Computers & Industrial Engineering 220 (2026) 112282

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Scalable knowledge-guided framework for large multi-depot vehicle routing☆

Narges Movahedkor [a,1], Reza Shahbazian [b,2], Francesca Guerriero [a,*,3]

[a] Department of Mechanical, Energy and Management Engineering (DIMEG), University of Calabria, 87036 Rende (CS), Italy
[b] Department of Humanities, University of Palermo, 90123 Palermo, Italy

ARTICLE INFO

Dataset link: https://github.com/ShahbazianR/Scalable-KGCAF.git

Keywords:
Multi-depot vehicle routing problem (MDVRP)
Graph attention networks (GAT)
Graph neural networks (GNN)
Multi-agent reinforcement learning (MARL)
Hybrid optimization

ABSTRACT

Solving the vehicle routing problem (VRP) in large-scale, multi-depot logistics is complex due to the intricate trade-offs between computational scalability and constraint satisfaction While deep reinforcement learning (DRL)-based approaches seem to be promising, they often struggle with the logical feasibility and interpretability required for real-world operations. We propose the Knowledge-Guided Collaborative Attention Framework (KGCAF), which combines depot-level profile-based decomposition with feasibility-aware graph attention and hybrid heuristic refinement. By employing a depot-level profile decomposition, it enables localized and parallelizable route construction while ensuring logical feasibility through explicit masking. KGCAF unifies learning-based initialization with time-budgeted heuristic optimization. Evaluations on multi-depot VRPTW benchmarks show KGCAF reduces total distance by up to 86%, cuts average time window violations from over 1000 to under 100, and outperforms decomposition-only or graph neural network (GNN)-only baselines in both sequential and parallel executions. These results demonstrate competitive or superior solution quality, strict feasibility adherence, and scalable performance, highlighting the framework's practicality for large-scale, real-world multi-agent routing.

1. Introduction

The evolution of intelligent transportation systems and autonomous logistics has introduced new paradigms to vehicle routing problems (VRPs). In the context of fleet management, last-mile delivery, and electric mobility, optimized vehicle routing remains a fundamental and complex challenge, involving intricate trade-offs among travel cost, energy consumption, and emissions, while respecting time windows and operational constraints (Ajao & Oludamilare, 2023; Guerriero et al., 2025; Movahedkor et al., 2026; Zhu et al., 2023). These challenges are further amplified for electric vehicle (EV) fleets due to limited charging infrastructure and battery capacity, which necessitate adaptive, knowledge-driven routing strategies.

At its core, the VRP is a classical NP-hard combinatorial optimization problem, whose complexity becomes significantly more pronounced with the introduction of temporal constraints, leading to the Vehicle Routing Problem with Time Windows (VRPTW). Classical and metaheuristic approaches, including Adaptive Neighborhood Search, Tabu Search, and GRASP, have demonstrated strong performance in this setting (Bezerra, Souza, & de Souza, 2023). However, their sensitivity to initialization and limited adaptability in real-time environments restrict their practical effectiveness. In contrast, deep learning (DL) and reinforcement learning (RL) offer flexible, data-driven alternatives for routing optimization (Hildebrandt et al., 2023; Nazari et al., 2018; Shahbazian et al., 2025), yet their scalability, training overhead, and lack of interpretability continue to hinder widespread deployment (Bogyrbayeva et al., 2023). To address these limitations, knowledge-guided learning has emerged as a promising direction, explicitly encoding environmental constraints and dynamics to guide exploration and accelerate convergence (Eßer et al., 2022; Mazumder et al., 2022). Within this paradigm, naturally distributed decision-making can be achieved under uncertainty-driven and dynamically evolving conditions (Zhang et al., 2021).

In this work, we introduce a novel framework for large-scale multi-agent routing, termed the Knowledge-Guided Collaborative Attention

☆ This article is part of a Special issue entitled: 'Operations Research and AI in logistics' published in Computers & Industrial Engineering.

* Correspondence to: Via P. Bucci, Cubo 42C 87036 Rende (CS), Italy.
E-mail addresses: narges.movahed@unical.it (N. Movahedkor), reza.shahbazian@unipa.it (R. Shahbazian), Francesca.guerriero@unical.it (F. Guerriero).
[1] Researcher, Artificial Intelligence in Logistics.
[2] Assistant Professor (Tenure Track), Artificial Intelligence (Computer Eng.).
[3] Full Professor, Operations Research.

https://doi.org/10.1016/j.cie.2026.112282

Available online 4 August 2026

Framework (KGCAF). The proposed approach integrates graph attention-based representation learning with adaptive, lightweight heuristic search under a profile-based decomposition strategy. Each depot is represented through a profile embedding that captures spatiotemporal structure and intra-depot coordination patterns. These localized representations enable decentralized decision-making, parallelizable heuristic refinement, and constraint-aware routing policies via knowledge-guided node masking. Unlike prior approaches that employ collaborative attention across all agents globally, KGCAF emphasizes structured intra-depot coordination informed by shared domain knowledge, resulting in a hybrid framework that tightly integrates learning, reasoning, and heuristic optimization in a scalable and interpretable manner.

While the MDVRP is inherently a deterministic combinatorial optimization problem, it can be naturally reformulated as a sequential decision-making process to leverage the strengths of RL. In our framework, the route construction phase is modeled as a Markov Decision Process (MDP), where each decision step involves selecting the next customer to visit from the current vehicle location. The state space captures the evolving route progress, remaining vehicle capacity, unvisited customers, and time-window constraints; the action space consists of feasible next-customer selections; and the reward function penalizes travel distance and time-window violations. This sequential decomposition transforms the static routing problem into a series of interdependent decisions, enabling a learned policy to construct complete routes incrementally while respecting all operational constraints. By embedding logical feasibility rules directly into the action masking mechanism, we ensure that the sequential decisions always remain within the feasible solution space, thereby guaranteeing that the resulting routes are valid for the original deterministic MDVRP.

