Assumption 1.** Each RW has an i.i.d. evolution on this graph with transition probability matrix  $P$ , conditioned on their initial locations.

**Assumption 2.** The graph  $\mathcal{G}$  defined in Definition 2 is a robustly connected graph.

Next, we adopt the common assumptions used in standard distributed optimization problems, as follows.

**Assumption 3.** Each local function  $f_u(\mathbf{x})$  in (1) with  $u \in \mathcal{V}$  is  $\mu$ -strongly convex and  $L$ -smooth.

**Assumption 4.** Bounded norm of the local gradient at the global optimum  $\mathbf{x}^*$ , i.e.  $\sup_{u \in \mathcal{V}} \|\nabla f_u(\mathbf{x}^*)\|^2 \leq \sigma^2$ , where  $\mathbf{x}^*$  is the minimizer of (1).

In the remainder of this article, we assume that Assumptions 1 – 4 hold.

## APPENDIX B PROOF OF THEOREM 1

**Definition 7.** Let  $\Pr_u$  denote the probability measure under which the RW  $j$  starts at node  $u$ , i.e.,  $X_{0,j} = u$ . Given a distribution  $\nu$  over the node set  $[N]$ , we define the mixed law  $\Pr_\nu$

$$\Pr_\nu = \int_{u \in [N]} \Pr_u d\nu(u),$$

which corresponds to initializing RW  $j$  according to  $\nu$ .

**Definition 8.** Consider the number of active RWs  $Z_t$  at time  $t \in \mathbb{Z}_+$ . The natural filtration for the random sequence  $Z \triangleq (Z_t : t \in \mathbb{Z}_+)$  is denoted by  $\mathcal{F}_\bullet \triangleq (\mathcal{F}_t : t \in \mathbb{Z}_+)$  where  $\mathcal{F}_t \triangleq \sigma(Z_s, s \leq t)$ .

Consider the graph  $\mathcal{G}'$  defined in Definition 4. Since the original graph  $\mathcal{G}$  is connected and finite, then the Pac-Man is *reachable* from any other node  $u \in [N]$ , i.e., there exists  $n_u \in \mathbb{N}$  such that  $(P'_{u0})^{n_u} > 0$ .

<sup>9</sup>We assume a unique maximum eigenvalue for simplicity. If  $Q^{(\zeta)}$  has multiple dominant eigenvalues, the arguments can be extended using a standard Jordan decomposition.

**Definition 9.** We define the smallest number of steps to reach the Pac-Man from node  $u \in [N]$  as

$$d_u \triangleq \inf \{n \in \mathbb{N} : (P'_{u0})^n > 0\}.$$

We define the maximum of the minimum time steps to reach the Pac-Man from node  $u \in [N]$  as

$$d \triangleq \max_{u \in [N]} d_u. \quad (10)$$

Accordingly, we define the smallest probability of reaching the Pac-Man within  $d$  steps from node  $u \in [N]$ , as

$$c \triangleq \min_{u \in [N]} (P'_{u0})^{d_u}.$$

Since the Pac-Man is reachable,  $n_u$  is finite for each  $u \in [N]$ , then  $d_u \leq n_u$  and hence is finite. By Definition 4,  $P'_{u0} > 0$  for each  $u \in [N]$ , and then  $c$  is positive. Thus,  $d$  and  $c$  are well-defined.

**Definition 10.** Consider a finite connected graph  $\mathcal{G}$  with  $d$  defined in (10). During a fixed finite and half-open time interval  $T \subseteq \mathbb{R}_+$ , we denote the number of random walks that hit PacMan by  $D_T$  and the number of RWs generated by  $G_T$ .

**Lemma 1.** Consider  $Z_t$  independent, aperiodic, active RWs at time  $t$ , each following the identical law over a finite connected graph  $\mathcal{G}$  with  $d$  defined in (10). Then,

$$\mathbb{E}[Z_{t+d} - Z_t \mid \mathcal{F}_t] \leq -c\zeta Z_t + (N-1)d. \quad (11)$$

*Proof:* From the definition of  $D$  and  $G$  from Definition 10, we can write the difference in the number of active RWs at time  $t+d$  and  $t$  as

$$Z_{t+d} = Z_t - D_{(t,t+d]} + G_{(t,t+d]}. \quad (12)$$

We note that the graph  $\mathcal{G}$  has  $(N-1)$  benign nodes, and at most one RW can be generated at each node at each time  $t$ . Thus, we have

$$\mathbb{E}[G_{(t,t+d]} \mid \mathcal{F}_t] \leq \sum_{s=0}^{d-1} (Z_{t+s} \wedge (N-1)) \leq (N-1)d. \quad (13)$$

At time  $t$ , there are  $Z_t$  active RWs. From Definition 9, for any RW  $X_j$  with transition probability matrix  $P$ , we have

$$\Pr_u \left( \bigcup_{n=1}^d \{X_{n,j} = 1\} \right) \geq \Pr_u (X_{d_u,j} = 1) \geq c,$$

where  $\Pr_u$  is defined in Definition 7. That is,  $c$  is the uniform lower bound on the probability of ending up at the Pac-Man *within*  $d$  steps, over all possible initial positions. It follows that the number of deaths for RWs is lower bounded by the number of active RWs at time  $t$  hitting Pac-Man (ignoring the RWs generated during this interval and hitting Pac-Man), and hence

$$\mathbb{E}[D_{(t,t+d]} \mid \mathcal{F}_t] \geq c\zeta Z_t. \quad (14)$$

Taking conditional expectation of (12) given history  $\mathcal{F}_t$ , substituting the upper bound on the conditional mean number of births (13) and the lower bound on the conditional mean number of deaths (14), we obtain the result. ■



**Corollary 1.** Consider independent aperiodic RWs on a finite connected graph  $\mathcal{G}$  with identical probability laws and  $d$  defined in (10). For any  $\epsilon > 0$ , there exists positive constants  $b, B$ , such that  $B \geq b$  and the random sequence  $Z$  satisfies the following conditions.

- (a) If  $Z_t \leq B$ , then  $\mathbb{E}[Z_{t+d} | \mathcal{F}_t] \leq b$ .
- (b) If  $Z_t > B$ , then  $\mathbb{E}[Z_{t+d} - Z_t | \mathcal{F}_t] < -\epsilon$ .

*Proof:* Let  $\epsilon > 0$  and  $N$  be the number of nodes in  $\mathcal{G}$ . We define  $B \triangleq \frac{1}{c\zeta}((N-1)d + \epsilon)$  and  $b \triangleq (1 - c\zeta)B + (N-1)d$  for  $c$  defined in Definition 9. It follows that  $b, B$  are positive and  $b \leq B$ .

(a) Let  $Z_t \leq B$ . It follows from (11), that

$$\mathbb{E}[Z_{t+d} | \mathcal{F}_t] \leq (1 - c\zeta)B + (N-1)d = b.$$

(b) Let  $Z_t > B$ . It follows from (11) and definition of  $B$ , that

$$\mathbb{E}[Z_{t+d} - Z_t | \mathcal{F}_t] \leq -c\zeta Z_t + (N-1)d \leq -\epsilon.$$

**Definition 11** (Supermartingale). Consider independent aperiodic RWs on a finite connected graph  $\mathcal{G}$  with identical probability laws and  $d$  defined in (10),  $\epsilon > 0$ , and positive constants  $b, B$  defined in Corollary 1. We define a Lyapunov function  $V : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  for each  $z \in \mathbb{R}_+$

$$V(z) \triangleq z \mathbb{1}_{\{z > B\}} + B \mathbb{1}_{\{z \leq B\}}. \quad (15)$$

For  $t_0 \geq 0$  and  $k \in \mathbb{Z}_+$ , we define periodic samples of number of active RWs and its natural filtration at time  $t_0 + dk$  as

$$M_k \triangleq V(Z_{t_0+dk}), \quad \mathcal{H}_k \triangleq \mathcal{F}_{t_0+dk}.$$

We define a random sequence  $M \triangleq (M_k : k \in \mathbb{Z}_+)$  and filtration  $\mathcal{H}_\bullet \triangleq (\mathcal{H}_k : k \in \mathbb{Z}_+)$ .

