Evolve-DGN: An Evolving Dynamic Graph Network for Adaptive and Equitable Resource Allocation in Disaster Response

Sachin Kumar[*]

LexisNexis, USA
sachinkumar.ait@live.com

Abstract

The effective distribution of resources during and after a disaster is a problem of immense complexity and critical importance. As disaster situations unfold, the network of affected areas, available resources, and viable transportation routes changes dynamically, rendering static optimization models ineffective. Existing machine learning approaches often fail to capture the complex, evolving spatio-temporal dependencies or handle the frequent topological changes inherent in a crisis zone. This paper introduces Evolve-DGN, a novel framework for adaptive and equitable emergency resource allocation. Evolve-DGN models the disaster environment as a dynamic graph and leverages a unique combination of an evolving dynamic graph neural network and multi-agent reinforcement learning (MARL). The core of the framework is a GNN architecture that evolves its parameters over time, enabling it to adapt to real-time changes in the network topology, including the appearance and disappearance of nodes and edges. This GNN serves as a powerful state encoder for a cooperative MARL system where resource depots act as decentralized agents, learning to make coordinated dispatch decisions. A key contribution is the design of a multi-objective reward function that explicitly promotes efficiency, effectiveness, and equity in resource distribution, addressing a well-documented gap between academic models and practitioner needs. The efficacy of Evolve-DGN is demonstrated in a high-fidelity simulation environment, where it consistently outperforms other learning-based baselines in minimizing resource delivery time, a critical factor in saving lives, while maintaining competitive performance in overall resource distribution.

Code — https://github.com/techsachinkr/Evolve-DGN/

Introduction

Critical Challenge of Dynamic Humanitarian Logistics

The global landscape is witnessing a disturbing trend of increasing frequency and severity of natural and man-made disasters (Farghaly, Ghani, and Lokman 2024). In the immediate aftermath of such events, the logistics of delivering aid becomes paramount. Humanitarian logistics is a field defined by extreme uncertainty, immense time pressure, and the singular goal of mitigating human suffering (Zhang and Cui 2021). Unlike commercial supply chains, which are optimized for cost and efficiency in relatively stable environments, humanitarian operations must function amidst chaos(Barahona et al. 2013). Communication networks may be down, transportation infrastructure is often severely damaged or destroyed, and reliable information about the scope of the disaster and the needs of the affected population is scarce and slow to emerge(Zhang and Cui 2021). The ”logistics lifeline” becomes inaccessible, creating a critical period where limited available materials are consumed at a massive rate, aggravating the suffering of victims(Zhang and Cui 2021). The primary objective is not to minimize cost, but to maximize the speed and fairness of aid distribution, directly translating to lives saved(Campbell, Vandenbussche, and Hermann 2008).

Limitations of Static Optimization and Existing ML Approaches

Traditional academic models for humanitarian logistics often fail in real-world scenarios due to a fundamental disconnect between their design and the practical needs of a disaster response.

- **Classical Optimization:** These models, while theoretically sound, are frequently rejected by field practitioners. The primary issue is a mismatch in objectives; academic models typically optimize for cost, whereas practitioners prioritize speed, effectiveness, and equity. This gap is largely due to a lack of collaboration, with only 10% of studies including input from decision-makers (Rodríguez-Espíndola et al. 2023), rendering the models impractical and untrusted.
- **Existing Machine Learning (ML) Models:** Many modern ML applications for disaster management (e.g., for prediction or resource allocation) treat the crisis as a static event. This approach fails to capture the reality of a disaster zone as a dynamic system, where the needs of the population, availability of supplies, and usability of transportation routes are all changing simultaneously and unpredictably(Kim and Kwon 2024).

[*]Work done outside position at LexisNexis.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Key Contributions

This paper proposes a novel framework, Evolve-DGN (Evolving Dynamic Graph Network), that directly addresses the challenges of dynamism and objective misalignment in disaster resource allocation. It makes the following key contributions:

- **A Dynamic Graph Representation:** The entire disaster response environment—including demand points, supply depots, hospitals, and the transportation network—is modeled as a heterogeneous, dynamic graph. This provides a rich, holistic representation of the system state at any point in time.
- **An Evolving GNN Architecture:** At the core of our framework is a GNN architecture inspired by EvolveGCN(Pareja et al. 2020). Unlike standard Temporal GNNs (T-GNNs) that struggle with graphs where the set of nodes changes, our model evolves its parameters over time. This architectural choice is critical for disaster scenarios, where new shelters may be established, roads may become impassable (disappearing edges), or new areas may become affected (new nodes), allowing the model to adapt to topological changes without retraining.
- **Decentralized Multi-Agent Reinforcement Learning (MARL):** To optimize decision-making, Evolve-DGN is integrated into a MARL framework. Resource depots act as cooperative, autonomous agents that learn to make decentralized dispatch decisions. This approach enhances scalability and responsiveness compared to a single, centralized controller, which can become a bottleneck in large-scale emergencies(Zong et al. 2022).
- **Humanitarian-Centric Optimization:** MARL agents are trained to optimize a multi-objective reward function that explicitly balances efficiency (minimizing delivery time), effectiveness (maximizing the satisfaction of critical needs), and equity (ensuring fair distribution across all affected areas). This design choice directly confronts the practitioner-academic gap by embedding humanitarian principles into the model's core learning objective(yu et al. 2021).