In summary, our main contributions are:

1. Knowledge-Guided Multi-Agent Routing: Embeds logical constraints into route construction via node masking, ensuring feasibility and interpretability.
2. Hierarchical, Profile-Based Decomposition: Uses depot-specific profiles to capture relational and temporal knowledge, enabling scalable intra depot optimization and inter-depot coordination.
3. Hybrid Learning and Optimization: Unifies graph attention reasoning with adaptive heuristics, improving efficiency and solution quality.
4. Parallelizable Multi-Agent Execution: Supports asynchronous, profile-synchronized optimization across depots for real-time performance.

Extensive experimentation on state-of-the-art (SOTA) baselines, including a range of (meta)-heuristics, graph RL, and Deep RL methods, has demonstrated improvements across multiple dimensions of route quality, cost, time feasibility, and computational efficiency, thereby validating the practical advantages of KGCAF. To assess the contribution of each architectural component, we conduct an extensive ablation study isolating the impacts of our profile-based decomposition and the learning-based GAT initialization. Furthermore, we perform a comparative analysis of Local (depot-specific) versus Global (depot-flexible) heuristic refinement paradigms, examining their distinct interactions with the structured solution space learned by the framework. Our evaluation integrates a comprehensive suite of heuristic operators, including greedy, tabu, GRASP, and simulated annealing strategies, alongside a state-of-the-art Enhanced Adaptive Large Neighborhood Search, demonstrating how the synergy between neural initialization and structured search consistently discovers new best-known solutions across established benchmarks. Overall, the framework uniting profile-based decomposition with knowledge-guided learning and heuristic search will further advance real-world vehicle routing toward cooperative, scalable multi-agent intelligence development.

The remainder of this paper is structured to guide the reader through our methodology and findings. We begin by reviewing the relevant literature in Section 2, focusing on the intersection of artificial intelligence (AI) and traditional optimization. Section 3 details the architecture of our KGCAF framework and the logic-guided masking process. In Section 4, we describe our experimental setup and present a comparative analysis of our results against current benchmarks. Finally, Section 5 presents a detailed discussion on benefits and limitations of the proposed framework, and Section 6 concludes the paper with a discussion on the practical implications of our work and potential directions for future research.

2. Related literature

VRP research has evolved through exact optimization, heuristic/metaheuristic search, and learning-based paradigms (Bouanane et al., 2022; Liu et al., 2023). While exact methods guarantee optimality (Sluijk et al., 2023), they are intractable for large or dynamic instances (Macrina et al., 2017). In particular, their centralized formulations and exponential complexity make them unsuitable for real-time or large-scale multi-depot settings, where frequent state updates and partial observability are required. Heuristics like variable neighborhood search, Tabu search, and genetic algorithms, offer greater efficiency but rely on manual rule design and adapt poorly to dynamic environments (Zhang et al., 2024; Zhou et al., 2025). Moreover, such methods typically encode feasibility as hard-coded procedures external to the optimization logic, limiting transferability and interpretability when problem conditions change.

RL provides a data-driven alternative for learning routing policies (Hildebrandt et al., 2023; Nazari et al., 2018). Initial efforts tackled static, single-depot problems, later adding attention mechanisms (Nazari et al., 2018) or stochastic models (Pan & Liu, 2023), though scalability and solution feasibility remain challenging. In these approaches, feasibility is often enforced implicitly via penalties or post-hoc repair, which complicates training and provides no guarantees during inference. Graph-based RL methods, such as (Zhang et al., 2023), have been applied to multi-depot VRPs but often lack structured feasibility guidance, limiting interpretability and practical deployment. Although graph representations capture relational structure, constraints are typically learned implicitly in embeddings rather than enforced explicitly at decision time. Moreover, multi-agent RL enables decentralized coordination but struggles with training instability and real-time efficiency (Zhang et al., 2021), particularly when agent interactions are mediated only through shared rewards rather than structured state abstractions.

Hybrid learning-and-search methods aim to combine the strengths of RL and heuristics. Examples include DRL with Monte Carlo Tree Search (Alcaraz et al., 2022), actor-critic RL with Large Neighborhood Search (Zhao et al., 2020), bi-level hybrid search for arc routing (Zhou et al., 2024), and multi-objective RL for time-dependent pickup and delivery (Santiyuda et al., 2024). While effective, these approaches typically treat learning and search as loosely connected modules, lacking a unified and interpretable coupling. In most cases, neural policies generate initial solutions while heuristics operate as black-box post-processors, without access to learned representations or domain knowledge encoded during training. This separation limits its mutual reinforcement between global policy learning and local optimization.

Therefore, three key limitations persist: (i) an oversimplification of real-world logistics, often ignoring multi-depot and dynamic constraints; (ii) insufficient integration of domain knowledge into neural architectures for feasibility guidance; and (iii) a weak coupling between global policy learning and local heuristic exploitation. Existing graph-based or hybrid methods address these issues only partially, either by increasing model complexity without explicit constraint handling or by relying on external heuristics without principled integration.

In this respect, we propose the KGCAF, addressing these gaps by embedding logical feasibility rules directly into the route construction mechanism and employing depot-profile-based decomposition for

Table 1
Comparison of representative VRP solution paradigms and learning-based approaches with the proposed KGCAF framework.