*Remark 4.* By definition, we have  $V(z) - z \leq 0$  for all  $z \in \mathbb{R}_+$ . It follows that

$$M_{k+1} - M_k \leq Z_{t_0+d(k+1)} - M_k. \quad (16)$$

**Lemma 2.** Sequence  $M$  is a supermartingale adapted to filtration  $\mathcal{H}_\bullet$ .

*Proof:* We first observe that  $Z_t$  is  $\mathcal{F}_t$  measurable, and hence  $M_k$  is a  $\mathcal{H}_k$  measurable by definition. For each  $k \in \mathbb{Z}_+$ , we can define a  $\mathcal{F}_k$  measurable event  $\mathcal{A}_k \triangleq \{M_k \leq B\}$ . In terms of these events, we can write the conditional mean as

$$\mathbb{E}[(M_{k+1} - M_k) | \mathcal{F}_k] = \mathbb{E}[(M_{k+1} - M_k)(\mathbb{1}_{\mathcal{A}_k} + \mathbb{1}_{\mathcal{A}_k^c}) | \mathcal{F}_k].$$

From Corollary 1, definition (15), and (16), we obtain that the conditional  $\mathbb{E}[(M_{k+1} - M_k) \mathbb{1}_{\mathcal{A}_k} | \mathcal{H}_k]$  is upper bounded by

$$\mathbb{E}[Z_{t_0+d(k+1)} | \mathcal{H}_k] \mathbb{1}_{\mathcal{A}_k} - B \mathbb{1}_{\mathcal{A}_k} \leq 0.$$

From definition (15), we have  $\mathbb{1}_{\mathcal{A}_k^c} M_k = Z_{t_0+dk} \mathbb{1}_{\mathcal{A}_k^c}$ . From Corollary 1, we obtain that the conditional mean  $\mathbb{E}[(M_{k+1} - M_k) \mathbb{1}_{\mathcal{A}_k^c} | \mathcal{H}_k]$  is upper bounded by

$$\mathbb{E}[(Z_{t_0+d(k+1)} - Z_{t_0+dk}) | \mathcal{H}_k] \mathbb{1}_{\mathcal{A}_k^c} \leq 0.$$

Combining the two results, we get the result. ■

#### A. Proof of Theorem 1

We define a stopping time  $\tau_0 \triangleq \inf \{t \in \mathbb{Z}_+ : Z_t = 0\}$ . If  $\tau_0 < \infty$ , then  $Z_t = 0$  for all  $t \geq \tau_0$ , which implies  $\limsup_{t \rightarrow \infty} Z_t < \infty$ .

Therefore, without loss of generality, we consider the case when  $Z_t > 0$  for any finite time  $t$ . From the definition of sequence  $M$  and filtration  $\mathcal{H}_\bullet$  in Definition 11, Lemma 2, and positivity of  $Z$ , we observe that  $M$  is a positive supermartingale adapted to filtration  $\mathcal{H}_\bullet$ . By the Doob's supermartingale convergence Theorem [26], supermartingale  $M$  converges to a limit  $M_\infty$  almost surely, i.e.

$$\lim_{k \rightarrow \infty} M_k = M_\infty < \infty, \text{ a.s.}$$

From the definition of supermartingale  $M$  in Definition 11, it follows that for any  $t_0 \geq 0$ ,

$$\limsup_{k \rightarrow \infty} Z_{t_0+dk} \leq \max\{B, M_\infty\} < \infty.$$

Since the choice of  $t_0 \in \mathbb{Z}_+$  was arbitrary, we have

$$\limsup_{t \rightarrow \infty} Z_t < \infty, \text{ a.s.}$$

#### APPENDIX C

##### SOFT PHASE TRANSITION IN COMPLETE GRAPHS

##### A. Definitions and the Weak Version of AC Algorithm

**Definition 12** (Almost fully connected graph  $\mathcal{G}$ ). Let the killing and forking probabilities  $\zeta = q = 1$ . Consider a finite graph  $\mathcal{G}'$  from Definition 4 with a single malicious node 1, and edges

$$E = \binom{[N]}{2} \cup \{(1, d)\} \setminus \{(1, n) : n \geq 1\}.$$

We call this an *almost fully connected graph*, and a RW on this graph with uniform transition probability<sup>10</sup> is called *uniform RW* on the almost fully connected graph. We assume that the initial location of each initial uniform RWs on this graph is sampled *i.i.d.* uniform from all nodes  $\mathcal{V}$ , and the duplication threshold at each benign node is identically  $A_u = A \geq 1$ .

**Definition 13** (Lifetime). Consider a RW  $j \in \mathcal{Z}_t$  with the birth time  $\theta_j$  and the initial location  $u_j \in \mathcal{B}$ . We define its lifetime as the first time it visits the Pac-Man node, given by

$$K^{(j)} = \inf \{t \geq \theta_j : X_{t,j} = 0, X_{\theta_j,j} = u_j\}. \quad (17)$$

The distribution of  $K^{(j)}$  is independent of the initial location  $u_j$  because the graph structure is complete and the transition matrix is uniform.

**Definition 14** (Age of Visiting). Let  $L_t^{(u,j)}$ , for  $u \in \mathcal{B}$  and  $t \in [\theta_j, K^{(j)}]$ , denote the most recent time up to  $t$  at which RW  $j \in \mathcal{Z}_t$  visited node  $u$ :

$$L_t^{(u,j)} = \max \{t' \in [\theta_j, t] : X_{t',j} = u, X_{\theta_j,j} = u_j\}. \quad (18)$$

We define the *age of visiting* (AoV) of node  $u$  with respect to RW  $j \in \mathcal{Z}_t$  at time  $t$ , denoted  $H_t^{(u,j)}$ , as follows:

<sup>10</sup>i.e.  $P_{uv} = \frac{1}{N}$  for all  $u \in \mathcal{B}$  and  $v \in \mathcal{V}$ .

(1) if RW  $j$  visits node  $u$  at time  $t$  for the first time, then

$$H_t^{(u,j)} = 0; \quad (19)$$

(2) if node  $u$  was visited by RW  $j$  at least once before time  $t$ , then

$$H_t^{(u,j)} = t - L_{t-1}^{(u,j)}. \quad (20)$$

Both  $L_t^{(u,j)}$  and  $H_t^{(u,j)}$  are defined only after the active RW  $j \in \mathcal{Z}_t$  has visited node  $u$ . If RW  $j$  has not yet visited node  $u$ , or has already been terminated, then  $L_t^{(u,j)}$  and  $H_t^{(u,j)}$  are assigned a null value.

We specify the AoV of a newly generated RW  $j$  at node  $u_j$  at time  $t$ , following Definition 3 and Definition 14. Suppose that RW  $j$  is generated at node  $u_j$  at time  $t$ , we have  $j \in \mathcal{Z}_{t+1}$ , and it begins its movement in the next time slot, i.e., at time  $t+1$ . As the RW  $j$  is initialized at node  $u_j$  at time  $t$ , we define the AoV at that moment as 0, i.e.,  $H_t^{(u_j,j)} = 0$ .

Instead of analyzing the full AC algorithm, we focus on a simplified variant known as *weak version of AC (W-AC)* algorithm, which is outlined below.

---

**Algorithm 2** Weak Version of AC (W-AC)

---

- 1: **Input:** The graph  $\mathcal{G}$ , the threshold  $A \geq 0$ , the forking probability  $q = 1$ , and the indices of initial RWs  $\mathcal{Z}_0$ .
  - 2: **for**  $t \geq 0$  **do**
  - 3:   **for**  $j \in \mathcal{Z}_t$  **do**
  - 4:     If  $X_{j,t} = u \in \mathcal{B}$  and  $H_t^{(u,j)} \geq A$ , RW  $j$  forks a new RW, denoted by  $j'$ , as an identical copy of it. Set  $H_t^{(u,j')} = 0$  and  $\mathcal{Z}_{t+1} = \mathcal{Z}_t \cup \{j'\}$ ; otherwise, if  $H_t^{(u,j)} < A$ , RW  $j$  is moved to one of the neighbors of node  $u$  uniformly.
  - 5:   **end for**
  - 6: **end for**
- 

*B. Proof of Proposition 1*

**Definition 15** (Duplication Events). Consider any  $t \geq 0$  and  $j \in \mathcal{Z}_t$ . Let  $B_t^{(j)} \in \{0, 1\}$  be the indicator for a duplication occurring by RW  $j \in \mathcal{Z}_t$  at time  $t$ .