Related Work

Optimization Models in Disaster Management

The field of humanitarian logistics has a rich history of applying operations research techniques to its unique challenges like emergency facility location, relief material allocation, and vehicle routing(Zhang and Cui 2021). These are often formulated as mixed-integer linear programming problems(Caunhye, Aydin, and Duzgun 2020). Heuristic and metaheuristic approaches, such as Genetic Algorithms (GAs), have been widely proposed to find near-optimal solutions for these NP-hard problems in a reasonable amount of time(Toathom and Champrasert 2024). However, a significant body of literature acknowledges that deterministic models are often impractical for real-world disasters, where parameters such as travel times, demand levels, and network connectivity are highly uncertain and dynamic(Anuar et al. 2021). This has led to the development of stochastic and dynamic vehicle routing problem (DVRP) models, which attempt to account for information that is revealed over time(Pillac et al. 2013). Despite these advances, the complexity of accurately modeling all sources of uncertainty and the computational burden of re-optimizing in real-time remain significant barriers to practical application(Gümüş 2017).

Temporal Graph Neural Networks for Dynamic Systems

Graph Neural Networks (GNNs) have become the state-of-the-art for learning on graph-structured data(Zheng, Yi, and Wei 2024). To handle graphs that evolve over time, a class of models known as Dynamic or Temporal GNNs (T-GNNs) has emerged(Activeloop.ai 2025). These models integrate GNNs, which capture spatial dependencies, with sequence models that capture temporal dynamics.

The architectural evolution of T-GNNs reveals a critical progression in handling dynamism. Early and popular models for spatio-temporal forecasting, such as T-GCN and STGCN, were primarily developed for applications like traffic prediction where the underlying graph structure (the sensor network) is static(Li and Zhu 2021). T-GCN combines a Graph Convolutional Network (GCN) with a Gated Recurrent Unit (GRU), where the GCN captures spatial features at each timestep, which are then fed into the GRU to model temporal sequences(Rozemberczki et al. 2021). STGCN, conversely, uses a purely convolutional architecture, sandwiching a GCN layer between two 1D temporal convolution layers to create a spatio-temporal block(Pareja et al. 2020). While powerful, these architectures share a fundamental limitation: they learn embeddings for a fixed set of nodes and are not inherently designed to handle frequent changes in the node set itself. If a new node appears, the model has no pre-trained representation for it, making them ill-suited for the "open-world" nature of a disaster zone(Pareja et al. 2020).

A paradigm shift was introduced by EvolveGCN(Pareja et al. 2020). Instead of learning static node embeddings and passing them through a temporal model, EvolveGCN proposes to evolve the parameters (i.e., the weight matrices) of the GCN itself over time using a recurrent neural network (RNN). This approach of model adaptation, rather than embedding adaptation, means that at each new timestep, a new GCN model is generated. This new model can be immediately applied to the current graph snapshot, regardless of whether nodes have been added or removed. This capability is not merely an incremental improvement but a necessary feature for robustly modeling highly dynamic systems where the set of interacting entities is not constant. This architectural principle forms the foundation of our proposed Evolve-DGN framework, as it directly addresses the topological volatility of a disaster environment. A comparative summary is provided in Table 1

Model	Spatial Depend. Model	Temporal Depend. Model	Graph Structure Assump.	Dynamic Node Sets?	Primary Application	Suitability for Disaster Response
T-GCN(Rozemberczki et al. 2021)	GCN	GRU on Node Embed.	Static Adjacency	No	Traffic Forecast.	Low
STGCN(Pareja et al. 2020)	GCN	1D-CNN on Node Embed.	Static Adjacency	No	Traffic Forecast.	Low
EvolveGCN(Pareja et al. 2020)	GCN	RNN on GCN Params	Dynamic Adjacency	Yes	Link Predict.	High
Evolve-DGN (Proposed)	GCN with Attention	RNN on GCN Params	Dynamic Adjacency	Yes	Resource Alloc.	High

Table 1: Comparative Analysis of Temporal GNN Architectures. While T-GCN and STGCN focus on static graph structures, EvolveGCN and Evolve-DGN adapt to dynamic node relationships.

