Age of Information Minimization on Graphs with GNN-DQN (Complete Simulator: Tasks 3a and 3b)

Roman Iashchenko*, Manyou Ma (Instructor)†
*HSE University, Faculty of Computer Science, Moscow, Russia
Email: roma.yashchenko3@gmail.com
†School of Artificial Intelligence, Shenzhen Technology University, Shenzhen, China
Email: mamanyou@sztu.edu.cn

***Abstract*—We study Age of Information (AoI) minimization in a graph-based status update system via reinforcement learning. We implement a custom simulator and compare heuristic baselines against graph neural network deep Q-learning (GNN-DQN). Task 3a trains a GNN-DQN on a fixed graph type, while Task 3b trains a generalized GNN-DQN on a distribution of random graphs and evaluates on unseen graph families (line, star, and new Erdos graphs).**

I. Motivation

Age of Information (AoI) measures how fresh the receiver's knowledge is about each node/process. Minimizing AoI is a canonical objective for status update systems with stochastic delivery and constrained actions [1]. A key difficulty is that the optimal policy depends on the *graph structure*, motivating graph-aware learning methods.

II. Related Work

AoI was introduced and studied in queueing/networked systems and scheduling [1], and surveyed in [2]. Deep reinforcement learning (DRL) has enabled scalable control via value-based methods such as DQN [4], building on the standard RL framework [3]. Graph neural networks (GNNs) allow policies/value functions to use relational structure; we use a message-passing encoder inspired by GCN-style aggregation [5].

III. System Model

We consider a graph $G = (V, E)$ with $|V| = N$ nodes. Each node i has an AoI state $a_i(t) \in \mathbb{R}_{\geq 0}$. At each discrete step t, the agent selects an action $u(t) \in \{1, \ldots, N\}$ (select a node to update). The update succeeds with probability p; on success, the chosen node's age is reset (or reduced) while others increase by one. We use the per-step reward:

$$r(t) = -\frac{1}{N}\sum_{i=1}^{N} a_i(t), \tag{1}$$

so maximizing return corresponds to minimizing mean AoI. Each episode lasts a fixed horizon T.

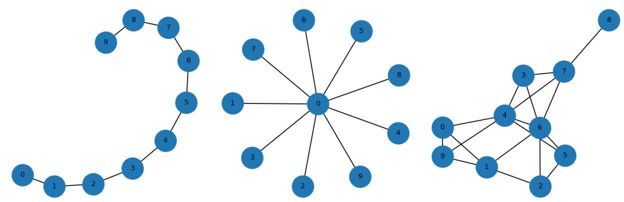


Fig. 1: Examples of evaluated graph families (illustrations).

A. Graph Families

We evaluate on three graph families:

- Line graph (path) with $N = 10$
- Star graph with $N = 10$
- Erdos–Renyi $G(N, p_e)$ (training distribution and unseen test graphs)

IV. Simulation Study

A. Baselines

We compare against:

- **Random**: uniform node selection
- **Greedy**: select the node with the largest current AoI
- **Degree-weighted**: select $\arg\max_i \big(\deg(i) \cdot a_i\big)$

B. DRL Methods

Task 3a (Fixed-graph GNN-DQN). We train a GNN-based Q-network $Q_\theta(s, i)$ on a fixed graph type (e.g., line or erdos) using DQN-style TD learning [4]. The GNN processes node features (AoI and degree-related signal) and outputs Q values for each node action.

Task 3b (Generalized GNN-DQN). We train on a *distribution* of random Erdos graphs, re-sampling a graph instance across episodes. We checkpoint and select the best model using an unseen graph set, then evaluate on line/star/erdos-new families.

C. Simulation Parameters

Table I summarizes the key parameters used in the experiments.

TABLE I: Simulation Parameters (3a/3b)

Parameter	Value
Nodes N	10
Max episode length T	50
Success probability p	0.8
Discount γ	0.99
Learning rate	1×10^{-3}
Hidden size	64
Batch size	64
Replay buffer	20000
Warmup	500 steps
Target update	every 200 steps
ϵ schedule	multiplicative decay

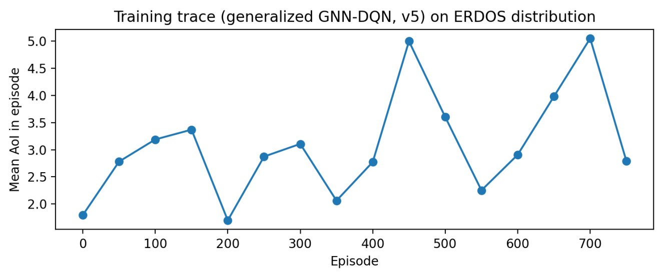


Fig. 2: Training trace: episodic mean AoI for generalized GNN-DQN (Task 3b).