From the W-AC algorithm defined in Algorithm 2, we have the expression for  $B_t^{(j)}$ :

$$B_t^{(j)} = \mathbb{1}_{\{X_{t,j} \neq 0\}} \mathbb{1}_{\{H_t^{(X_{t,j},j)} \geq A\}}. \quad (21)$$

**Definition 16.** Define  $\Delta_t(A, z)$  as the expected number of newly duplicated RWs at time  $t$ , given that the current number of active RWs is  $Z_t = z$ . That is,

$$\Delta_t(A, z) \triangleq \mathbb{E} \left[ \sum_{j \in \mathcal{Z}_t} B_t^{(j)} \middle| Z_t = z \right]. \quad (22)$$

1) *Expression for  $\Delta_t(A, z)$ :* In this subsection, we first derive a closed-form expression for  $\Delta_t(A, z)$ . For simplicity in analysis, conditional on the event  $Z_t = z$ , we re-label the active RWs in  $\mathcal{Z}_t$  as  $[z]$  without loss of generality. That is, we assign each  $j \in \mathcal{Z}_t$  a new index in  $[z]$  via a one-to-one mapping. Accordingly, the associated variables  $L_t^{(u,j)}$  and  $H_t^{(u,j)}$  are re-indexed under this mapping. With a slight abuse of notation, we continue to denote the re-indexed random walk as  $j$ , where  $1 \leq j \leq z$ .

According to Definition 3, each active RW  $j \in [z]$  was generated *strictly* before time  $t$ . Consider any  $j \in [z]$  and denote  $X_{t,j} = u$ . A new RW is duplicated from RW  $j$  if the following 2 conditions are satisfied: (i)  $u \in \mathcal{B}$  and (ii) the associated age variable satisfies  $H_t^{(u,j)} \geq A$ . We denote every initial condition of the active RWs in  $[z]$  as:

$$I_{t,z} = \{(\theta_j, u_j) : \theta_j \in [0, t), u_j \in \mathcal{B}, j \in [z]\}. \quad (23)$$

By the law of total expectation, (22) can be re-written as

$$\begin{aligned} \Delta_t(A, z) &= \sum_{I_{t,z}} \mathbb{E} \left[ \sum_{j \in [z]} B_t^{(j)} \middle| Z_t = z, I_{t,z} \right] \\ &\quad \times \Pr(I_{t,z} | Z_t = z). \end{aligned} \quad (24)$$

We now focus on analyzing each conditional expectation term  $\mathbb{E} [B_t^{(j)} | Z_t = z, I_{t,z}]$  with  $j \in [z]$ . For each  $j \in [z]$ , let the lifetime  $K^{(j)}$  be defined in Definition 13. When  $t \leq K^{(j)}$ , we define the first hitting time of node  $u \in \mathcal{B}$  by the RW  $j \in [z]$ , starting from  $u_j \in \mathcal{B}$ , as:

$$K_u^{(j)} = \inf \{ \tau \in [\theta_j, t] : X_{\tau,j} = u, X_{\theta_j,j} = u_j \}. \quad (25)$$

For each  $j \in [z]$ , we have:

$$\begin{aligned} \mathbb{E} [B_t^{(j)} | Z_t = z, I_{t,z}] &= \mathbb{E} [B_t^{(j)} | 1 \leq j \leq z, I_{t,z}] \\ &= \sum_{u \in [N]} \Pr(X_{t,j} = u, H_t^{(u,j)} \geq A | 1 \leq j \leq z, I_t) \\ &= \sum_{u \in [N]} \Pr(X_{t,j} = u, K_u^{(j)} \leq t, H_t^{(u,j)} \geq A | 1 \leq j \leq z, I_t) \\ &\triangleq \sum_{u \in [N]} Q_{u,j}. \end{aligned} \quad (26)$$

By the joint distribution factorization,  $Q_{u,j}$  defined in (26) can be computed as follows:

$$\begin{aligned} Q_{u,j} &= \Pr(X_{t,j} = u | 1 \leq j \leq z, I_{t,z}) \\ &\quad \times \Pr(K_u^{(j)} \leq t, H_t^{(u,j)} \geq A | 1 \leq j \leq z, I_{t,z}, X_{t,j} = u) \\ &= \Pr(X_{t,j} = u | 1 \leq j \leq z, I_{t,z}) \\ &\quad \times \sum_{k=\theta_j}^t \Pr(K_u^{(j)} = k | 1 \leq j \leq z, I_{t,z}, X_{t,j} = u) \\ &\quad \times \Pr(H_t^{(u,j)} \geq A | 1 \leq j \leq z, I_{t,z}, X_{t,j} = u, K_u^{(j)} = k). \end{aligned} \quad (27)$$

Note that RW  $j$  was born at time  $\theta_j$ , and the killing probability  $\zeta = 1$ , it cannot visit the Pac-Man node during the interval  $(\theta_j, t]$ . Therefore:

$$Q_{u,j}^{(1)} \triangleq \Pr(X_{t,j} = u \mid 1 \leq j \leq z, I_{t,z}) \\ = \frac{\left(\frac{N-1}{N}\right)^{t-\theta_j-1} \frac{1}{N}}{\left(\frac{N-1}{N}\right)^{t-\theta_j}} = \frac{1}{N-1}. \quad (28)$$

Recall that  $k$  is the value of the first hitting time  $K_u^{(j)}$ . For each  $k \in [\theta_j, t]$ , the event  $K_u^{(j)} = k$  implies that RW  $j$  cannot visit node  $u$  before time  $k$ . Therefore:

$$Q_{u,j}^{(2,k)} \triangleq \Pr\left(K_u^{(j)} = k \mid X_{t,j} = u, 1 \leq j \leq z, I_{t,z}\right) \\ = \frac{\Pr\left(K_u^{(j)} = k, X_{t,j} = u \mid 1 \leq j \leq z, I_{t,z}\right)}{\Pr(X_{t,j} = u \mid 1 \leq j \leq z, I_{t,z})} \\ = \begin{cases} \mathbb{1}_{\{u_j=u\}} & k = \theta_j \\ \mathbb{1}_{\{u_j \neq u\}} \frac{1}{N-1} \left(\frac{N-2}{N-1}\right)^{k-\theta_j-1} & \theta_j < k < t \\ \mathbb{1}_{\{u_j \neq u\}} \left(\frac{N-2}{N-1}\right)^{t-\theta_j-1} & k = t. \end{cases} \quad (29)$$

Consider the condition  $H_t^{(u,j)} \geq A$ . It implies that RW  $j$  cannot visit node  $u$  during the interval  $[t-A, t-1]$ . Therefore, if  $t-A < k$ , then  $H_t^{(u,j)} \geq A$  and  $K_u^{(j)} = k$  are incompatible and cannot occur simultaneously. Therefore:

$$Q_{u,j}^{(3,k)} \triangleq \Pr\left(H_t^{(u,j)} \geq A \mid X_{t,j} = u, K_u^{(j)} = k, I_{t,z}\right) \\ = \frac{\Pr\left(H_t^{(u,j)} \geq A, K_u^{(j)} = k \mid X_{t,j} = u, I_{t,z}\right)}{\Pr\left(K_u^{(j)} = k \mid X_{t,j} = u, I_{t,z}\right)}.$$

We evaluate this quantity in three cases:

- (i) *Case 1:*  $k = \theta_j$ . Given that  $X_{t,j} = u$ , by definition of  $H_t^{(u,j)}$  in Definition 14, we have  $H_t^{(u,j)} = t - L_{t-1}^{(u,j)} \geq 1 > 0$ . Then:

$$Q_{u,j}^{(3,k)} = \mathbb{1}_{\{u=u_j\}} \mathbb{1}_{\{t-A \geq \theta_j\}} \\ \times \left( \mathbb{1}_{\{A \geq 1\}} \left(\frac{N-2}{N-1}\right)^{A-1} + \mathbb{1}_{\{A=0\}} \right). \quad (30)$$

- (ii) *Case 2:*  $\theta_j < k < t$ . Again, since  $X_{t,j} = u$ , we know  $H_t^{(u,j)} = t - L_{t-1}^{(u,j)} \geq 1 > 0$ . Then:

$$Q_{u,j}^{(3,k)} = \mathbb{1}_{\{u \neq u_j\}} \mathbb{1}_{\{t-A \geq k\}} \\ \times \left( \mathbb{1}_{\{A \geq 1\}} \left(\frac{N-2}{N-1}\right)^{A-1} + \mathbb{1}_{\{A=0\}} \right). \quad (31)$$