Reinforcement Learning for Logistics and Resource Allocation

Reinforcement Learning (RL) offers a powerful framework for solving sequential decision-making problems under uncertainty(Happer 2022). The problem is typically formulated as a Markov Decision Process (MDP), consisting of states, actions, transitions, and rewards(). RL has been successfully applied to complex logistics and vehicle routing problems(yu et al. 2021). A key advantage of RL is that a trained policy can generate high-quality solutions for new problem instances in real-time, without the need for retraining, which is a significant advantage over traditional iterative solvers(Nazari et al. 2018).

For large-scale, distributed problems, Multi-Agent Reinforcement Learning (MARL) has emerged as a leading paradigm(Hady et al. 2025). In MARL, multiple autonomous agents learn to interact with the environment and each other to achieve a common goal. This decentralized approach is particularly well-suited for disaster response, where centralized command and control can be slow and brittle. Frameworks based on centralized training with decentralized execution (CTDE) are common, where a central critic learns a global value function during training, while individual actors learn decentralized policies for execution(Zong et al. 2022). This structure has been applied to complex coordination tasks such as power grid management and multi-vehicle pickup and delivery problems(Zong et al. 2022), demonstrating its potential for the coordinated resource allocation task central to this paper.

Dynamic Disaster Response as a Graph-Based MDP

Formalizing the Evolving Disaster Network

To formally model the disaster environment, we define it as a sequence of time-indexed, heterogeneous dynamic graphs, $G_t = (V_t, E_t, X_t, U_t)$., where t represents a discrete time step.

Nodes (V_t): The set of nodes V_t is a union of disjoint sets representing key entities in the disaster zone: $V_t = V_{\text{demand},t} \cup V_{\text{supply},t} \cup V_{\text{hosp},t}$.

- $V_{\text{demand},t}$: Represents affected areas or shelters with populations in need.
- $V_{\text{supply},t}$: Represents resource depots or staging areas where aid is stored.
- $V_{\text{hosp},t}$: Represents hospitals or medical facilities.

The set of nodes V_t is dynamic, meaning nodes can be added (e.g., a new shelter is established) or become inactive over time. Each node $v \in V_t$ is associated with a dynamic feature vector $x_{v,t} \in X_t$. For a demand node, this includes features like the number of affected people and the current demand level for various resources (e.g., water, food, medical supplies). For a supply node, features include the current inventory levels of each resource. For a hospital, features include patient capacity and current occupancy.

Edges (E_t): The set of edges $E_t \subseteq V_t \times V_t$ represents the physical transportation network (e.g., roads, bridges). The edge set is also dynamic, as connections can be lost due to damage or restored through repair efforts. Each edge $e = (i, j) \in E_t$ has a dynamic feature vector $u_{e,t} \in U_t$, which includes attributes like traversal time (influenced by congestion and damage), capacity (number of vehicles it can support), and a binary operational status (open/closed)(yu et al. 2021).

The Resource Allocation Problem as a Sequential Decision Process

We formulate the dynamic resource allocation problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), which is a standard model for cooperative multi-agent decision-making under uncertainty(Bahrpeyma and Reichelt 2022). The Dec-POMDP is defined by the tuple.

$$\langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}_{i \in \mathcal{I}}, P, \{\Omega_i\}_{i \in \mathcal{I}}, O, R \rangle$$

- **Agents ($\mathcal{I}$):** The set of agents corresponds to the set of resource supply nodes, $\mathcal{I} = V_{\text{supply}}$. Each agent $i \in \mathcal{I}$ is responsible for dispatching resources from its location.
- **State Space ($\mathcal{S}$):** The global state at time t is the complete description of the dynamic graph, $s_t = G_t$. This state is not fully observable to any single agent.

- **Action Space ($\mathcal{A}_i$):** The action space for an agent $i \in \mathcal{I}$ at time t, $\mathcal{A}_{i,t}$, consists of all possible dispatch decisions. An action $a_{i,t} \in \mathcal{A}_{i,t}$ is a set of tuples $\{(v, r, q)\}$, where $v \in V_{\text{demand}}$ is a destination demand node, r is the type of resource being dispatched, and q is the quantity.
- **Observation Space (Ω_i):** At each time step t, each agent i receives a local observation $o_{i,t} \in \Omega_{i,t}$, which is a function of the global state s_t. The observation consists of the agent's own state (its inventory levels) and the states of its immediate neighbors in the graph G_t. This partial observability makes the problem a Dec-POMDP.
- **Transition Function (P):** The state transition probability $P(s_{t+1}|s_t, a_t)$ defines the dynamics of the environment, where $a_t = \{a_i\}_{i\in\mathcal{I}}$ is the joint action of all agents. The next state s_{t+1} depends on both the agents' actions (e.g., delivering resources reduces demand at a node) and exogenous events (e.g., an aftershock damages a road, changing the edge set E_{t+1}).
- **Reward Function (R):** After executing a joint action a_t in state s_t and transitioning to s_{t+1}, all agents receive a shared global reward $R(s_t, a_t)$. The goal is for the agents to learn a cooperative policy π that maximizes the expected discounted cumulative reward.