Method	Paradigm	VRP Setting	Dynamic	Depot	Feasibility handling	Learning–search coupling	Scalability	Interpretability
Exact Optimization (Sluijk et al., 2023)	Exact	Static	No	S	Explicit mathematical constraints	N/A	Low	High
Meta/Heuristics (Zhang et al., 2024; Zhou et al., 2025)	Heuristic	Static/Limited dynamic	Limited	M	Handcrafted rules	None	Medium	Medium
RL (Nazari et al., 2018; Pan & Liu, 2023)	RL	Static, single-depot	Limited	S	Implicit (penalty-based)	Policy-only	Medium	Low
Graph-RL (Zhang et al., 2023)	Graph RL	Multi-depot	Limited	M	Implicit	Weak	Medium	Low
Multi-Agent RL (Zhang et al., 2021)	MARL	Decentralized VRP	Yes	M	Implicit	Policy coordination	Low–Medium	Low
RL + MCTS (Alcaraz et al., 2022)	Hybrid	Online VRP	Yes	S	Search-time pruning	Loose	Low	Low
RL + LNS (Zhao et al., 2020)	Hybrid	Static VRP	No	M	Heuristic repair	Loose	Medium	Medium
Bi-level Hybrid Search (Zhou et al., 2024)	Hybrid	Arc Routing	No	M	Search constraints	Hierarchical	Medium	Medium
Multi-objective RL (Santiyuda et al., 2024)	RL	Time-dependent VRP	Yes	M	Penalty-based	Policy-only	Medium	Low
KGATAVNS (Movahedkor et al., 2026)	Hybrid	Dynamic EVRP	Yes	M	Region-based masking	Loose	Low	Medium
KGCAF (Ours)	Knowledge-Guided Hybrid	Dynamic Multi-Depot VRP	Yes	M	Embedded logical rules	Unified and tight	High	High

scalable optimization. Unlike prior graph RL or hybrid methods, KGCAF unifies knowledge-guided representation learning and heuristic refinement in a single, interpretable loop, enabling constraint-aware and parallelizable routing for large-scale multi-depot VRPs. Specifically, depot-level profile abstraction provides a structured state representation that supports decentralized coordination, while logic-guided attention ensures that both neural inference and heuristic refinement operate exclusively within the feasible solution space. This tight integration allows KGCAF to scale to dynamic, multi-depot settings while maintaining interpretability, feasibility guarantees, and computational efficiency.

A closely related hybrid approach is the KGATAVNS framework proposed in Movahedkor et al. (2026), which explicitly modeled energy consumption for dynamic multi-depot electric vehicle routing. While effective at establishing a physics-informed baseline, KGATAVNS utilized a static, pre-processed nearest-depot assignment and handled domain knowledge through simple region-based masking. Consequently, it faced scalability limitations and lacked the explicit logical predicates necessary for evaluating complex operational rules. To address these critical gaps, our proposed KGCAF framework introduces a fully dynamic, modular architecture with a unified and tight learning-search coupling. We replace static region assignments with a dynamic, profile-based decomposition strategy that maintains continuously updated states for each depot. Furthermore, KGCAF transcends simple region masking by embedding a comprehensive logical predicate component. This layer dynamically evaluates a conjunction of feasibility rules in real time, strictly pruning infeasible actions during encoding and decoding.

Table 1 summarizes the related works and compares them with the proposed KGCAF framework.

3. Proposed framework

Our proposed KGCAF framework integrates multi-agent settings, reinforcement learning, knowledge-guided graph attention reasoning (KGAT), and quick heuristic search under a depot-level decomposition strategy. Fig. 1 illustrates the overall view of the proposed framework.

The KGCAF architecture proceeds sequentially from the problem inputs through a hierarchical pipeline. First, a profile-based decomposition partitions the global problem into localized, depot-specific sub-problems with explicit feasibility rules. Within each depot, a GAT-based route initialization stage constructs feasible initial solutions, guided by a sub-component for feasible graph formation that enforces logical constraints at the attention level, directly interacting with the environment. The environment consequently updates the knowledge-base, which directs the route construction in the knowledge-guided GAT via the knowledge integration module. The produced initial routes are then enhanced through a heuristic refinement phase that performs time-budgeted local search under the same feasibility predicates, ultimately producing the final set of optimized, constraint-satisfying routes.

Before proceeding, we define two concepts that are central to our proposed framework:

- *Logical feasibility*: refers to the set of hard, non-negotiable constraints defined by the problem, such as vehicle capacity limits, customer time windows, and depot regional assignments, that any valid solution must satisfy. Unlike penalty-based methods, which allow these constraints to be violated and later corrected, KGCAF treats them as Boolean predicates that are evaluated at each decision step.
- *Explicit masking*: is the computational mechanism by which logical feasibility is enforced. At each step of the route construction, the framework algorithmically evaluates all candidate next customers against the current vehicle state. Any action that would result in a constraint violation is assigned a logit of $-\infty$ (approximated by a large negative constant, $M = 10^6$), which, after a Softmax operation, yields a selection probability of strictly zero. This guarantees that the neural network and the subsequent heuristic search operate exclusively within the feasible solution space.

3.1. Problem statement

We address the capacitated VRP with time windows in a multi-depot, dynamic, large-scale multi-agent setting. The goal is to coordinate multiple vehicle agents per depot to construct feasible and cost-efficient routes while respecting capacity, time-window constraints, and knowledge-guided processing.

Generally, the problem is defined as below:

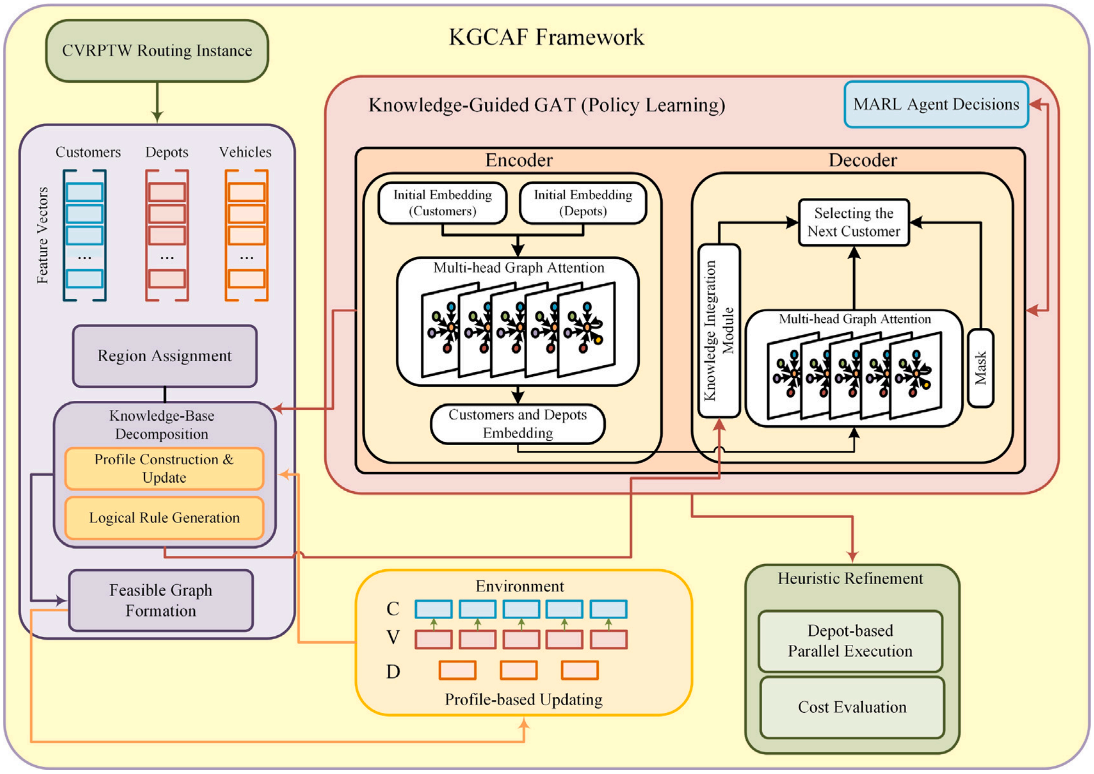

Fig. 1. Schematic overview of the Knowledge-Guided Collaborative Attention Framework (KGCAF). The architecture integrates a high-level Depot-level Profile Decomposition for scalable state abstraction with a Logics-Guided Graph Attention Network (KGAT).

Let $\mathcal{D} = \{1, \dots, |\mathcal{D}|\}$ be the set of depots and $\mathcal{C} = \{1, \dots, |\mathcal{C}|\}$ the set of customers. Each depot d has a set of homogeneous fleet of vehicles $\mathcal{K}_d \in \mathcal{K}$, each with capacity Q. Each customer $c_i \in \mathcal{C}$ has coordinates (x_i, y_i), demand q_i, service time s_i, and time window $[e_i, \ell_i]$ within which service must begin. Travel costs between nodes $i, j \in \mathcal{V} = \mathcal{D} \cup \mathcal{C}$ are $t_{ij} \geq 0$, with all routes starting and ending at the depot. A route $r \in \mathcal{R} = \cup_{d \in \mathcal{D}} \cup_{k \in \mathcal{K}} \mathcal{R}_{kd}$ is an ordered sequence of customers starting and ending at depot, and a feasible solution $S = \{r_{k,d} | d \in \mathcal{D}, k \in \mathcal{K}_d\}$ covers all customers exactly once while satisfying problem constraints. Let $x_{ij}^{kd} \in \{0,1\}$ $\forall i, j \in \mathcal{V}, d \in \mathcal{D}, k \in \mathcal{K}_d$ equals 1 if a vehicle k of depot d travels from i to j, $x_{ij}^{r} \in \{0,1\}$ equals 1 if a route $r \in \mathcal{R}$ travels from i to j, where $x_{ij}^{kd} = \sum_{r \in \mathcal{R}_{kd}} x_{ij}^{r}$ $\forall i, j \in \mathcal{V}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K}$. Finally, a_j denotes the arrival time at customer j.

We aim to minimize a weighted combination of travel cost and temporal penalties. The objective is defined as:

$$\min \mathrm{Cost}(S) = \gamma \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} t_{ij} x_{ij}^{kd} + \delta \sum_{j \in \mathcal{C}} \pi_j(a_j),$$

with

$$\pi_j(a_j) = \begin{cases} \alpha(e_j - a_j), & a_j < e_j, \\ \beta(a_j - \ell_j), & a_j > \ell_j, \\ 0, & e_j \leq a_j \leq \ell_j, \end{cases}$$

where $\gamma, \delta, \alpha, \beta \geq 0$ are scaling weights for distance-time window violation objectives and penalty rates for time-window violation, satisfying the following constraints:

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} \sum_{j \in \mathcal{V}, j \neq i} x_{ij}^{kd} = 1, \qquad \forall i \in \mathcal{C}$$

$$\sum_{i \in \mathcal{V}, i \neq h} x_{ih}^{kd} - \sum_{j \in \mathcal{V}, h \neq j} x_{hj}^{kd} = 0, \qquad \forall h \in \mathcal{C}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d$$

$$\sum_{i \in \mathcal{C}} q_i \sum_{j \in \mathcal{V}, j \neq i} x_{ij}^{r} \leq Q, \qquad \forall r \in \mathcal{R}$$

$$a_i + s_i + t_{ij} - M(1 - x_{ij}^{kd}) \leq a_j, \qquad \forall i, j \in \mathcal{V}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d.$$

These constraints enforce coverage, flow conservation, and capacity adherence, providing a rigorous foundation for hybrid learning-heuristic routing methods with guaranteed feasibility. We aim to develop a scalable, logic-guided framework that ensures strict constraint satisfaction and high-quality routing solutions for large-scale, multi-depot logistics systems.

Remark on solution methodology. While the mixed-integer programming (MIP) formulation in the equations above formally defines the optimal solution to the MDVRPTW, solving it to optimality for large-scale instances is computationally prohibitive due to its NP-hard nature. Consequently, this formulation is not employed directly as an exact solver in our experiments. Instead, it serves two fundamental roles in the design of the proposed KGCAF framework. First, the objective function components γ, δ and the penalty function π_j directly inform the design of the reward function used to train the neural policy. Second, the logical constraints on capacity, flow, and time windows are algorithmically encoded into the feasibility-aware masking and profile-based rules that guarantee all generated routes are strictly feasible. Thus, the MIP model provides a rigorous formal specification that is operationally translated into a scalable, learning-based solver.

