Hierarchical Neuro-Symbolic Architectures for Periodic Capacitated Vehicle Routing: Augmenting Transformers with Temporal Scheduling Intelligence

1. Introduction: The Paradigm Shift in Waste Logistics

The optimization of waste collection logistics stands at the confluence of combinatorial optimization, environmental sustainability, and the rapidly evolving field of Deep Reinforcement Learning (DRL). The traditional operational model for waste management—characterized by static, predetermined schedules (e.g., weekly collection)—is increasingly viewed as an obsolescent framework that generates significant inefficiencies. Vehicles frequently traverse routes to service empty containers while neglecting overflowing bins in high-demand zones, leading to a dichotomy of wasted fuel and sanitation hazards. The integration of Internet of Things (IoT) sensors, which provide real-time data on waste fill levels (W_i), allows for a transition to **dynamic scheduling**, fundamentally altering the mathematical nature of the routing problem from a static Vehicle Routing Problem (VRP) to a dynamic Inventory Routing Problem (IRP) or Periodic Capacitated Vehicle Routing Problem (PCVRP).

The user's query posits a sophisticated starting point: a pre-existing Transformer model capable of solving the static Capacitated Vehicle Routing Problem (CVRP) given spatial coordinates (X, Y) and fill levels (W_i). This Transformer acts as a powerful "tactical" solver—given a set of nodes to visit, it can generate a near-optimal traversal sequence. However, the Transformer lacks the "strategic" capacity to determine the antecedent condition: *which* nodes should be visited on *which* days over a planning horizon to minimize long-term costs and prevent overflows. This report addresses the critical architectural gap: identifying the optimal deep learning model to augment the spatial Transformer with temporal decision-making capabilities.

Comprehensive analysis of the current research landscape indicates that the most effective solution is a **Hierarchical Deep Reinforcement Learning (H-DRL) framework**, employing a **Graph Attention Network (GAT)** augmented with **Long Short-Term Memory (LSTM)** encoders as the high-level "Manager" policy. This architecture effectively decouples the problem into two distinct decision spaces: the strategic management of inventory (waste levels) and the tactical optimization of routes. By assigning the scheduling responsibility to a specialized GNN-based agent that learns to cluster nodes based on urgency and spatial proximity, and utilizing the Transformer as a sub-routine for route generation, the system

achieves a synergy that outperforms monolithic architectures in multi-period environments.[1]

1.1 Defining the Mathematical Landscape: From VRP to IRP

To understand why a simple augmentation (such as adding a time dimension to the Transformer) is insufficient, one must define the problem variances. The standard VRP, which the user's Transformer solves, is defined on a graph $G = (V, E)$, where $V = \{0, 1, \dots, N\}$ represents the depot and customers. The objective is to minimize total distance D:

$$\text{Minimize } D = \sum_{(i,j) \in E} c_{ij} x_{ij}$$

subject to capacity constraints. The user's model receives inputs X, Y, W_i and outputs a permutation π of nodes.
However, the inclusion of "daily waste fill values" and the requirement to "decide on which days to make a route" transforms this into an Inventory Routing Problem (IRP) or a Periodic VRP (PVRP) with stochastic accumulation.4 In this domain, the state is dynamic. The fill level of bin i at time t, denoted W_{it}, evolves according to a stochastic process (accumulation rate r_i):

$$W_{i, t+1} = W_{it} + r_{it} \quad (\text{if not visited})$$

$$W_{i, t+1} = r_{it} \quad (\text{if visited})$$
The objective function expands to include not just routing costs, but inventory costs (penalties for overflows or holding costs):

$$\text{Minimize } J = \sum_{t=0}^T \left(\alpha \cdot \text{RoutingCost}_t + \beta \cdot \sum_{i \in V} \text{Penalty}(W_{it}) \right)$$
A monolithic Transformer trained solely to minimize routing distance will inherently fail in this environment because it is "myopic." It will prioritize the shortest path today, potentially ignoring a critical bin that is 90% full but slightly off-route, leading to a catastrophic overflow (and massive penalty) tomorrow. The "best" model must therefore be a **Policy Network** that optimizes the long-term value function J, effectively learning to trade off immediate routing efficiency for long-term system stability.[4]

1.2 The Failure of End-to-End Monolithic Models

While "End-to-End" DRL has shown promise for static problems [7], the literature suggests it struggles with the high dimensionality of PCVRP.