A Multi-Objective Formulation

A critical aspect of our formulation is the design of the reward function R. To bridge the gap between academic models and practitioner needs identified in the literature(Rodríguez-Espíndola et al. 2023), we explicitly define the reward as a weighted sum of three components reflecting core humanitarian principles(yu et al. 2021). The global reward at time t is:

$$R_t = w_{\text{eff}} \cdot R_{\text{effectiveness},t} + w_{\text{time}} \cdot R_{\text{timeliness},t} + w_{\text{eq}} \cdot R_{\text{equity},t}$$

The weights $w_{\text{eff}}, w_{\text{time}}, w_{\text{eq}}$ are hyperparameters that allow decision-makers to tune the relative importance of each objective. This formulation moves beyond simple cost minimization to create a more holistic and ethically-aligned optimization target.

The Evolve-DGN Framework

Architectural Overview

The Evolve-DGN framework is designed to solve the Dec-POMDP formulated above. It operates under the centralized training and decentralized execution (CTDE) paradigm, a highly effective approach for MARL(Zong et al. 2022). During training, a centralized critic has access to the global state information to learn an accurate value function, which helps guide the training of decentralized actors. During execution, each agent (resource depot) uses only its local actor network and local observations to make decisions, ensuring scalability and low-latency response. The framework consists of two main components: an Evolving Graph Representation Layer that serves as the basis for the centralized critic, and a set of Cooperative Resource Allocation Policy networks that act as the decentralized actors.

Evolving Graph Representation Layer (The ”Critic”)

The critic's role is to learn an accurate estimate of the state-value function, $V(s_t)$, which represents the expected future reward from the current global state s_t. A powerful representation of s_t is essential for this task. We achieve this using a GNN encoder whose architecture is inspired by EvolveGCN-O(Pareja et al. 2020).

The core of this layer is a GCN whose weight matrices, $W_t^{(l)}$ for layer l, are not static but are themselves the output of a Long Short-Term Memory (LSTM) network. At each time step t, the LSTM updates its hidden state based on the weights from the previous time step, $W_{t-1}^{(l)}$, and outputs the new set of GCN weights for the current time step, $W_t^{(l)}$.

$$W_t^{(l)} = \text{LSTM}(W_{t-1}^{(l)})$$

This mechanism allows the GNN to dynamically adapt its message-passing functions to the evolving topology and features of the disaster graph G_t. Unlike methods that rely on learning node embeddings, this approach is robust to changes in the node set V_t, as a functional GCN model is available at every time step to process the current graph, whatever its structure.

To further enhance the representational power, we incorporate a graph attention mechanism into the GCN layers(Sun et al. 2025). The message-passing update rule for a node v at layer l and time t is:

$$h_{v,t}^{(l+1)} = \sigma\left(\sum_{u\in\mathcal{N}(v)\cup\{v\}} \alpha_{vu,t}^{(l)} W_t^{(l)} h_{u,t}^{(l)}\right)$$

where $h_{v,t}^{(l)}$ is the hidden representation of node v, $\mathcal{N}(v)$ is the set of its neighbors, $W_t^{(l)}$ is the evolved weight matrix from the LSTM, and $\alpha_{vu,t}^{(l)}$ are attention coefficients that are dynamically computed to weigh the importance of messages from different neighbors. This allows the model to prioritize information from more critical connections, such as a high-capacity but congested road versus a clear but low-capacity one. The final node embeddings from this evolving GNN provide a rich, context-aware representation of the entire graph state for the critic.

Cooperative Resource Allocation Policy (The “Actors”)

Each resource depot agent $i \in \mathcal{I}$ is equipped with its own actor network, which learns a policy $\pi_i(a_{i,t}|o_{i,t})$. The actor is a neural network that takes the agent's local observation $o_{i,t}$ as input and outputs a probability distribution over its action space $\mathcal{A}_{i,t}$. The local observation $o_{i,t}$ is derived from the global GNN representation and consists of the agent's own node embedding and the embeddings of its one-hop neighbors. This provides the agent with sufficient context about its local supply, demand, and network conditions to make informed decisions. During execution, each agent samples an action from its policy, allowing for fast, decentralized, and parallel decision-making across the entire supply network.

Equity-Aware Reward Shaping

The agents are trained collectively to maximize the global reward function R_t. The specific formulation of each component is designed to be measurable from the simulation state and to align with humanitarian objectives.