- (iii) *Case 3:* If  $k = t$ , we have:

$$Q_{u,j}^{(3,k)} = \mathbb{1}_{\{u \neq u_j\}} \mathbb{1}_{\{A=0\}}. \quad (32)$$

Substituting (28), (29), (30), (31), and (32) into (27), substituting (27) into (26), and substituting (26) into (24), we obtain the expression for  $\Delta_t(A, z)$  directly as follows:

$$\Delta_t(A, z) = \sum_{I_{t,z}} \Pr(I_{t,z} \mid Z_t = z) \\ \times \sum_{j \in [z]} \left( \sum_{u \in [N]} Q_{u,j}^{(1)} \sum_{k=\theta_j}^t Q_{u,j}^{(2,k)} Q_{u,j}^{(3,k)} \right). \quad (33)$$

2) *Proof of Part (a):* Since each RW hits the Pac-Man at node 1 with probability  $\frac{1}{N}$  at every step, and is killed with probability  $\zeta = 1$  upon hitting it, then, given the number of active RWs  $Z_t = z$ , the expected number of RWs that die at time  $t$  is

$$\frac{z}{N}. \quad (34)$$

From (22) and (34), given the number of active RWs  $Z_t = z$ , we derive

$$\mathbb{E}[Z_{t+1} \mid Z_t = z] = z \left( 1 - \frac{1}{N} + \frac{\Delta_t(A, z)}{z} \right) \\ \triangleq zm_t(A, z),$$

where we define

$$m_t(A, z) \triangleq 1 - \frac{1}{N} + \frac{\Delta_t(A, z)}{z}. \quad (35)$$

**Lemma 3.** *If there exists a positive integer  $\bar{\alpha}$ , such that when  $A \geq \bar{\alpha}$ , the following inequality holds*

$$\sup_{t \geq 0} \sup_{z \in \mathbb{Z}_+} \frac{\Delta_t(A, z)}{z} < \frac{1}{N}. \quad (36)$$

Then, we have

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) = 1$$

for all  $z_0 \geq 1$ .

*Proof:* Let  $A \geq \bar{\alpha}$ . By assumption (36),

$$\frac{\Delta_t(A, z)}{z} < \frac{1}{N}$$

holds for all  $z \in \mathbb{Z}_+$  and  $t \geq 0$ . Then we have

$$m_t(A, z) = 1 - \frac{1}{N} + \frac{\Delta_t(A, z)}{z} < 1$$

holds for all  $z \in \mathbb{Z}_+$ , and  $t \geq 0$ . We define

$$\epsilon \triangleq 1 - \sup_{t \geq 0, z \in \mathbb{Z}_+} m_t(A, z) > 0.$$

Therefore,

$$\mathbb{E}[Z_{t+1} \mid Z_t = z] = zm_t(A, z) \leq (1 - \epsilon)z.$$

This implies

$$\mathbb{E}[Z_{t+1} \mid Z_t] \leq (1 - \epsilon)Z_t < Z_t. \quad (37)$$

It follows that  $\{Z_t\}_t$  is a supermartingale. Since  $Z_t \geq 0$  for all  $t$ , Doob's Super-Martingale Convergence Theorem ensures that

$$Z_t \rightarrow Z_\ell, \text{ a.s.}$$

Hence,

$$\mathbb{E}[Z_t] \rightarrow \mathbb{E}[Z_\ell]. \quad (38)$$

We now prove that  $Z_\ell = 0$  almost surely by contradiction. If the statement is false, then  $\mathbb{E}[Z_\ell] > 0$ , taking the limit (38) into (37), we obtain

$$\mathbb{E}[Z_\ell] \leq (1 - \epsilon)\mathbb{E}[Z_\ell] < \mathbb{E}[Z_\ell],$$

which is a contradiction. Thus,  $Z_\infty = 0$  almost surely. This implies that all RWs are eventually absorbed by the Pac-Man, and hence

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) = 1$$

for all  $z_0$ . ■

In what follows, we prove the existence of  $\bar{\alpha}$ . To ensure that condition (36) holds, we start from  $Q_{j,u}^{(2,k)}$  defined in (29) and  $Q_{j,u}^{(3,k)}$  defined in (30), (31), (32). Without loss of generality, we consider  $A \geq 1$ . Specifically:

(a) In (29), since  $\frac{1}{N-1} < 1$  and  $A \geq 1$ , then

$$Q_{j,u}^{(2,k)} < \left(\frac{N-2}{N-1}\right)^{k-\theta_j-1}, \quad \theta_j \leq k \leq t. \quad (39)$$

(b) In (30), (31) and (32), since  $\frac{1}{N} \leq \frac{N-2}{N} < \frac{N-1}{N} < 1$  (when  $N \geq 3$ ) and  $A \geq 1$ , then

$$Q_{j,u}^{(3,k)} < \left(\frac{N-2}{N-1}\right)^{A-1}. \quad (40)$$

Substituting (39) and (40) into (33), we obtain:

$$\begin{aligned} \Delta_t(A, z) &< z \frac{N-1}{N-2} \left(\frac{N-2}{N-1}\right)^{A-1} \sum_{I_{t,z}} \Pr(I_{t,z} \mid Z_t = z) \\ &= z \frac{N-1}{N-2} \left(\frac{N-2}{N-1}\right)^{A-1}. \end{aligned} \quad (41)$$

To ensure the condition (36) holds for all  $t \geq 0$  and  $z \in \mathbb{Z}_+$ , it suffices to ensure

$$\left(\frac{N-2}{N-1}\right)^{A-1} < \frac{N-2}{(N-1)N}.$$

Since  $\left(\frac{N-2}{N-1}\right)^{A-1}$  is a decreasing function of  $A$ , then  $\left(\frac{N-2}{N-1}\right)^{A-1} < \frac{N-2}{N(N-1)}$  when  $A \geq 1 + \frac{\ln(\frac{N-2}{(N-1)N})}{\ln(\frac{N-2}{N-1})} \triangleq \bar{\alpha}$ .

3) *Proof of Part (b):* Now, we re-visit the recursion:

$$\mathbb{E}[Z_{t+1} \mid Z_t = z] = z m_t(A, z),$$

where  $m_t(A, z)$  is defined in (35):

$$m_t(A, z) \triangleq 1 - \frac{1}{N} + \frac{\Delta_t(A, z)}{z}.$$

**Lemma 4.** *If there exists a positive integer  $\underline{\alpha}$ , such that when  $A \leq \underline{\alpha}$ , the following inequality holds:*

$$\inf_{t \geq 0} \inf_{z \in \mathbb{Z}_+} \frac{\Delta_t(A, z)}{z} > \frac{1}{N}. \quad (42)$$

*Then, for any  $A \leq \underline{\alpha}$ , we have*

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) < 1$$

for all  $z_0$ .

*Proof:* Let  $A \leq \underline{\alpha}$ . By assumption (42),

$$\frac{\Delta_t(A, z)}{z} > \frac{1}{N}$$

holds for all  $z \in \mathbb{Z}_+$  and  $t \geq 0$ . Then we have

$$m_t(A, z) = 1 - \frac{1}{N} + \frac{\Delta_t(A, z)}{z} > 1$$

for all  $z \in \mathbb{Z}_+$  and  $t \geq 0$ . Define

$$\epsilon \triangleq \inf_{t \geq 0, z \in \mathbb{Z}_+} m_t(A, z) - 1 > 0.$$

Then, for all  $z \in \mathbb{Z}_+$ ,

$$\mathbb{E}[Z_{t+1} \mid Z_t = z] = z m_t(A, z) \geq (1 + \epsilon)z.$$

This implies

$$\mathbb{E}[Z_{t+1}] \geq (1 + \epsilon)\mathbb{E}[Z_t]. \quad (43)$$

Assume, for contradiction, that

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) = 1$$

for any  $z_0$ . Then,  $\lim_{\ell \rightarrow \infty} Z_\ell = 0$  almost surely, and hence  $\lim_{\ell \rightarrow \infty} \mathbb{E}[Z_\ell] = 0$ . However, by recursively applying (43), we obtain:

$$\lim_{\ell \rightarrow \infty} \mathbb{E}[Z_\ell] \geq \lim_{\ell \rightarrow \infty} (1 + \epsilon)^\ell z_0 \rightarrow \infty,$$

which contradicts to assumption. Therefore,

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) < 1$$

for all  $z_0 \in \mathbb{Z}_+$ . ■

In what follows, to prove the existence of  $\underline{\alpha}$ . We begin by examining the case where  $A = 0$ . Substituting  $A = 0$  into (33), we have:

$$\Delta_t(0, z) = \frac{N-1}{N}z. \quad (44)$$

Substituting (44) into (35), we observe that the resulting expression satisfies condition (42) for all  $t \geq 0$  and  $z \in \mathbb{Z}_+$ . This implies that  $A = 0$  is always a valid threshold under the given condition.