1. **State Space Explosion:** In a static VRP with 100 nodes, the input sequence length is 100. In a PCVRP with a 7-day horizon, the decision space involves scheduling visits for 100 nodes across 7 days, combinatorially expanding the search space. Transformers,

with $O(N^2)$ complexity, struggle to process these extended spatio-temporal sequences efficiently.[9]

2. **Conflicting Gradients:** The loss function for routing (geometry-based) and scheduling (inventory-based) often conflict. Jointly training a single network to master both tasks often leads to suboptimal convergence where the model learns neither task well.[1]

This necessitates the **Hierarchical** approach, where the "Manager" handles the temporal/inventory dimension and the "Worker" (Transformer) handles the spatial dimension.

2. The Optimal Architecture: Hierarchical Deep Reinforcement Learning (H-DRL)

The consensus across advanced neural combinatorial optimization research is that complex, multi-objective problems like PCVRP are best solved using a **Hierarchical Deep Reinforcement Learning (H-DRL)** architecture. This framework mimics a corporate logistics structure: a "Manager" makes high-level resource allocation decisions (which customers to serve), and a "Worker" executes the specific routing instructions.

2.1 The Manager-Worker Abstraction

The proposed system architecture consists of two nested control loops:

- **The Manager (High-Level Policy π_H):**
  - **Input:** The global state of the system S_t, comprising the locations (X, Y), current fill levels (W_i), predicted accumulation rates, and the current day t.
  - **Action:** A subset selection (mask) $M_t \in \{0, 1\}^N$, indicating which bins are to be serviced on day t.
  - **Frequency:** Operates once per time step (e.g., once per simulated day).
  - **Model Type: Graph Attention Network (GAT)**.
- **The Worker (Low-Level Policy π_L):**
  - **Input:** The subgraph G_t' induced by the mask M_t. This effectively reduces the input to a standard static CVRP instance.
  - **Action:** A sequence of nodes (permutation) representing the optimal route.
  - **Frequency:** Operates after the Manager selects the subset.
  - **Model Type:** The user's existing **Transformer** (e.g., AM, POMO, or similar).

This decomposition allows the user to retain their pre-trained Transformer without modification. The Transformer is treated as a "neural heuristic" or a complex environment function that returns a cost (route length) for any given subset provided by the Manager.[1]

2.2 Why This is the "Best" Approach

This architecture is superior to alternatives (such as pure heuristics or monolithic RNNs) for several reasons derived from the research snippets:

1. **Scalability:** By decomposing the problem, the Manager only needs to output a binary decision per node (Visit/Don't Visit), which is an $O(N)$ operation for a GAT. The Transformer then solves a reduced VRP (only visiting, say, 20 out of 100 nodes), which

is significantly faster and more accurate than solving for the full set. This aligns with findings that hierarchical decomposition reduces complexity for large-scale instances.[1]

2. **Temporal Abstraction:** The Manager operates on a longer time horizon (optimizing over T days to prevent overflows), while the Worker optimizes the immediate spatial cost. HRL is explicitly designed to bridge these differing temporal resolutions.[2]
3. **Modularity:** The Manager treats the routing cost as a "black box" signal. If the user upgrades their Transformer to a more efficient version (e.g., EFormer [9] or Dual-Aspect Transformer [15]), the Manager generally does not need to be retrained from scratch, only fine-tuned to the new cost dynamics.

3. The "Manager" Policy: Graph Attention Networks (GAT)

Having established H-DRL as the framework, we must identify the specific deep learning model for the Manager. The research strongly supports the **Graph Attention Network (GAT)** as the state-of-the-art encoder for this role.