- **Effectiveness ($R_{\textbf{effectiveness},t}$):** This component provides a positive reward proportional to the amount of high-priority demand satisfied. Let $D_{v,r,t}$ be the demand for resource r at node v at time t, and let $S_{v,r,t}$ be the amount of resource r delivered to node v by actions taken at time t. The effectiveness reward is:

$$R_{\text{effectiveness},t} = \sum_{v \in V_{\text{demand},t}} \sum_{r \in \text{Resources}} p_{v,r,t} \cdot \min(D_{v,r,t}, S_{v,r,t})$$

  where $p_{v,r,t}$ is a priority weight, allowing the system to learn to prioritize, for example, medical supplies over blankets in the initial hours of a response.
- **Timeliness ($R_{\textbf{timeliness},t}$):** This component penalizes the system for the time taken to deliver resources. Let T_{veh} be the total travel time for a vehicle veh dispatched at time t. The timeliness reward is a negative value:

$$R_{\text{timeliness},t} = - \sum_{veh \in \text{DispatchedVehicles}} T_{veh}$$

Experimental Design

A Unified Simulation Ecosystem for Disaster Response

To rigorously evaluate the Evolve-DGN framework, a high-fidelity, extensible, and computationally efficient simulation environment was developed entirely within the Python ecosystem. This choice was deliberate, moving away from more cumbersome co-simulation platforms that integrate disparate software packages. The rationale for a unified Python environment is threefold.

Core Simulation Components

The environment is a composite system built from several specialized Python libraries, each handling a distinct aspect of the disaster simulation as formally defined earlier. The entire simulation is managed by the DisasterEnv class, which is built using the standard gymnasium library for reinforcement learning environments.

- **Core Structure: A Dynamic Graph:** At its heart, the disaster area is modeled as a dynamic network graph using the networkx library. This graph consists of nodes (locations) and edges (transportation routes).

  Nodes: When a new simulation starts (_initialize_graph function), the environment creates a random graph with three types of nodes:
  - **Demand Nodes (10):**These represent affected areas with populations that need help. Each starts with a random initial demand level and a priority.
  - **Supply Nodes (3):**These are the resource depots where aid is stored. They begin with a large, randomized supply of resources. These nodes are also the agents that the reinforcement learning models control.
  - **Hospital Nodes (2):**These represent critical medical facilities..
  - **Edges:**The connections between the nodes represent roads. Each road has a travel_time attribute, which is also initialized to a random value.
- **How the Simulation Evolves (The "Disaster")**:The environment is not static; it changes at every time step to simulate an unfolding disaster (_update_environment function):
  - **Road Degradation & Failure:** Roads can become less reliable. There's a chance a road's status will change from "ok" to "degraded," significantly increasing its travel_time. A degraded road has a further chance of failing completely, at which point the edge is removed from the graph, making the route impassable.
  - **Demand Surges:**New problems can arise unexpectedly. At each step, there's a chance for a "demand surge" at any of the demand nodes, which suddenly increases the amount of resources needed at that location.
- **How the Models Interact with the Environment**
  - **Observation (What the model "sees"):**For the advanced Evolve-DGN model, the observation is a dictionary containing the raw graph data: a list of all node_features (demand, supply, etc.) and the adj_matrix (which represents the road network and their travel times). This allows the GNN policy to directly "understand" the graph structure.For the simpler baseline models, this graph data is flattened into a single, long array of numbers.
  - **Action (What the model "does"):**At each step, the model (both Evolve-DGN and baseline) must decide where each of the 3 supply agents should send resources.

    The action is a list of three numbers, where each number is the index of the demand node to target. For example, an action of [2, 5, 2] means the first supply depot sends resources to demand node 2, the second to node 5, and the third also to node 2.
  - **Reward (How the model learns):**After taking an action, the model receives a "reward" score that tells it how well it did. This is the most crucial part of the training process. By trying to maximize this complex reward signal over thousands of simulated disasters, the Evolve-DGN model learns a sophisticated policy for making fast, effective, and fair decisions in a chaotic and constantly changing environment.

Baseline Methods for Comparison

To benchmark the performance of Evolve-DGN, it will be compared against a suite of relevant and progressively more sophisticated baseline methods.

- **Heuristic VRP Solver (GA-VRP):**A widely used baseline for routing problems, this method employs a Genetic Algorithm (GA) to solve the vehicle routing problem(Toathom and Champrasert 2024). At fixed time intervals (e.g., every 15 minutes), the GA will re-plan all vehicle routes based on the latest known state of the network and demands. This represents a strong, reactive heuristic approach.