Next, we consider the case  $A = 1$ . From (28), (29), (30)~(32), and (33), we obtain:

$$\begin{aligned} \Delta_t(1, z) &= \sum_{I_{t,z}} \Pr(I_{t,z} \mid Z_t = z) \\ &\times \sum_{j \in [z]} \left( \sum_{u \in \mathcal{B}} \frac{1}{N} \left( 1 - \mathbb{1}_{\{u \neq u_j\}} \left( \frac{N-2}{N-1} \right)^{t-\theta_j-1} \right) \right). \end{aligned}$$

Since  $t - \theta_j > 1$ , i.e.,  $t - \theta_j - 1 \geq 0$ , and the equality  $t = \theta_j + 1$  can not hold for every  $j \in [z]$  and every  $I_{t,z}$ , then

$$\begin{aligned} \Delta_t(1, z) &= \sum_{I_{t,z}} \Pr(I_{t,z} \mid Z_t = z) \\ &\times \sum_{j \in [z]} \left( \frac{N-1}{N} - \frac{N-2}{N} \left( \frac{N-2}{N-1} \right)^{t-\theta_j-1} \right) \\ &> \frac{z}{N} \sum_{I_{t,z}} \Pr(I_{t,z} \mid Z_t = z) = \frac{z}{N}. \end{aligned} \quad (45)$$

Substituting (45) into (35), we see that condition (42) is satisfied for all  $t \geq 0$  and  $z \in \mathbb{Z}_+$ . This confirms that  $A = 1$  is always a valid threshold under the given condition.

For any  $z_0 \in \mathbb{Z}_+$ , we choose  $\underline{\alpha} = 1$ , then

$$\Pr(\exists t_0, \forall t \geq t_0, Z_t = 0 \mid Z_0 = z_0) < 1$$

for all  $A \leq \underline{\alpha}$ , i.e.,  $A \in \{0, 1\}$ .

#### APPENDIX D PROOF OF THEOREM 2

We treat a chain of RWs as a *single effective RW* and study its limiting behavior, if it exists. Consider an infinite chain of RWs  $\{j_s\}_s$ . Let  $\mathcal{A}$  be the set of absorbing states, i.e.,  $\mathcal{A} = \{1, w\}$  if  $\zeta = 1$  and  $\mathcal{A} = \{w\}$  if  $\zeta \in (0, 1)$ . Let  $u \in \mathcal{B}$  denote the initial location of RW  $j_0$ . Let  $\nu$  be a probability measure on  $\mathcal{B}$ , and  $\Pr_u, \Pr_\nu$  be defined in Definition 7. The stopping times of RW  $j_0$  with respect to  $\mathcal{A}$ , starting from  $X_{j_0}(0) = u$  and  $X_{j_0}(0) \sim \nu$ , are defined as:

$$K_u \triangleq \inf \{t > 0 : X_{j_0}(t) \in \mathcal{A}, X_{j_0}(0) = u\}, \quad (46)$$

$$K_\nu \triangleq \inf \{t > 0 : X_{j_0}(t) \in \mathcal{A}, X_{j_0}(0) \sim \nu\}. \quad (47)$$

We have  $\Pr_u(K_u < \infty) = 1$  and  $\Pr_\nu(K_\nu < \infty) = 1$ .

**Definition 17.** (Active distribution) Consider a strongly connected graph  $\mathcal{G}$  with absorbing states  $\mathcal{A}$ , as defined in Definition 5. Let a chain of RWs  $\{j_s\}_{s \geq 0}$  be defined in Definition 1. Let  $K_u$  be defined in (46). For any  $t$  and  $I \subset \mathcal{B}$ , we define the active distribution of RW  $j_0$  at time  $t$  as

$$\xi_{0;t}(I; u) \triangleq \Pr_u(X_{j_0}(t) \in I \mid K_u > t). \quad (48)$$

Let  $t_s$  denote the birth time of RW  $j_s$ , and suppose that its initial location  $X_{j_s}(t_s)$  is drawn from a distribution  $\nu_s$ , which depends on  $u$ . For any  $t \geq t_s$ , we define the active distribution of RW  $j_s$  as

$$\xi_{s;t}(I; u) \triangleq \Pr_{\nu_s}(X_{j_s}(t) \in I \mid K_{\nu_s} > t - t_s), \quad (49)$$

where the subscript  $\nu_s$  emphasizes that the RW is initialized according to  $\nu_s$ . The dependence of  $\nu_s$  on  $u$  is implicit in this notation  $\xi_{s;t}(I; u)$ .

Couple the newly created RW with its parent RW such that, after creation, the new RW is independently reinitialized as an *i.i.d.* replica of the original RW. That is, it evolves independently and has the same probability distribution as the original RW. Consequently, at any time while the parent RW remains active, the probability distribution of the newly created RW coincides with that of an independent copy of the parent RW. Recall that we remove the waiting time, applying this argument recursively, at any time  $t$ , the active probability distribution of any active RW coincides with that of any ancestor, as long as the ancestor remains active. Therefore, for any  $s > 0$ , we have

$$\xi_{s;t} \stackrel{d}{=} \xi_{0;t}. \quad (50)$$

At any time  $t$ , we re-parameterize the active distribution of the the most recently created (i.e., latest-born) RW as  $\xi_t$ . Since the

chain  $\{j_s\}_{s \geq 0}$  is infinite, we now analyze limiting behavior of the probability distribution  $\xi_t$ . From (48) and (50), we have

$$\begin{aligned} \lim_{t \rightarrow \infty} \xi_t(I; u) &= \lim_{t \rightarrow \infty} \xi_{0;t}(I; u) \\ &= \lim_{t \rightarrow \infty} \Pr_u(X_{j_0}(t) \in I \mid K_u > t). \end{aligned} \quad (51)$$

If the limit in (51) exists, it is referred to as the Yaglom limit [27]. This limit depends on the initial location. Intuitively, the Yaglom limit captures the long-term distribution of the process conditioned on survival. It remains to show that the limit  $\lim_{t \rightarrow \infty} \xi_t(I; u)$  exists and to derive its explicit expression.

**Definition 18.** (Quasi-Stationary Distribution [28]) Consider a strongly connected graph  $\mathcal{G}$  with absorbing states  $\mathcal{A}$ , as defined in Definition 5. Let  $K_u$  be defined in (46). We say that  $\nu$  is a quasi-stationary distribution (QSD) of RW  $j_0$  if, for all  $t \geq 0$  and any set  $I \subset \mathcal{B}$ ,

$$\nu(I) = \Pr_\nu(X_{j_0}(t) \in I \mid K_u > t).$$

The following Lemma 5 shows that the distribution of a chain of RWs converges asymptotically to that of a single RW conditioned on long-term survival. This provides a way to obtain the explicit expression of  $\lim_{t \rightarrow \infty} \xi_t(I; u)$ .

**Lemma 5.** Consider a robustly connected graph  $\mathcal{G}$  with absorbing states  $\mathcal{A}$ , as defined in Definition 5. Let  $\{j_s\}_{s \geq 0}$  be an infinite chain, as defined in Definition 1. Suppose the initial RW  $j_0$  starts at node  $u \in \mathcal{B}$ . We define the distribution of the chain at time  $t$  as

$$\pi_{\text{chain},t} \triangleq \xi_t. \quad (52)$$

Let  $t \rightarrow \infty$ , the distribution of a chain is convergent:

$$\lim_{t \rightarrow \infty} \pi_{\text{chain},t} = \nu^{(\zeta)} \quad (53)$$

where  $\nu^{(\zeta)}$  is the left normalized leading eigenvector of  $Q^{(\zeta)}$  (as defined in (8) and (9)).