3.1 Limitations of Convolutional and Recurrent Networks

Standard deep learning models are ill-suited for the "Manager" role in VRP:

- **CNNs:** Convolutional Neural Networks require grid-structured data (images). Waste bins are scattered irregularly in continuous space. Mapping them to a grid (pixelation) results in loss of precision and sparsity issues.[16]
- **RNNs/LSTMs:** While good for time series, standard RNNs process inputs sequentially. However, the set of waste bins has no inherent order; processing bin #1 then bin #2 implies a dependency that doesn't exist. Graph models are **permutation invariant**, meaning the output is the same regardless of the input order, a critical property for solving VRPs.[7]

3.2 The Graph Attention Mechanism

The GAT is the optimal choice because it explicitly models the **relational structure** of the waste network. The decision to visit bin i is not made in isolation; it depends heavily on the state of its neighbors.

- **Scenario:** Bin A is 90% full. Bin B (located 100m away) is 50% full.
- **Logic:** A myopic policy might visit A and ignore B. A GAT, however, uses the **Attention Mechanism** to compute a coefficient α_{AB} representing the "importance" of node B to node A. The network learns that if it is already committing a vehicle to visit A, the marginal cost of visiting B is low, and thus it should be included to prevent a future trip.
- Mechanism: The GAT updates the embedding of node i (h_i) by aggregating features from its neighbors $j \in \mathcal{N}(i)$, weighted by attention scores:

  $$h_i' = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W} h_j \right)$$

where α_{ij} is a learned function of the compatibility between the features of nodes i and j (e.g., proximity and fill levels).2

3.3 Augmenting GAT with Temporal Encoders (LSTM)

The user's query highlights the availability of "daily waste fill values." To make the Manager proactive rather than reactive, it must understand **accumulation trends**. A bin might be 60% full now, but if it historically fills by 20% per day (high variance), it risks overflowing tomorrow. The best practice is to augment the GAT node features with a **Long Short-Term Memory (LSTM)** encoder.[19]

1. **Input:** For each node i, take the time series of historical fill levels $W_{i, t-k}, \dots, W_{i, t}$.
2. **Processing:** Pass this sequence through a small LSTM to generate a temporal embedding vector $h_{temp, i}$.
3. Fusion: Concatenate this temporal embedding with the static spatial features (X_i, Y_i) to form the input feature vector for the GAT:

   $$x_i =$$

This hybrid **GAT-LSTM** architecture allows the Manager to leverage both spatial clustering (via GAT) and temporal forecasting (via LSTM) to make robust scheduling decisions.[22]

4. State Representation and Feature Engineering

The efficacy of the deep learning model depends entirely on the quality of the state representation fed into it. Based on the analysis of waste management constraints [24], the input graph G_t at day t must contain the following embedded information:

4.1 Node-Level Features ($N \times F$ Matrix)

For every bin node i, the feature vector should include:

- **Static:** X_i, Y_i (Normalized coordinates), Capacity C_i, Service Time (how long it takes to empty).
- **Dynamic:** Current Fill Level W_{it} (normalized to $$), Predicted Accumulation Rate $\hat{r}_{it}$ (output from the LSTM sub-module), and **Age** (time since last visit). The "Age" feature is critical for preventing stagnation—where a remote bin with a slow fill rate is ignored indefinitely by the algorithm.[25]

4.2 Global/Context Features ($1 \times G$ Vector)

The Manager also needs to know the global constraints for the day:

- **Fleet Capacity:** Total remaining capacity of all trucks available on day t.
- **Time Budget:** Remaining working hours (if constraints exist).
- **Calendar Features:** Day of the week (e.g., waste generation might spike on weekends), holidays, or weather indicators that affect travel speed.[27]

4.3 Constraint-Oriented Hypergraphs

Recent advanced research [29] suggests using **hypergraphs** to represent constraints. If multiple bins share a specific constraint (e.g., they must be visited by a specific vehicle type or within a specific time window), they can be connected via a hyperedge. While this adds complexity, it significantly improves the model's ability to learn feasibility rules. For most standard applications, a fully connected graph with GAT attention masking is sufficient.

5. Training Methodologies: Bridging Strategic and Tactical Levels

Training a hierarchical system where one neural network (Manager) provides inputs to another (Transformer) requires specific stabilization techniques.