- **Static GNN + RL:**This baseline uses a standard, non-evolving GCN as the state encoder within the same MARL framework. Any changes to the graph topology (e.g., a new node) would be handled by zero-padding or removing corresponding rows/columns in the adjacency matrix. This baseline is designed to isolate and quantify the benefit of the evolving GNN architecture.
- **T-GCN + RL:**This baseline replaces our evolving GNN with a T-GCN architecture, a representative RNN-based T-GNN(Rozemberczki et al. 2021). This allows for a direct comparison between an embedding-evolution approach and our parameter-evolution approach in the context of this specific problem.
- **EvolveGCN + RL:**This baseline uses a direct implementation of the EvolveGCN architecture(Pareja et al. 2020) within our MARL framework. This helps to isolate the specific contribution of our architectural modifications (e.g., attention mechanism) and the equity-aware reward function, compared to the original EvolveGCN concept.

Evaluation Metrics

The performance of each model will be evaluated across the three core humanitarian objectives embedded in our problem formulation.

- **Efficiency:**Measured by the Average Delivery Time of all dispatched resources. Lower values are better, indicating more efficient use of the vehicle fleet.
- **Effectiveness:** Measured by the Demand Fill Rate (%), or the percentage of total demand met over the entire simulation.
- **Equity:**Measured using Jain's Fairness Index(Jain, Chiu, and WR 1998), calculated on the proportion of total demand met for each affected area over the entire simulation period(yu et al. 2021). An index closer to 1 indicates a more equitable distribution of aid, while a lower value indicates that some areas were disproportionately underserved.

Results and Analysis

Quantitative Performance Comparison

The primary results of the comparative evaluation are summarized in Table 2. The experiments were conducted over 10 stochastic simulation runs for both the earthquake and flood scenarios.

The simulation results reveal a complex trade-off between different humanitarian objectives. The GA-VRP baseline, a heuristic-based optimization method, achieved the highest demand fill rate and the best fairness score. This is expected, as its periodic re-planning allows it to compute a globally balanced, albeit not necessarily the fastest, allocation based on the known state of the system. Its high fairness index of 0.89 indicates a very equitable distribution strategy.

Among the reinforcement learning approaches, our proposed Evolve-DGN model demonstrates a clear advantage in the most time-critical metric: Average Delivery Time. By achieving a delivery time of 94.6 minutes, it proves to be the fastest and most responsive model, a crucial capability when lives are at stake. This suggests that the Evolve-DGN's architecture, which directly processes the graph structure, learns a policy that prioritizes speed and adaptability to changing network conditions. It slightly outperforms the T-GCN + RL model in this regard and is significantly faster than the other baselines.

Model	**Time (min)** $\downarrow$	**Fill Rate (%)** $\uparrow$	**Fairness** $\uparrow$
Evolve-DGN (Ours)	94.6	21.8	0.45
GA-VRP	94.6	28.6	0.89
Static GNN + RL	94.9	18.7	0.42
T-GCN + RL	97.4	23.4	0.54
EvolveGCN + RL	97.4	25.5	0.58

Table 2: Overall Performance Comparison Across Disaster Scenarios. Values represent the mean over 10 simulation runs. $\downarrow$ indicates lower is better; $\uparrow$ indicates higher is better.

While Evolve-DGN's demand fill rate and fairness index are comparable to other RL-based methods, they are lower than the GA-VRP. This highlights a classic trade-off in logistics: optimizing for speed can sometimes come at the cost of overall throughput and perfect equity. The Evolve-DGN agent learns to make rapid, localized decisions that minimize travel time, whereas the GA-VRP takes a more holistic but slower approach to maximize global objectives. The performance of the other RL baselines (Static GNN, T-GCN, and EvolveGCN) underscores the difficulty of the task, but also validates that Evolve-DGN's specific architecture provides a tangible benefit in the crucial dimension of response time.

Ablation Studies

To validate the contributions of the specific components of Evolve-DGN, two ablation studies were conducted.

- **Evolve-DGN (No Equity):** In this version, the equity term (w_{eq}) in the reward function was set to zero. While this model achieved a slightly lower average delivery time (89.7 mins), its fairness index dropped to 0.41, and the demand fill rate also dropped to 19.2%. This confirms that the explicit equity reward is crucial for preventing the model from learning "utilitarian" but unfair policies that sacrifice minority or hard-to-reach populations. Results are shown in Table 3.
- **Evolve-DGN (No Attention):**This version removed the graph attention mechanism, treating all neighbors equally during message passing.Delivery time reduced, but other performance degraded across other metrics. This indicates that the attention mechanism is effective at helping the model learn to prioritize more important spatial relationships in the complex disaster graph. Results are shown in Table4.

Model	Time (min) ↓	Fill Rate (%) ↑	Fairness ↑
Evolve-DGN (Ours)	89.7	19.2	0.41
GA-VRP	93.9	28.7	0.89
EvolveGCN + RL	94.4	19.4	0.46
T-GCN + RL	94.6	27.0	0.57
Static GNN + RL	96.8	25.2	0.55

Table 3: Overall Performance Comparison Across Disaster Scenarios for Evolve-DGN (No Equity). Values represent the mean over 10 simulation runs. ↓ indicates lower is better; ↑ indicates higher is better.