3.2. Profile-based problem decomposition

To handle large-scale multi-depot routing problems, we adopt a *profile-based depot decomposition* that maintains structured depot-level knowledge for logically feasible and interpretable decision-making. By capturing inter-dependencies among customers, vehicles, and depots, it enables scalable, real-time coordination and route optimization.

Definition 1 (*Depot Profile Abstraction*)**.** Let $\mathcal{C}$ be the set of customers and $\mathcal{D}$ the set of depots. Each depot $d \in \mathcal{D}$ is assigned $\mathcal{C}_d \subseteq \mathcal{C}$ such that $\mathcal{C} = \bigcup_{d \in \mathcal{D}} \mathcal{C}_d$ and $\mathcal{C}_d \cap \mathcal{C}_{d'} = \emptyset$ for $d \neq d'$. Depot d maintains a profile $\mathbf{p}_d^{(t)}$ at time t:

$$\mathbf{p}_d^{(t)} = (\mathbf{X}_d^{(t)}, \mathbf{V}_d^{(t)}, \mathbf{s}_d^{(t)}), \tag{1}$$

where $\mathbf{X}_d^{(t)} = \{\mathbf{x}_j^{(t)} \mid j \in C_d\}$, $\mathbf{V}_d^{(t)} = \{\mathbf{v}_k^{(t)} \mid k \in \mathcal{K}_d\}$, $\mathbf{x}_j^{(t)} = (\text{loc}_j, q_j, s_j, [e_j, \ell_j], \mathbb{I}^j_{\text{visited}})$, $\mathbf{v}_k^{(t)} = (\ell_t^k, Q^k_{\text{rem}}, \tau_t^k)$, and $\mathbf{s}_d^{(t)} = (\mathcal{R}_d, Q_{\text{residual}}, \dots)$, capturing customer features, vehicle states, and depot-level information such as region assignments, residual capacity, and route progress.

Here, $t \in \{0, 1, 2, \dots\}$ denotes a discrete decision step indexing the sequential route construction process, distinct from the continuous physical time governing time windows and arrival times.

Profile update dynamics. At each discrete step t, depot profiles update based on local and inter-depot events:

$$\mathbf{p}_d^{(t+1)} = f_{\text{update}}\Big(\mathbf{p}_d^{(t)}, \Delta\mathbf{X}_d^{(t+1)}, \Delta\mathbf{V}_d^{(t+1)}, \{\mathbf{p}_{d'}^{(t)} : d' \neq d\}\Big), \tag{2}$$

where $\Delta\mathbf{X}_d^{(t+1)}$ and $\Delta\mathbf{V}_d^{(t+1)}$ denote local changes. Given event $\mathcal{E}^{(t)}$, updates follow $\mathbf{X}_d^{(t+1)}[j] \leftarrow F_c(\mathbf{X}_d^{(t)}[j], \mathcal{E}^{(t)})$, $\mathbf{V}_d^{(t+1)}[k] \leftarrow F_v(\mathbf{V}_d^{(t)}[k], \mathcal{E}^{(t)})$, and $\mathbf{s}_d^{(t+1)} \leftarrow F_d(\mathbf{s}_d^{(t)}, \mathcal{E}^{(t)})$. Changes are propagated asynchronously via messages containing only modified entries to reduce communication:

$$\mathbf{p}_{d'}^{(t+\delta)} \leftarrow \mathbf{p}_{d'}^{(t+\delta)} \cup \text{msg}(\Delta\mathbf{p}_d^{(t)}).$$

Logical feasibility rules. From $\mathbf{p}_d^{(t)}$, depot d derives logical predicates $\Phi_d = \{\phi_n^d\}_{n=1}^N$, where $\phi_n^d : (\mathcal{K}_d, C_d, \mathbf{p}_d^{(t)}) \to \{\text{True}, \text{False}\}$, mapping agents and customers to boolean feasibility. Examples include:

$$\phi^d_{\text{cap}}(i,j) : Q^k_{\text{rem}} \geq q_j, \qquad \phi^d_{\text{region}}(i,j) : j \in \mathcal{R}_d,$$
$$\phi^d_{\text{tw}}(i,j) : a_j + s_j + t_{ij} \leq \ell_j, \qquad \phi^d_{\text{visit}}(i,j) : \mathbb{I}^j_{\text{visited}} = 0,$$

for agent $k \in \mathcal{K}_d$ and customer $j \in C_d$.

Feasible action set. The feasible action set for the agent k is defined as:

$$\mathcal{F}_k^t = \Big\{ j \in C_d \;\Big|\; \bigwedge_{n=1}^N \phi_n^d(i, j; \mathbf{p}_d^{(t)}) = \text{True} \Big\}. \tag{3}$$

Strict masking ensures infeasible actions are prohibited: $\Pr(a_t^i = j \mid \mathbf{p}_d^{(t)}) = -M$ $(\forall j \notin \mathcal{F}_k^t)$, where M is a large positive number (in practice we use $M = 10^6$ to ensure numerical stability). In other words $\tilde{u}_{i,j} = u_{ij}$ if $j \in \mathcal{F}_k^t$, o.w. $\tilde{u}_{i,j} = -M$.

Inter-depot synchronization. Depots maintain environment consistency via asynchronous synchronization:

$$\mathbf{p}_d^{(t+1)} \leftarrow f_{\text{sync}}\big(\mathbf{p}_d^{(t+1)}, \{\mathbf{p}_{d'}^{(t)}\}_{d' \neq d}\big),$$

ensuring coherent regional boundaries and shared-customer handling.

In summary, profile-based decomposition enables: (i) structured depot-level state abstraction, (ii) rule-based feasibility enforcement, (iii) asynchronous consistency maintenance, and (iv) scalable decentralized optimization for multi-depot VRPTW.