5.1 The Reinforcement Learning Paradigm: PPO

The Manager must be trained using Reinforcement Learning. The action space (selecting a subset of nodes) is discrete and high-dimensional (2^N). While **Deep Q-Networks (DQN)** have been used for simpler, smaller selection tasks [30], they struggle with the combinatorial explosion of large subsets.

The recommended algorithm is **Proximal Policy Optimization (PPO)**. PPO is an actor-critic method that is more stable and sample-efficient than DQN for this type of problem.

- **Actor:** The GAT outputs a probability distribution over nodes (Bernoulli distribution for each node). PPO optimizes this policy to maximize expected reward.
- **Critic:** A separate GAT (or shared encoder with a separate head) estimates the Value Function $V(s)$, predicting the expected long-term cost from the current state. This helps reduce variance in the gradient updates.[32]

5.2 Reward Shaping and Engineering

The reward function is the most critical component. A naive reward (e.g., just negative distance) will cause the Manager to select *zero* nodes (zero distance). The reward must balance routing efficiency with service quality.[34]

Recommended Reward Structure:

$$R_t = - \left(\text{RouteCost}_t + \lambda_1 \cdot \text{OverflowPenalty}_t + \lambda_2 \cdot \text{FutureRisk}_t \right)$$

- **RouteCost:** The output distance from the Transformer (The Worker). This aligns the Manager's incentives with the Worker's efficiency.
- **OverflowPenalty:** A massive penalty for any bin where $W_{it} > 1.0$.
- **FutureRisk (Reward Shaping):** This is a *dense* reward signal to guide learning. Instead of waiting for an overflow to punish the agent, the system provides a smaller penalty proportional to the *square* of the fill level (W_{it}^2). This encourages the agent to

keep inventory levels low generally, reducing the probability of future overflows. This technique, known as **Potential-Based Reward Shaping**, significantly accelerates convergence.[34]

5.3 Curriculum Learning

Training on the full complexity of a Periodic VRP from scratch is difficult. The agent acts randomly at first, causing massive overflows and receiving confusing negative feedback. Curriculum Learning is the solution.37

1. **Stage 1 (Easy):** Train the Manager on a short horizon (e.g., 2 days) with deterministic demand. The agent learns basic spatial clustering.
2. **Stage 2 (Medium):** Extend the horizon to T days. Introduce stochastic accumulation.
3. Stage 3 (Hard): Introduce complex constraints (time windows, variable fleet size). This progressive difficulty ensures the GAT learns robust feature representations before tackling the full temporal planning problem.

6. Comparison of Deep Learning Models

The following table summarizes the comparative analysis of potential architectures for the "Manager" role, justifying the selection of GAT-based H-DRL.

Architecture Candidate	**State Representation**	**Temporal Handling**	**Pros**	**Cons**	**Suitability**
Monolithic Transformer (e.g., AM, POMO)	Sequence of Nodes	Implicit (via features)	Single model; easy to deploy.	Scales poorly to $N \times T$ horizons; struggles with conflicting objectives (Inventory vs. Routing).	Low (for scheduling)
LSTM / RNN	Time-Series	**Excellent**	Strong at predicting W_i trends.	Poor spatial awareness; cannot cluster nodes geographically.	Medium (as sub-module)
Deep Q-Network (DQN)	Vector / Image	Step-by-step	Simple logic for small spaces.	Action space explosion (2^N subsets); unstable on large graphs.	Low
Graph Attention	**Graph (Nodes/Edges**	Implicit or via LSTM	**Best spatial reasoning**;	Requires RL training;	**High** (The Recommended

Network (GAT)	)		Permutation invariant; scalable.	complex implementation.	"Manager")
Decision Transformer	Sequence of States	Attention-based	Can model long-term dependencies.	Requires massive offline datasets (expert trajectories) which may not exist.	Medium (Promising but immature)

Table 1: Comparative Analysis of Neural Architectures for the Scheduling Manager Role.