*Proof:* From Definition 5, the submatrix  $Q^{(\zeta)}$  is irreducible. By [27, Section 2] or [29, Theorem 16.11], the irreducibility of  $Q^{(\zeta)}$  ensures the existence of the corresponding Yaglom limits (see (51)), which is convergent in total variation.

Moreover, since  $Q^{(\zeta)}$  is irreducible and aperiodic, any existing Yaglom limit (with any initial state  $u$ ) coincides with a QSD, as established in [27, Proposition 1]. Therefore, the Yaglom limit in (51) is a QSD for every  $u \in \mathcal{B}$ .

In our case, each initial RW is defined on a finite state space  $\mathcal{V}$  with a nonempty absorbing set  $\mathcal{A}$ . The restricted transition matrix  $Q^{(\zeta)}$  on the transient states  $\mathcal{B}$  is reducible and aperiodic. According to [30], the QSD exists and is unique. As a result, the Yaglom limit in (51) converges to the same QSD for all initial states  $u \in \mathcal{B}$ .

Meanwhile, the QSD can be calculated as the leading left eigenvector of  $Q^{(\zeta)}$ , normalized to sum to one [30, Eqn. (10) and the third equation on p. 99]. Thus, according to (52), the limiting distribution  $\lim_{t \rightarrow \infty} \pi_{\text{chain},t}$  is given by:

$$\lim_{t \rightarrow \infty} \pi_{\text{chain},t} = \nu^{(\zeta)}.$$

Therefore

$$\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\| \leq \frac{2}{\mu} \|\nabla f(\tilde{\mathbf{x}}^*)\|.$$

The equality holds when  $\nabla f(\tilde{\mathbf{x}}^*)$  is co-linear with  $\tilde{\mathbf{x}}^* - \mathbf{x}^*$  [31].

Similarly, using the  $L$ -Lipschitz condition, we derive:

$$L\|\mathbf{x}^* - \tilde{\mathbf{x}}^*\| \geq \|\nabla f(\mathbf{x}^*) - \nabla f(\tilde{\mathbf{x}}^*)\|.$$

By the optimality conditions, we have:

$$\nabla f(\mathbf{x}^*) = 0,$$

which implies

$$\|\mathbf{x}^* - \tilde{\mathbf{x}}^*\| \geq \frac{1}{L} \|\nabla f(\tilde{\mathbf{x}}^*)\|.$$

The equality holds when  $\tilde{\mathbf{x}}^* - \mathbf{x}^*$  aligns with the directional of maximal curvature of  $f(\mathbf{x})$  [31].

*Proof of Part (2).* This proof follows the same argument as in [32, Theorem 1], with the necessary substitutions under our setting. Specifically, by Theorem 2, the modified stationary distribution of a single effective RW (i.e., the chain of RWs) is  $\tilde{\nu}^{(\zeta)}$  with  $\zeta \in (0, 1]$ . The corresponding transition probability matrix is given in Appendix E. Based on Assumption 4, we set  $w(u) = 1$  for all  $u \in \mathcal{V}$ . By substituting  $\tilde{\nu}^{(\zeta)}$ ,  $P_{\text{chain}}$ ,  $\eta_{\text{chain}}$ , and the original sampling distribution  $\pi$  into the proofs of [32, Lemmas 1, 2, Theorem 1], and by *artificially condensing the time interval between the child and its parent*, we obtain the following bounds:

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{x}}_T - \mathbf{x}^*\| &\leq 2(1 - \gamma\mu)^T \|\mathbf{x}_0 - \mathbf{x}^*\|^2 + \frac{\gamma L \sigma^2}{\eta_{\text{chain}} \mu^2} \\ &\quad + \frac{\|\tilde{\nu}^{(\zeta)} - \pi\|_{\text{TV}}^2 \sigma^2 L}{\mu^3}. \end{aligned}$$

## APPENDIX G

### ADDITIONAL SIMULATIONS

In this section, we present additional simulation results of Section V.

#### A. Additional Graphs

Besides the complete graph, we also consider three other connected graphs, including: (i) a random regular graph with degree  $d = 8$  (expander graph [33]), (ii) a regular graph with degree  $d = 2$  (ring topology), and (iii) an Erdős–Rényi graph with the edge probability  $p = 0.1$ .

#### B. Data Partitioning for the Public Benchmark Dataset

We use the standard MNIST handwritten digit dataset [25]. The dataset is evenly partitioned into 100 disjoint subsets, with each node is assigned a unique subset. We consider both *i.i.d.* and *non-i.i.d.* data partitioning schemes for distributing data across network nodes. In the *i.i.d.* case, the dataset is uniformly and independently divided into 100 disjoint subsets, one for each node. In the *non-i.i.d.* case, data heterogeneity is introduced by sampling from a Dirichlet distribution [34]: The concentration parameter  $\alpha$  controls the degree of heterogeneity: as  $\alpha \rightarrow 10$ , the partitioning approaches the *i.i.d.* case;

In each chain of RWs, every child inherits the current model state (i.e.,  $\mathbf{x}_t$ ) from its parent. As a result, under the RW-SGD algorithm, each infinite chain asymptotically behaves as if a single effective RW is solving a surrogate optimization problem with a time-varying sampling distribution  $\tilde{\pi}_t$ . Specifically:

- 1) When  $\zeta = 1$ , the absorbing state  $\mathcal{A} = \{1, w\}$ , so  $\tilde{\pi}_t = [0, \pi_{\text{chain},t}]$ , where  $\pi_{\text{chain},t}$  is a discrete distribution supported on a finite set of size  $N$ , and

$$\lim_{t \rightarrow \infty} \tilde{\pi}_t = [0, \nu^{(1)}].$$

- 2) When  $0 < \zeta < 1$ , the absorbing state  $\mathcal{A} = \{w\}$ , so  $\tilde{\pi}_t = \pi_{\text{chain},t}$ , where  $\pi_{\text{chain},t}$  is a discrete distribution supported on a finite set of size  $N + 1$ , and

$$\lim_{t \rightarrow \infty} \tilde{\pi}_t = \nu^{(\zeta)}.$$

## APPENDIX E

### THE EFFECTIVE TRANSITION PROBABILITY MATRIX $P_{\text{CHAIN}}^{(\zeta)}$

In fact, as discussed before, we condense the time interval between the termination of the last RW and the creation of the next RW, where  $\mathcal{A}$  represents the set of absorbing states (as defined in Appendix D), i.e.,  $\mathcal{A} = \{1, w\}$  if  $\zeta = 1$  and  $\mathcal{A} = \{w\}$  if  $\zeta \in (0, 1)$ . For any nodes  $u, v$ , the transition probability matrix  $P_{\text{chain}}$  can be written as

$$[P_{\text{chain}}]_{uv} = \Pr(\{X_j(1) = v\} \mid \{X_j(0) = u, X_j(1) \notin \mathcal{A}\}).$$

Taking marginal distribution of the chain at time 0 as  $\mu$  and applying Bayes' rule, we obtain:

$$\begin{aligned} [P_{\text{chain}}]_{uv} &= \frac{\Pr(\{X_j(1) = v, X_j(0) = u, X_j(1) \notin \mathcal{A}\})}{\Pr(\{X_j(0) = u, X_j(1) \notin \mathcal{A}\})} \\ &= \frac{\mu_u Q_{uv}^{(\zeta)}}{\mu_u \sum_v Q_{uv}^{(\zeta)}} = \frac{Q_{uv}^{(\zeta)}}{\sum_v Q_{uv}^{(\zeta)}}. \end{aligned}$$

## APPENDIX F

### PROOF OF PROPOSITION 2

*Proof of Part (1).*

Since a chain of RWs behaves like a single RW that never dies, we can apply the convergence results of RW-SGD. According to [22]–[24], the standard RW-SGD algorithm converges to a deterministic limit when the stepsize  $\eta_t$  decreases with the number of iterations and tends to 0. Consequently, under the same stepsize condition, a chain of RWs converges to the optimizer of the surrogate optimization problem (5).