These are enabled through a two-stage process. First, the logical predicates Φ_d, derived directly from the profile $\mathbf{p}_d^{(t)}$, function as a constraint firewall, strictly masking all infeasible actions so that only logically valid next customers can be selected. This provides the rule-based feasibility enforcement. Second, because each depot's state is fully captured by its own profile and its updates depend only on local events and lightweight synchronization messages, the computationally intensive steps — GAT-based route construction and heuristic refinement — can execute independently for each depot. This naturally supports structured state abstraction, asynchronous consistency maintenance, and scalable decentralized optimization, as the global problem is decomposed into parallel, loosely coupled sub-problems.

3.3. Modeling the problem

Inspired by Hua et al. (2025), we model each depot $d \in \mathcal{D}$ as a cooperative multi-agent system, where each agent $k \in \mathcal{K}_d$ corresponds to a vehicle constructing a route within its assigned region.

Agents and action selection. At time t, agent k selects the next customer $c_t^k \in \mathcal{F}_k^t \subseteq C_d$ according to a stochastic policy π_θ:

$$c_t^k = \arg\max_{j \in \mathcal{F}_k^t} \pi_\theta(c_t^k = j \mid s_t^k, \mathbf{p}_d^{(t)}),$$

where s_t^k is the agent state and $\mathbf{p}_d^{(t)}$ is the depot profile. The feasible set $\mathcal{F}_k^t$ enforces depot-level constraints.

State space. Each agent's state is $s_t^k = (\mathbf{v}_t^k, \mathbf{N}_t^d)$, where vehicle state $\mathbf{v}_t^k = (\ell_t^k, Q^k_{\text{rem}}, \tau_t^k)$ and node states $\mathbf{N}_t^d = \{\mathbf{n}_j^t \mid j \in C_d\}$ with

$$\mathbf{n}_j^t = (x_j, y_j, q_j, s_j, e_j, \ell_j, \mathbb{I}^j_{\text{visited}}, \Delta t_{kj}).$$

Here, ℓ_t^k is the current location, Q^k_{rem} the remaining capacity, τ_t^k elapsed route time, (x_j, y_j) coordinates, q_j demand, $[s_j, e_j, \ell_j]$ service and time window, $\mathbb{I}^j_{\text{visited}}$ visitation indicator, and Δt_{kj} estimated travel time. This representation provides agents full local and vehicle context for informed routing decisions.

Action space. The feasible set for agent k is:

$$\mathcal{F}_k^t = \Big\{ j \in C_d \mid q_j \leq Q^k_{\text{rem}},\ a_i + s_i + t_{ij} \leq \ell_j,\ \bigwedge_{n=1}^N \phi_n^d(i, j; \mathbf{p}_d^{(t)}) \Big\},$$

where i denotes the current node of vehicle k, $Q^k_{\text{rem}} = Q - \sum_{m \in r_k^{(t)}} q_m$, $r_k^{(t)}$ is the partial route of vehicle k at time t, and ϕ_n^d are logical depot-level constraints (e.g. capacity, time-window, regional feasibility).

Reward function. The immediate reward for taking action $c_t^k = j$ is:

$$r_t^k = -\gamma t_{ij} - \delta \pi_j(a_j) \tag{4}$$

with t_{ij} travel cost and $\gamma, \delta > 0$ balancing distance and temporal penalties.

Policy learning. Each agent learns a stochastic policy π_θ that maps its local state and the depot-level context to a probability distribution over feasible next customers:

$$\pi_\theta(c_t^k = j \mid s_t^k, \mathbf{p}_d^{(t)}) = \Pr(c_t^k = j \mid s_t^k, \mathbf{p}_d^{(t)}), \tag{5}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_t^k, c_t^k \sim \pi_\theta}\big[\nabla_\theta \log \pi_\theta(c_t^k \mid s_t^k, \mathbf{p}_d^{(t)})\, \hat{A}_t^k\big], \tag{6}$$

where $\hat{A}_t^k$ is the advantage estimate. Agents share route states and visitation information within the depot for coordinated decision-making.

3.4. GAT-based route initialization

Each depot d is represented as a local graph $G_d = (V_d, E_d)$, where $V_d = \{d\} \cup C_d$ and E_d encodes feasible travel connections under spatial, temporal, and regional constraints. Node features are

$$\mathbf{x}_i = [x_i, y_i, \mathbb{I}_{i=d}, e_i, \ell_i, s_i, q_i], \quad i \in \mathcal{V}_d,$$

with (x_i, y_i) as coordinates, $\mathbb{I}_{i=d}$ indicating depot nodes, $[e_i, \ell_i]$ the time window, s_i service duration, and q_i demand.

3.4.1. Encoder-decoder framework

Route construction is a sequential decision process parameterized by θ, with probability of a feasible route π_d:

$$p_\theta(\pi_d \mid I_d) = \prod_{t=1}^{T_d} p_\theta(\pi_t^d \mid I_d, \pi^d_{1:t-1}), \tag{7}$$

where T_d is the number of decisions. An encoder–decoder framework generates high-dimensional embeddings and constructs routes under knowledge-informed feasibility.

3.4.2. Encoder: Logic-guided graph attention network

Node embeddings are initialized via a linear transformation:

$$h_i^{(0)} = W^{(0)}\mathbf{x}_i + b^{(0)}. \tag{8}$$

For each GAT layer, attention is computed over feasible neighbors $j \in \mathcal{N}_i^{(\phi)}$ defined by depot-level constraints Φ_d:

$$\alpha_{ij} = \frac{\exp(e_{ij}) \cdot \mathbb{I}[\Phi_d = 1]}{\sum_{k\in\mathcal{N}_i^{(\phi)}} \exp(e_{ik})}, \tag{9}$$

$$h_i' = \sum_{j\in\mathcal{N}_i^{(\phi)}} \alpha_{ij} W^V h_j. \tag{10}$$

Multi-head attention with P heads and residual connections with batch normalization enhance expressiveness:

$$h_i' = \sum_{p=1}^{P} W_p^O h_{i,p}', \tag{11}$$

$$\hat{h}_i = \mathrm{BN}(h_i + \mathrm{MLP}_1(h_i')), \tag{12}$$

$$h_i = \mathrm{MLP}_2(\hat{h}_i + \mathrm{ReLU}(\hat{h}_i)). \tag{13}$$

After Γ layers, $h_i^{(\Gamma)}$ is the final node embedding. The representation of the depot d is then aggregated to the customers' information through the following equation, resulting in the final output embedding of the encoder component:

$$\overline{h_d} = \frac{h_d^{(\Gamma)} + \sum_{i\in C} h_i}{N+1} \tag{14}$$

While the proposed encoder enforces feasibility through constraint-based neighborhood selection and attention masking, feasibility knowledge can be further integrated directly into the embedding process to produce dynamic, context-aware node representations. This can be achieved by conditioning node embeddings not only on static features but also on depot-level profiles and logical predicates that capture capacity, temporal, and regional constraints. In practice, feasibility signals derived from $\mathbf{p}_d^{(t)}$ and Φ_d can be injected as additional node attributes or used to modulate attention parameters, allowing embeddings to evolve as the feasibility landscape changes during route construction. Moreover, logical predicates can act as message-level gates within the attention mechanism, ensuring that information propagation is explicitly governed by constraint satisfaction rather than implicit learning. Such integration yields embeddings that encode both relational structure and feasibility semantics, enabling the encoder to reflect dynamic operational states and providing a principled bridge between symbolic constraints and representation learning.

3.4.3. Decoder: Feasibility-aware route construction

Routes are incrementally constructed for agent $k \in \mathcal{K}_d$ using embeddings $h_i^{(\Gamma)}$, the embedding of the nodes in time step $(t-1)$ (h_c), customer vectors (v_c), and depot profile $\mathbf{p}_d^{(t)}$. The decoder calculates a new context vector embedding representation h_c' of the candidate customer vectors v_c' using $\mathcal{L}$ layers of a multi-head graph attention network. Then, the compatibility scores are calculated as:

$$u_{i,t}^k = \tanh\left(\frac{(W^Q h_c')^T (W^K v_c')}{\sqrt{H_D}}\right) \cdot C, \tag{15}$$

with learnable matrices $W^Q, W^K \in \mathbb{R}^{H_D, d_h}$, embedding dimension d_h, head depth H_D, and clipping constant C. Constraints are enforced through:

$$\tilde{u}_{i,t}^k = \begin{cases} u_{i,t}^k, & \mathcal{M}_t^{(v)}(i) \wedge \mathcal{M}_t^{(\phi)}(i), \\ -M(\approx -\infty), & \text{otherwise,} \end{cases} \tag{16}$$

$$p_{i,t}^k = \log\mathrm{Softmax}(\tilde{u}_{i,t}^k), \quad c_{t+1}^k = \arg\max_i p_{i,t}^k, \quad c_{t+1}^k \in \mathcal{F}_k^t, \tag{17}$$

where $\mathcal{M}_t^{(v)}$ masks visited or infeasible nodes and $\mathcal{M}_t^{(\phi)}$ enforces depot-level logical feasibility. After each selection, the environment updates $\mathbb{I}_{\text{visited}}^j$, Q_{rem}^k, depot profiles $\mathbf{p}_d^{(t)}$, and feasible sets $\mathcal{F}_k^{t+1}$. The GAT-based initialization integrates depot profile $\mathbf{p}_d^{(t)}$ directly into feasibility-aware attention, providing informed, intra-depot route construction without cross-depot coordination and serving as a high-quality starting point for subsequent heuristic refinement. The complete GAT-based initialization procedure for a single depot is formalized in Algorithm 1.

Algorithm 1 Logic-Guided GAT Route Construction for a Single Depot

Require: Depot d, assigned customers C_d, initial profile $\mathbf{p}_d^{(0)}$, trained policy π_θ.
Ensure: Set of feasible initial routes $S_d^{\text{init}} = \{r_k\}_{k\in\mathcal{K}_d}$.

```
// Step 1: Graph Encoding
Build local graph G_d = (V_d, E_d) from C_d and depot d.
Compute node features x_i for all i ∈ V_d.
Initialize embeddings: h_i^(0) = W^(0) x_i + b^(0).
Compute node embeddings H_d^(Γ) = {h_i^(Γ)}_{i∈V_d} and aggregated depot
    representation \overline{h_d} using multi-layer multi-head GAT.
// Step 2: Sequential Multi-Agent Route Decoding
Initialize all agent partial routes r_k ← [d], ∀k ∈ K_d.
Initialize visited set V_visited ← ∅.
for all k ∈ K_d do
    while F_k^t ≠ ∅ do
        // 2.1. Update dynamic state and feasible action set
        Update vehicle state v_k^(t) and node states N_d^(t) from p_d^(t).
        Compute feasible set F_k^t ← {j ∈ C_d | ⋀_{n=1}^N φ_n^d(i, j; p_d^(t)) = True}.
        // 2.2. Decoder GAT for context embedding
        Obtain current context h_c and candidate vectors v_c.
        Update context representation h'_c using multi-layer multi-head
            attention over candidates.
        // 2.3. Compatibility scoring with knowledge masking
        Compute unnormalized scores u_{i,t}^k
        Apply knowledge mask:
            ũ_{i,t}^k = u_{i,t}^k if i ∈ F_k^t ∧ i ∉ V_visited, else −M (≈ −∞).
        Compute selection probabilities: p_{i,t}^k = Softmax(ũ_{i,t}^k).
        // 2.4. Action selection and environment update
        Select next customer: c_{t+1}^k = arg max_i p_{i,t}^k, where c_{t+1}^k ∈ F_k^t.
        Append c_{t+1}^k to route r_k, add c_{t+1}^k to V_visited.
        Update depot profile p_d^(t+1) via f_update to reflect the new visit.
        Update remaining capacity Q_rem^k, feasible sets F_k^{t+1}.
    end while
    Append depot d to r_k to complete the route.
end for
return S_d^init = {r_k}_{k∈K_d}.
```

3.5. Heuristic refinement

While the GAT-based decoder produces feasible initial routes, sequential decoding and local attention biases may leave solutions suboptimal. To improve solution quality without retraining, we apply a single classical or metaheuristic algorithm H to perform time-bounded local refinements over the GAT-initialized routes.