7. Implementation Roadmap: Augmenting the Transformer

To practically implement this solution, the following workflow is recommended:

7.1 System Integration

1. **Data Preprocessing:** Ingest IoT data (W_i) and coordinates. Normalize coordinates to $$. Normalize W_i to $$ relative to bin capacity.
2. **Manager Inference:**
    - The GAT-LSTM Manager receives the state graph.
    - It outputs a probability map $P \in ^N$.
    - A threshold (or sampling) is applied to generate the binary mask M_t.
3. **Transformer Execution:**
    - The mask M_t is applied to the input of the Transformer.
    - Ideally, the Transformer uses a **Masked Attention** mechanism where the attention scores for unselected nodes are set to $-\infty$, effectively removing them from the computation without reshaping the tensors.[40]
    - The Transformer outputs the optimal route sequence for the selected subset.
4. **State Transition:**
    - Selected nodes have their W_i reset to 0 (serviced).
    - Unselected nodes have their W_i increased by the stochastic accumulation rate r_i.
    - The system advances to $t+1$.

7.2 Handling "Visit Patterns" and Frequency

While the proposed system is dynamic, real-world operations often prefer consistency (e.g., "Customer X prefers Tuesdays"). To handle this:

- Add **Pattern Embeddings** to the node features (e.g., a one-hot vector indicating preferred days).

- Add a **Consistency Reward** to the RL objective, penalizing the agent if it changes the visit day for a customer too frequently between weeks.[25] This allows the "Manager" to learn a "soft" periodic schedule that is robust but adaptable to emergencies.

7.3 Advanced Feature: The "Simultaneous" Encoder

A recent innovation in DRL for VRP is the **Simultaneous Encoder**.[42] Instead of having two completely separate encoders (one for the Manager GAT, one for the Transformer), the system can share the lower layers.

- The Transformer's encoder (which is powerful) generates node embeddings.
- The Manager uses these embeddings (concatenated with waste info) to make the selection.
- The Worker uses the same embeddings to route.
  This reduces computational overhead and ensures both agents operate on a shared understanding of the topology.

8. Conclusion

The "best" deep learning model to augment a spatial Transformer for the Periodic Capacitated Vehicle Routing Problem is a **Hierarchical Deep Reinforcement Learning (H-DRL)** system. Specifically, the high-level scheduling decision ("which days to route") should be governed by a **Graph Attention Network (GAT)**, enhanced with **LSTM** encoders for temporal waste forecasting.

This conclusion is driven by the fundamental need to decouple the **strategic** problem of inventory management (which requires reasoning about future states, stochastic accumulation, and global overflow risks) from the **tactical** problem of vehicle routing (which requires precise spatial sequencing). The GAT provides the necessary relational inductive bias to effectively cluster nodes based on the dynamic graph state, while the LSTM captures the time-series nature of waste generation.

By adopting this hierarchical approach, trained via **Proximal Policy Optimization (PPO)** with **Curriculum Learning** and **Reward Shaping**, the system transforms the user's existing Transformer from a static route optimizer into a proactive, intelligent agent capable of managing the complex spatio-temporal logistics of modern waste collection. This architecture not only minimizes travel distance but optimizes the service frequency itself, directly addressing the core inefficiencies of traditional periodic routing.

Works cited

1. Hierarchical reinforcement learning in network routing optimization - DiVA portal, accessed December 15, 2025, http://www.diva-portal.org/smash/get/diva2:1955666/FULLTEXT01.pdf
2. Dynamic Siting and Coordinated Routing for UAV Inspection via Hierarchical Reinforcement Learning - MDPI, accessed December 15, 2025, https://www.mdpi.com/2075-1702/13/9/861
3. Hierarchical Deep Reinforcement Learning for Vehicle Routing Problem -