Let  $\tilde{\mathbf{x}}^*$  be the optimizer of either (5). Applying strong convexity, we obtain:

$$f(\mathbf{x}^*) \geq f(\tilde{\mathbf{x}}^*) + \nabla f(\tilde{\mathbf{x}}^*)(\mathbf{x}^* - \tilde{\mathbf{x}}^*) + \frac{\mu}{2} \|\mathbf{x}^* - \tilde{\mathbf{x}}^*\|^2,$$

which implies

$$0 \geq f(\mathbf{x}^*) - f(\tilde{\mathbf{x}}^*) \geq \nabla f(\tilde{\mathbf{x}}^*)(\mathbf{x}^* - \tilde{\mathbf{x}}^*) + \frac{\mu}{2} \|\mathbf{x}^* - \tilde{\mathbf{x}}^*\|^2.$$

By Cauchy–Schwarz inequality, it follows that

$$\frac{\mu}{2} \|\mathbf{x}^* - \tilde{\mathbf{x}}^*\|^2 \leq -\nabla f(\tilde{\mathbf{x}}^*)(\mathbf{x}^* - \tilde{\mathbf{x}}^*) \leq \|\nabla f(\tilde{\mathbf{x}}^*)\| \|\mathbf{x}^* - \tilde{\mathbf{x}}^*\|.$$

as  $\alpha \rightarrow 0$ , the data becomes highly non-*i.i.d.* Throughout this paper, we set  $\alpha = 0.5$ , which corresponds to a moderate level of non-*i.i.d.* partitioning. Under both partitioning schemes, whenever a RW visits a node, it uniformly samples a mini-batch of size  $B = 256$  from the node’s local data to perform a training update.

### C. Comparison with GOSSIP-BASED SGD

Another baseline we consider is the classical GOSSIP-BASED SGD [3]. In this scheme, each node transmits its locally updated model to all of its neighbors at every iteration, and model parameters are updated via neighborhood averaging. We incorporate the same adversarial setting as before: the Pac-Man node independently terminates each incoming model update with the termination probability  $\zeta$ . As a result, the Pac-Man node is unable to reliably incorporate all information from its neighbors, leading to biased and incomplete aggregation.

Fig. 4 shows that the SGD based on AC and DECAFORK algorithms converge significantly faster than GOSSIP-BASED SGD: the loss curves of AC and DECAFORK (blue and green curves, respectively) exhibit much faster decay than that of the gossip-based method (orange curve). This behavior is expected. GOSSIP-BASED SGD propagates information through repeated local averaging, which leads to diffusive information spread and systematic attenuation of gradient updates. Moreover, although the final consensus error of gossip-based SGD is small, the final average loss remains relatively large. This indicates that good consensus among nodes does not necessarily imply a good global model, especially in adversarial environments. In contrast, SGD based on the AC and DECAFORK algorithms preserve the strength of informative gradients and enables faster global impact.

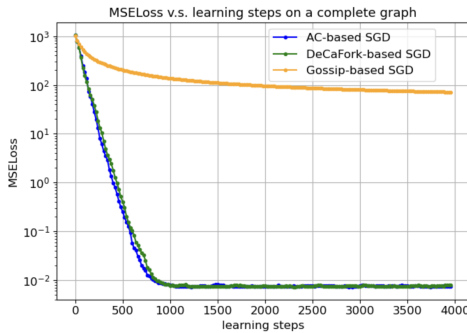


Fig. 4: Loss v.s. learning steps on a complete graph: comparison between AC, DECAFORK and gossip-based SGD.

### D. Boundedness and Persistence

In Fig. 5, we plot the RW population process  $\{Z_t\}_{t \in \mathbb{Z}}$  over time on regular, ring, and Erdős–Rényi graphs. The  $y$ -axis indicates the number of RWs (averaged over 100 iterations), and the  $x$ -axis denotes the time steps. On each graph, when the threshold  $A$  is small (note that the threshold for “small” and “large” varies by graph), the RW population stabilizes

at a relatively large value. In contrast, with large  $A$ , RW duplication becomes relatively rare, leading to population extinction.

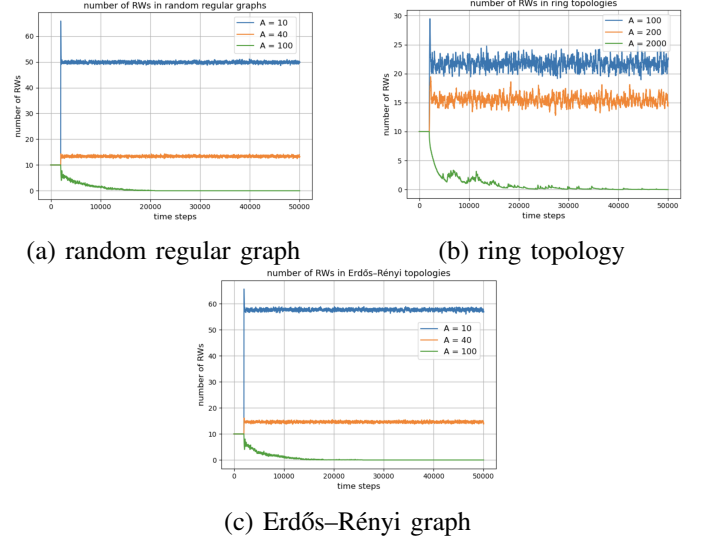


Fig. 5: Number of RWs over time on different graphs.

Although Proposition 1 focuses exclusively on a simplified version of the AC algorithm on a complete graph, we extend our investigation by simulating the AC algorithm extinction behavior on regular, ring and Erdős–Rényi graphs. In Fig. 6, the  $y$ -axis shows the approximated extinction probability<sup>11</sup>, while the  $x$ -axis denotes the value of  $A$ . when  $A$  exceeds a critical value (which depends on the graph topology), extinction occurs with probability 1. Conversely, when  $A$  falls below this critical threshold, the extinction probability drops sharply and approaches zero for sufficiently small values of  $A$ .

### E. Convergence

We begin with experiments on the synthetic dataset. Fig. 7 presents the convergence performance of RW-SGD under the proposed AC algorithm, in comparison with the DECAFORK baseline, on regular, ring, and Erdős–Rényi graphs. In each subfigure, the  $y$ -axis represents the value of the global loss ( $\log$  scale), while the  $x$ -axis denotes the number of time steps. Across different graph topologies, the loss consistently decreases over time and eventually reaches zero, indicating effective convergence.

The distance between the new convergence point and the original optimal solution, along with the corresponding theoretical bounds in the synthetic dataset, is reported in Table II. This table empirically validate the theoretical guarantees established in Proposition 2.

We then conducted experiments on the real-world dataset, considering both *i.i.d.* partitioning and non-*i.i.d.* partitioning scenarios.

<sup>11</sup>We approximate the extinction probability by running a large number of simulations over a long time horizon and computing the ratio of runs in which the RW population goes extinct to the total number of runs.



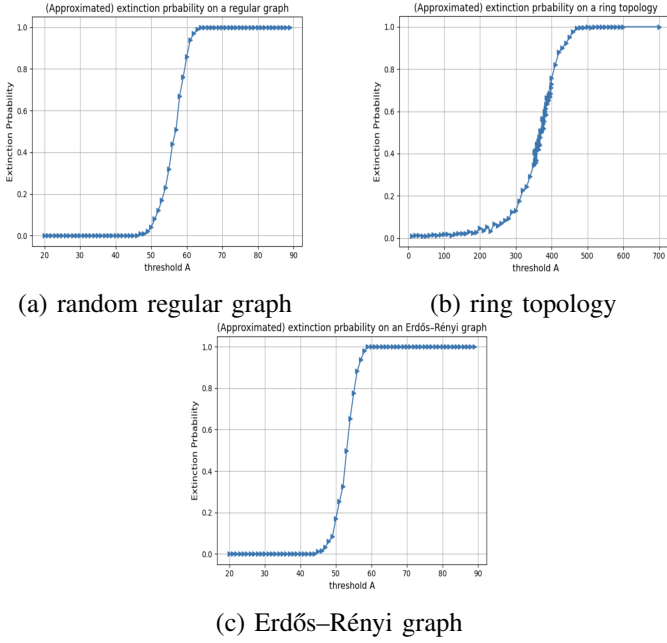


Fig. 6: Extinction probability vs. threshold  $A$  under different graphs.