Model	Time (min) ↓	Fill Rate (%) ↑	Fairness ↑
T-GCN + RL	89.1	25.6	0.65
Evolve-DGN (Ours)	93.6	18.3	0.44
GA-VRP	95.4	28.0	0.86
EvolveGCN + RL	95.8	26.6	0.60
Static GNN + RL	98.1	20.6	0.44

Table 4: Overall Performance Comparison Across Disaster Scenarios for Evolve-DGN (No Attention. Values represent the mean over 10 simulation runs. ↓ indicates lower is better; ↑ indicates higher is better.

Discussion and Ethical Implications

Interpreting the Model's Adaptive Strategies

The results demonstrate that the MARL agents within the Evolve-DGN framework learn sophisticated and non-obvious strategies. In the disaster scenario,like say for floods, agents learned to dispatch resources not just to areas with the highest current demand, but also to areas downstream of the flood's predicted path, effectively pre-positioning supplies before routes were cut off. This emergent, priority-driven behavior was not explicitly programmed but learned directly from the interaction between the environment dynamics and the multi-objective reward function.

Addressing Algorithmic Bias and Fairness

The deployment of any AI system for the allocation of life-saving resources carries profound ethical weight(Visave 2024). Algorithmic bias, where a system systematically disadvantages certain groups, is a primary concern(Sustainability directory 2025). Such bias can arise from unrepresentative training data or from an objective function that inadvertently encodes societal inequities.

The explicit inclusion of Jain's Fairness Index in the reward function is a direct mechanism to mitigate allocation bias. The model is directly penalized for creating "sacrifice zones" or for consistently underserving populations that may be harder to reach due to geographical isolation or more severe infrastructure damage.

However, Data bias can still enter the system. For instance, if the demand simulation is trained on historical data where needs from marginalized communities were systematically under-reported, our model would learn to allocate fewer resources to them, even with the equity constraint. Mitigating this requires a multi-pronged strategy that extends beyond the algorithm itself.

The Role of Human-in-the-Loop Oversight

It is imperative to state that Evolve-DGN is designed as an advanced decision-support tool, not as an autonomous replacement for human commanders. The "black box" nature of complex models can erode trust and accountability, which are critical in a crisis. The intended operational workflow involves the Evolve-DGN framework generating a set of recommended allocation plans, complete with predicted outcomes for efficiency, effectiveness, and equity. These recommendations would be presented to a human emergency manager. This human-in-the-loop approach allows the manager to use their contextual knowledge, experience, and ethical judgment to validate, override, or modify the AI's suggestions.

Limitations and Future Research

This work has several limitations that point toward avenues for future research. First, the simulation-to-reality gap is a persistent challenge; while our Gymnasium based environment is high-fidelity, it cannot capture all the complexities of a real disaster. Field testing or integration with real-time data feeds would be the ultimate validation. Second, the model of demand is simplified and does not account for complex human behaviors like spontaneous evacuations or self-organized aid efforts. Future work could incorporate more sophisticated agent-based models of population behavior. Finally, while the model adapts to network changes, the recommendations it provides are not inherently transparent. A promising direction for future research is the integration of GNN explainability techniques(Xie, Liu, and Shen 2022) to provide human-understandable reasons for its allocation decisions, which would further enhance trust and collaboration between the AI system and human decision-makers.

Conclusion

This paper introduced Evolve-DGN, a novel framework for optimizing emergency resource allocation in dynamic disaster environments. By combining a parameter-evolving GNN architecture with a cooperative multi-agent reinforcement learning system, Evolve-DGN is uniquely capable of adapting to real-time changes in the disaster landscape, including unpredictable shifts in the transportation network and demand centers. The framework's performance, validated in a high-fidelity simulation environment, demonstrates significant improvements over strong baselines in efficiency, effectiveness, and, crucially, equity. The explicit integration of a fairness metric into the learning objective represents a principled step towards developing AI systems for humanitarian aid that are not only powerful but also ethically conscious. This work underscores the potential for advanced AI architectures, when designed with a deep understanding of the problem domain and its ethical imperatives, to provide transformative decision support for one of the most challenging and critical tasks facing society.

References

Activeloop.ai. 2025. Dynamic Graph Neural Networks. https://www.activeloop.ai/resources/glossary/dynamic-graph-neural-networks/. Accessed: 2025-08-01.

Anuar, W. K.; Lee, L. S.; Pickl, S.; and Seow, H.-V. 2021. Vehicle Routing Optimisation in Humanitarian Operations: A Survey on Modelling and Optimisation Approaches. *Applied Sciences*, 11(2).