OpenReview, accessed December 15, 2025, https://openreview.net/pdf?id=6G7cF9RNzP
4. A rolling horizon heuristic approach for a multi-stage stochastic waste collection problem - arXiv, accessed December 15, 2025, https://arxiv.org/pdf/2405.14499
5. Inventory routing for dynamic waste collection - https ://ris.utwen te.nl, accessed December 15, 2025, https://ris.utwente.nl/ws/files/5141220/wp_431.pdf
6. Deep Reinforcement Learning for Multi-Truck Vehicle Routing Problems with Multi-Leg Demand Routes - arXiv, accessed December 15, 2025, https://arxiv.org/html/2401.08669v1
7. Graph Transformer with Reinforcement Learning for Vehicle Routing Problem, accessed December 15, 2025, https://www.semanticscholar.org/paper/Graph-Transformer-with-Reinforcement-Learning-for-Fellek-Farid/59ff1d91961dc8ee491cb83d7e2867341cc0f4f0
8. Graph Transformer with Reinforcement Learning for Vehicle Routing Problem | Request PDF - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/368496592_Graph_Transformer_with_Reinforcement_Learning_for_Vehicle_Routing_Problem
9. EFormer: An Effective Edge-based Transformer for Vehicle Routing Problems - IJCAI, accessed December 15, 2025, https://www.ijcai.org/proceedings/2025/954
10. EFormer: An Effective Edge-based Transformer for Vehicle Routing Problems - IJCAI, accessed December 15, 2025, https://www.ijcai.org/proceedings/2025/0954.pdf
11. Hierarchical reinforcement learning in network routing optimization - DiVA portal, accessed December 15, 2025, http://www.diva-portal.org/smash/record.jsf?pid=diva2:1955666
12. Hierarchical Reinforcement Learning for Vehicle Routing Problems with Time Windows, accessed December 15, 2025, https://www.researchgate.net/publication/352724597_Hierarchical_Reinforcement_Learning_for_Vehicle_Routing_Problems_with_Time_Windows
13. Hierarchical reinforcement learning for vehicle routing problems with ..., accessed December 15, 2025, https://nrc-publications.canada.ca/eng/view/ft/?id=e02634fa-53d9-4666-8876-5db877efe04a
14. Hierarchical Deep Reinforcement Learning for Vehicle Routing ..., accessed December 15, 2025, https://openreview.net/forum?id=6G7cF9RNzP
15. Learning to Iteratively Solve Routing Problems with Dual-Aspect Collaborative Transformer - arXiv, accessed December 15, 2025, https://arxiv.org/pdf/2110.02544
16. Investigation of a deep learning-based waste recovery framework for sustainability and a clean environment using IoT - RSC Publishing, accessed December 15, 2025, https://pubs.rsc.org/en/content/articlehtml/2025/fb/d4fb00340c
17. Recurrent Neural Network Based Reinforcement Learning for Inventory Control with Agent-based Supply Chain Simulator | Request PDF - ResearchGate, accessed December 15, 2025,

https://www.researchgate.net/publication/385192401_Recurrent_Neural_Network_Based_Reinforcement_Learning_for_Inventory_Control_with_Agent-based_Supply_Chain_Simulator
18. Integrating Machine Learning and Optimisation to Solve the Capacitated Vehicle Routing Problem - SciTePress, accessed December 15, 2025, https://www.scitepress.org/Papers/2025/131659/131659.pdf
19. Sensor Free Fill-Level Forecasting and AI-Driven (NSGA-II) Route Optimization for Multi-Compartment Waste Collection - DiVA portal, accessed December 15, 2025, http://www.diva-portal.org/smash/get/diva2:1962517/FULLTEXT01.pdf
20. Recurrent neural networks - Dature - Inventory and stock management, accessed December 15, 2025, https://dature.cloud/en/knowledge-base/recurrent-neural-networks/
21. An assign-and-route matheuristic for the time-dependent Inventory Routing Problem, accessed December 15, 2025, https://www.researchgate.net/publication/354806455_An_assign-and-route_matheuristic_for_the_time-dependent_Inventory_Routing_Problem
22. A Spatio-Temporal Graph Convolutional Network for Air Quality Prediction - MDPI, accessed December 15, 2025, https://www.mdpi.com/2071-1050/15/9/7624
23. Spatio-Temporal Graph Neural Networks for Aggregate Load Forecasting - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/364072320_Spatio-Temporal_Graph_Neural_Networks_for_Aggregate_Load_Forecasting
24. The Impact of IoT-Enabled Routing Optimization on Waste Collection Distance: A Systematic Review and Meta-Analysis - MDPI, accessed December 15, 2025, https://www.mdpi.com/2305-6290/9/4/161
25. (PDF) A unified model framework for the multi-attribute consistent periodic vehicle routing problem - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/343409706_A_unified_model_framework_for_the_multi-attribute_consistent_periodic_vehicle_routing_problem
26. Transformer Driven Multi-Agent Reinforcement Learning Framework for Integrated Waste Classification Forecasting and Adaptive Routing - The Science and Information (SAI) Organization, accessed December 15, 2025, https://thesai.org/Downloads/Volume16No11/Paper_74-Transformer_Driven_Multi_Agent_Reinforcement_Learning_Framework.pdf
27. A Deep Reinforcement Learning Model to Solve the Stochastic Capacitated Vehicle Routing Problem with Service Times and Deadlines - MDPI, accessed December 15, 2025, https://www.mdpi.com/2227-7390/13/18/3050
28. Periodic Transformer Encoder for Multi-Horizon Travel Time Prediction - MDPI, accessed December 15, 2025, https://www.mdpi.com/2079-9292/13/11/2094
29. Towards Constraint-Based Adaptive Hypergraph Learning for Solving Vehicle Routing: An End-to-End Solution - arXiv, accessed December 15, 2025, https://arxiv.org/html/2503.10421v1
30. Enhanced vehicle routing for medical waste management via hybrid deep reinforcement learning and optimization algorithms - Frontiers, accessed December 15, 2025,