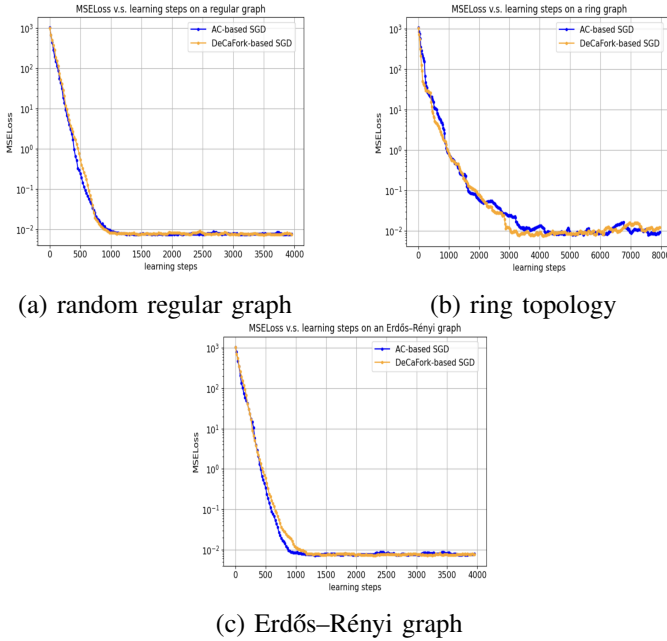


Fig. 7: Loss function v.s. learning steps on different graphs.

Fig. 8 and Fig. 9 show the convergence behavior of RW-SGD under the AC and DECAFORK algorithms, on the public benchmark dataset [25], using *i.i.d.* and non-*i.i.d.* data partitioning across nodes, respectively. In Fig. 8 and Fig. 9, each subfigure plots the global loss (on the  $y$ -axis) over time steps (on the  $x$ -axis). In both partitioning settings, the global loss drops quickly to near-zero levels and remains stable thereafter, confirming that the algorithm converges. Moreover, RW-SGD

graph type	Complete	Regular	Ring	Erdős-Rényi
$\ \tilde{\mathbf{x}}^* - \mathbf{x}^*\ $	0.019	0.015	0.051	0.018
$\frac{1}{L} \ \nabla f(\tilde{\mathbf{x}}^*)\ $	0.015	0.010	0.048	0.013
$\frac{1}{\mu} \ \nabla f(\tilde{\mathbf{x}}^*)\ $	0.044	0.032	0.146	0.035

TABLE II:  $\|\tilde{\mathbf{x}}^* - \mathbf{x}^*\|$  and its bounds on different graphs.

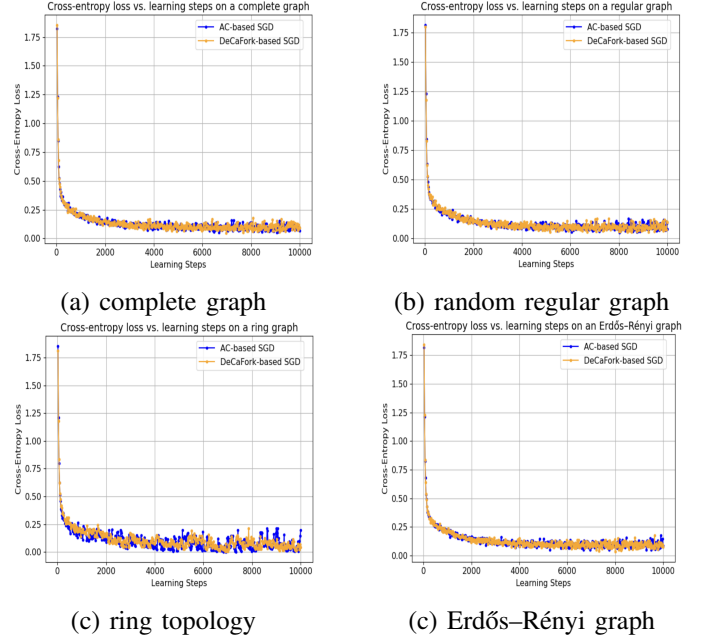


Fig. 8: Loss function v.s. learning steps on different graphs under *i.i.d.* partitioning of the real-world dataset.

exhibits almost identical convergence behavior under all three self-duplication mechanisms. This aligns with the observations in Fig. 7, as all three algorithms share the same RW transition matrix, differing only in their replication mechanisms, not in how the RWs traverse the network.

	Complete	Regular	Ring	Erdős-Rényi
A single SGD without the Pac-Man	0.9749	0.9748	0.9743	0.9751
AC	0.9732	0.9779	0.9764	0.9709
DeCaFork	0.9725	0.9731	0.9749	0.9712

TABLE III: Testing accuracies on different graphs under *i.i.d.* partitioning.

Finally, we evaluate the performance of the final models across different graphs using testing accuracies as the performance metric. We first examine the surrogate optimization problem (5), particularly the quasi-stationary distribution  $\nu^{(1)}$  (with  $\zeta = 1$ ). Given the target distribution  $\pi = (\frac{1}{100}, \dots, \frac{1}{100})$ , Theorem 2 implies that  $\nu^{(1)}$  is uniform over the 99 benign nodes in complete, random regular, and ring graphs, and approximately uniform in the Erdős-Rényi graph due to its near-regular structure. This indicates that in the presence of a Pac-Man node, active RWs under AC and DECAFORK visit benign nodes uniformly or near-uniformly.

Table III presents the performances of final models under *i.i.d.* data partitioning. From this table, we observe that the

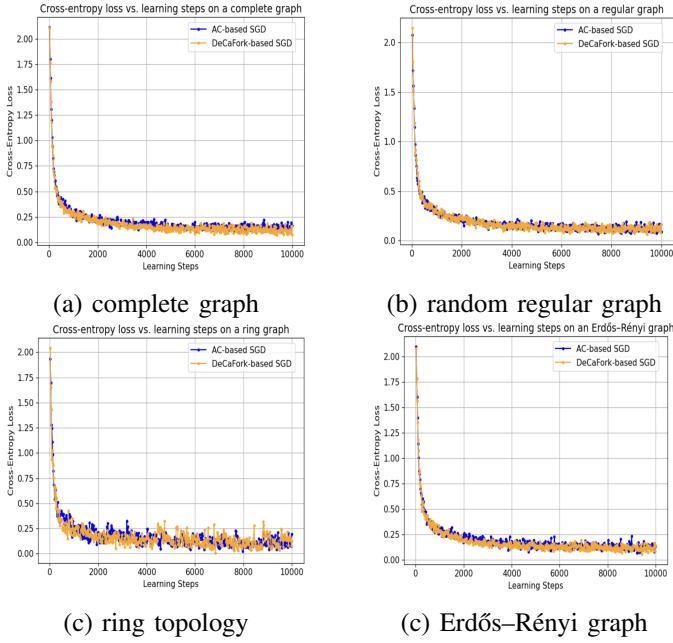


Fig. 9: Loss function v.s. learning steps on different graphs under non-*i.i.d.* partitioning of the real-world dataset.

performances are nearly identical across all graph types, regardless of whether a Pac-Man node is present. In the case of the single-SGD baseline without Pac-Man, the RW visits all nodes uniformly, and each node holds an *i.i.d.* portion of the dataset, yielding a faithful representation of the full data. In the Pac-Man cases, although the Pac-Man node is never visited, the benign nodes share the same data distribution (due to the *i.i.d.* partitioning) and are accessed uniformly (or approximately so) by active RWs. As the dataset is large (100 nodes with 600 samples each), excluding one node’s data has negligible effect, and the final models still perform well.

	Complete	Regular	Ring	Erdős-Rényi
A single SGD without the Pac-Man	0.9764	0.9719	0.9782	0.9751
AC	0.9613	0.9611	0.9617	0.9596
DeCaFork	0.9620	0.9606	0.9596	0.9621

TABLE IV: Testing accuracies on different graphs under non-*i.i.d.* partitioning.

Table IV shows the results under non-*i.i.d.* partitioning. We evaluate the performance of the final models across different graphs using testing accuracies as the performance metric. Table IV shows that the presence of a Pac-Man node impacts performance: the performance in the single SGD without the Pac-Man outperforms those in the Pac-Man cases, as the RW has full access to all nodes and thus to the complete data distribution; in contrast, the active RWs in the Pac-Man cases never access the Pac-Man node, and the remaining benign nodes—now with heterogeneous data (due to the non-*i.i.d.* partitioning)—no longer provide a representative sample of the entire dataset, resulting in degraded performance.