D. Training Trace

Figure 2 plots the episodic mean AoI during generalized training (Task 3b). Lower is better.

E. Final Performance

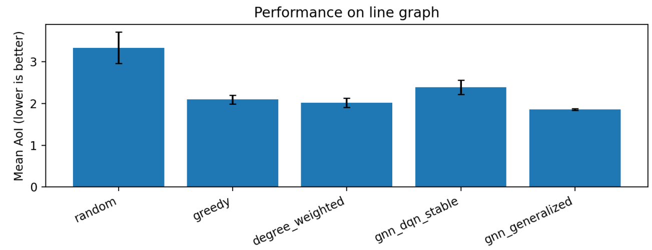
Figure 3 show final mean AoI on the evaluated graph families, comparing baselines with the learned policies.

F. Policy Behavior Diagnostics

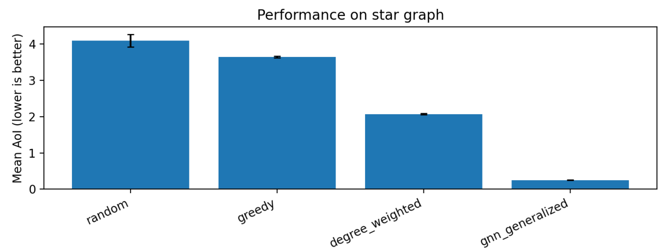
We also inspect action histograms for the generalized policy (Task 3b), shown in Fig. 4. This helps detect policy collapse (e.g., repeatedly selecting the same node).

V. Reflection (Facts)

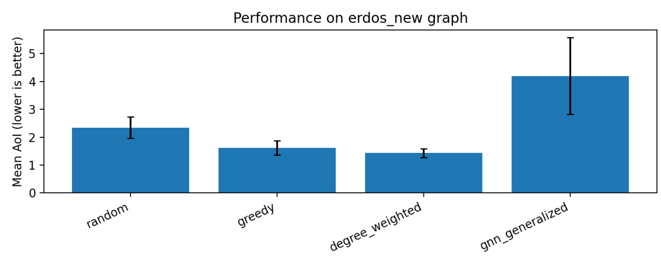
- Task 3a (fixed-graph training) achieved competitive AoI on line graphs but failed to generalize to erdos graphs under the same setup.
- Task 3b (generalized training on Erdos distribution) achieved strong performance on star and line graphs, while remaining worse than the degree-weighted heuristic on unseen erdos graphs.
- The action histogram shows that on star graphs the learned policy can collapse to always selecting a single action (node index 0), consistent with the structure where the center dominates.



(a) Line graphs.



(b) Star graphs.



(c) Unseen Erdos graphs (erdos_new).

Fig. 3: Final mean AoI across graph families (lower is better).

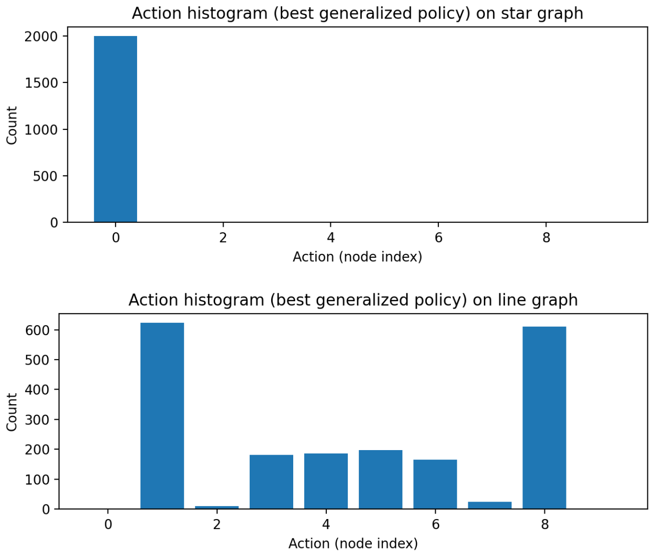


Fig. 4: Action histograms for the generalized policy on star and line graphs.

APPENDIX

Due to space constraints, we provide the full simulator and training code in the accompanying GitHub repository. The implementation includes the environment (`reset/step/observation`), graph generation utilities, baselines, and GNN-DQN training loops for Tasks 3a and 3b.

REFERENCES

[1] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Real-time status: How often should one update?," in *Proc. IEEE INFOCOM*, 2012.

[2] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[4] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017.