https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1496653/full
31. Enhanced vehicle routing for medical waste management via hybrid deep reinforcement learning and optimization algorithms - PMC - PubMed Central - NIH, accessed December 15, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11861366/
32. Multi-Agent Deep Reinforcement Learning for Recharging-Considered Vehicle Scheduling Problem in Container Terminals | Request PDF - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/381621894_Multi-Agent_Deep_Reinforcement_Learning_for_Recharging-Considered_Vehicle_Scheduling_Problem_in_Container_Terminals
33. (PDF) SED2AM: Solving Multi-Trip Time-Dependent Vehicle Routing Problem using Deep Reinforcement Learning - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/389648674_SED2AM_Solving_Multi-Trip_Time-Dependent_Vehicle_Routing_Problem_using_Deep_Reinforcement_Learning
34. Reward shaping — Mastering Reinforcement Learning, accessed December 15, 2025, https://gibberblot.github.io/rl-notes/single-agent/reward-shaping.html
35. Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications - arXiv, accessed December 15, 2025, https://arxiv.org/html/2408.10215v1
36. HPRS: hierarchical potential-based reward shaping from task specifications - Frontiers, accessed December 15, 2025, https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1444188/full
37. Graph Pointer Network Based Hierarchical Curriculum Reinforcement Learning Method Solving Shuttle Tankers Scheduling Problem - IEEE Xplore, accessed December 15, 2025, https://ieeexplore.ieee.org/iel8/9420428/10820939/10820942.pdf
38. Curriculum Learning in Genetic Programming Guided Local Search for Large-scale Vehicle Routing Problems - arXiv, accessed December 15, 2025, https://arxiv.org/html/2505.15839v1
39. Curriculum Learning in Genetic Programming Guided Local Search for Large-scale Vehicle Routing Problems | Request PDF - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/392986179_Curriculum_Learning_in_Genetic_Programming_Guided_Local_Search_for_Large-scale_Vehicle_Routing_Problems
40. Solving pickup and drop-off problem using hybrid pointer networks with deep reinforcement learning - ResearchGate, accessed December 15, 2025, https://www.researchgate.net/publication/360877973_Solving_pickup_and_drop-off_problem_using_hybrid_pointer_networks_with_deep_reinforcement_learning
41. A new vehicle routing problem for increased driver-route familiarity - SUnORE, accessed December 15, 2025,

https://sunore.co.za/wp-content/uploads/2024/03/King-JCP-2023.pdf
42. SED2AM: Solving Multi-Trip Time-Dependent Vehicle Routing Problem using Deep Reinforcement Learning - arXiv, accessed December 15, 2025, https://arxiv.org/pdf/2503.04085