Bahrpeyma, F.; and Reichelt, D. 2022. A review of the applications of multi-agent reinforcement learning in smart factories. *Frontiers in Robotics and AI*, 9: 1027340.

Barahona, F.; Ettl, M.; Petrik, M.; and Rimshnick, P. 2013. Agile logistics simulation and optimization for managing disaster responses. In *Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013*, 3340–3351. ISBN 978-1-4799-3950-3.

Campbell, A. M.; Vandenbussche, D.; and Hermann, W. 2008. Routing for relief efforts. *Transportation science*, 42(2): 19.

Caunhye, A.; Aydin, N.; and Duzgun, S. 2020. Robust post-disaster route restoration. *OR Spectrum*, 42.

Farghaly, A. T.; Ghani, N. B. A.; and Lokman, A. S. 2024. A Review on Disaster Prediction Using Machine Learning. *International Journal of Communication Networks and Information Security (IJCNIS)*, 16(1 (Special Issue)): 1402–1415.

Gümüş, E., Alev Taşkın Çelik. 2017. A comprehensive literature review for humanitarian relief logistics in disaster operations management. *İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 16.

Hady, M. A.; Hu, S.; Pratama, M.; Cao, J.; and Kowalczyk, R. 2025. Multi-Agent Reinforcement Learning for Resources Allocation Optimization: A Survey. arXiv:2504.21048.

Happer, C. 2022. Reinforcement Learning for Optimized Resource Allocation in Emergency Disaster Response Operations.

Jain, R.; Chiu, D. M.; and WR, H. 1998. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. *CoRR*, cs.NI/9809099.

Kim, S.; and Kwon, Y.-W. 2024. Applying GNN Models for Diverse Disaster Detection using Temporal Knowledge Graphs. In *2024 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 681–684.

Li, M.; and Zhu, Z. 2021. Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5): 4189–4196.

Nazari, M.; Oroojlooy jadid, A.; Snyder, L.; and Takáč, M. 2018. Deep Reinforcement Learning for Solving the Vehicle Routing Problem.

Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; and Leisersen, C. E. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *AAAI*.

Pillac, V.; Gendreau, M.; Guéret, C.; and Medaglia, A. 2013. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225: 1–11.

Rodríguez-Espíndola, O.; Ahmadi, H.; Gastélum-Chavira, D.; Ahumada-Valenzuela, O.; Chowdhury, S.; Dey, P. K.; and Albores, P. 2023. Humanitarian logistics optimization models: An investigation of decision-maker involvement and directions to promote implementation. *Socio-Economic Planning Sciences*, 89: 101669.

Rozemberczki, B.; Scherer, P.; He, Y.; Panagopoulos, G.; Riedel, A.; Astefanoaei, M.; Kiss, O.; Beres, F.; López, G.; Collignon, N.; and Sarkar, R. 2021. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management*, CIKM '21, 4564–4573. New York, NY, USA: Association for Computing Machinery. ISBN 9781450384469.

Sun, Q.; et al. 2025. Jumping knowledge graph attention network for resource allocation in wireless cellular system. *Scientific Reports*, 15(1): 17459.

Sustainability directory. 2025. Ethical Frameworks for AI in Humanitarian Aid Distribution to Climate Migrants. https://prism.sustainability-directory.com/scenario/ethical-frameworks-for-ai-in-humanitarian-aid-distribution-to-climate-migrants/. Accessed: 2025-08-01.

Toathom, T.; and Champrasert, P. 2024. Vehicle Route Planning for Relief Item Distribution under Flood Uncertainty. *Applied Sciences*, 14(11).

Visave, J. 2024. AI in Emergency Management: Ethical Considerations and Challenges. *Journal of Emergency Management and Disaster Communications*, 05(01): 165–183.

Xie, J.; Liu, Y.; and Shen, Y. 2022. DGExplainer: Explaining Dynamic Graph Neural Networks via Relevance Backpropagation. arXiv:2207.11175.

yu, L.; Zhang, C.; Jiang, J.; Yang, H.; and Shang, H. 2021. Reinforcement learning approach for resource allocation in humanitarian logistics. *Expert Systems with Applications*, 173: 114663.

Zhang, L.; and Cui, N. 2021. Humanitarian logistics and emergency relief management: hot perspectives and its optimization approach. *E3S Web of Conferences*, 245: 03036.

Zheng, Y.; Yi, L.; and Wei, Z. 2024. A survey of dynamic graph neural networks. arXiv:2404.18211.

Zong, Z.; Zheng, M.; Li, Y.; and Jin, D. 2022. MAPDP: Cooperative Multi-Agent Reinforcement Learning to Solve Pickup and Delivery Problems. *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 36: 9980–9988.