Let $S_d = \{r_k \mid k \in \mathcal{K}_d\}$ denote the initial routes for depot d, refined according to the knowledge-guided cost function:

$$\mathrm{Cost}(r_k) = \gamma \sum_{(i,j)\in r_k} t_{ij} + \delta \sum_{i\in r_k, i\neq d} \pi_i(a_i) + \eta\Psi(r_k, \mathbf{p}_d^{(t)}), \tag{18}$$

with trade-off coefficients $\gamma, \delta, \eta > 0$, where $\pi_i(a_i)$ penalizes time-window violations and $\Psi(r_k, \mathbf{p}_d^{(t)})$ enforces depot-level knowledge consistency, spatial bounds, and load feasibility.

The refinement searches for improved routes within a local neighborhood $\mathcal{N}(r_k)$:

$$r_k' = \arg\min_{r'\in\mathcal{N}(r_k)} \mathrm{Cost}(r'), \quad \text{s.t. } \phi_n^d(r') = \text{True } \forall\phi_n^d \in \Phi_d, \tag{19}$$

where ϕ_n^d are depot-level logical constraints.

The heuristic H is applied to all routes in depot d as:

$$S_d^* = H(S_d; \mathbf{p}_d^{(t)}, \Theta_H), \tag{20}$$

with Θ_H denoting the heuristic's parameterization. Intra- and inter-route refinements are permitted, producing S_d^* that satisfy all feasibility constraints.

Although only one heuristic is active per run, multiple algorithms from a predefined family $\mathcal{H}$ are evaluated across experiments to compare effectiveness and convergence. The process is computationally efficient, typically polynomial in route size, and significantly improves GAT-initialized solutions without retraining.

This procedure, which tightly couples heuristic search with the knowledge-guided cost function and logical feasibility predicates, is formalized in Algorithm 2.

Algorithm 2 Knowledge-Guided Heuristic Refinement for a Single Depot

Require: Initial route set $S_d = \{r_k\}$, depot profile $\mathbf{p}_d^{(t)}$, heuristic H, time budget $T_d^{(H)}$.
Ensure: Refined solution S_d^*.

```
Initialize S*_d ← S_d.
Extract logical predicates Φ_d from p_d^(t).
repeat
   // Step 1: Generate candidate solution using heuristic H
   S'_d ← H(S*_d; Θ_H). {H can be any operator.}
   // Step 2: Evaluate candidate solution
   Compute Cost(S'_d) using the knowledge-guided cost function:
      Cost(S'_d) = Σ_{r_k∈S'_d} [γ Σ_{(i,j)∈r_k} t_ij + δ Σ_{i∈r_k} π_i(a_i) + ηΨ(r_k, p_d^(t))].
   // Step 3: Verify logical feasibility
   if ⋀_{φ^d_n∈Φ_d} φ^d_n(S'_d) = True AND Cost(S'_d) < Cost(S*_d) then
      S*_d ← S'_d {Accept improved, feasible solution.}
   else
      S'_d is rejected. {Infeasible or non-improving.}
   end if
until T_d^(H) is exhausted
return S*_d.
```

3.5.1. Time-budgeted parallel refinement

Profile-based decomposition allows independent concurrent optimization across depots, with parallelism primarily applied during heuristic refinement. Each depot executes its selected heuristic $H \in \mathcal{H}$ independently under a global time or iteration budget.

Let T_d denote the total runtime for depot d completing GAT-based initialization and heuristic refinement:

$$T_d = T_d^{(\mathrm{GAT})} + T_d^{(\mathrm{H})}. \tag{21}$$

The overall parallel runtime is the critical path across depots:

$$T_{\mathrm{par}} = \max_{d \in D} T_d. \tag{22}$$

Depots maintain asynchronous checkpoints of their best solutions:

$$S_d^{(t+1)} = \begin{cases} S_d^*, & \text{if } \mathrm{Cost}(S_d^*) < \mathrm{Cost}(S_d^{(t)}), \\ S_d^{(t)}, & \text{otherwise,} \end{cases} \tag{23}$$

ensuring monotonic improvement without inter-depot coordination.

This mechanism provides a scalable post-optimization layer, enabling each depot to refine neural-initialized routes using localized heuristics in parallel. It achieves near-optimal intra-depot solutions with controlled computational cost and full parallel scalability.

3.6. Training

The policy parameters θ are optimized using the REINFORCE policy gradient estimator with a baseline $b(G)$ to reduce variance and stabilize learning (Sutton et al., 1999; Williams, 1992). The baseline is implemented as a target network with the same architecture as the actor (online) network, updated every λ epochs, inspired by double deep Q-networks.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi\sim\pi_\theta}\left[\Big(L(\pi \mid G) - b(G)\Big) \cdot \sum_{d\in D}\sum_{k\in\mathcal{K}_d}\sum_{t=1}^{T_k} \nabla_\theta \log \pi_\theta(c_t^k \mid s_t^k, \mathbf{p}_d^{(t)})\right], \tag{24}$$

where $L(\pi|G)$ is the total cost of the constructed route set π on instance G, and T_k is the number of decoding steps for vehicle k. In practice, the expectation is approximated by Monte Carlo sampling over a batch of B instances. Therefore, the loss is calculated via:

$$\nabla_\theta \mathcal{L}(\theta) \approx \frac{1}{B}\sum_{b=1}^{B}\Big(L(\pi_b \mid G_b) - b(G_b)\Big) \cdot \sum_{d,k,t} \nabla_\theta \log \pi_\theta(a_{b,t}^k \mid s_{b,t}^k, \mathbf{p}_{d,b}^{(t)}). \tag{25}$$

Training is performed for 20 epochs, using the Adam optimizer, minimizing the expected cost of generated routes while leveraging the baseline to accelerate convergence and enhance stability.

3.7. End-to-end solution workflow

Building upon the components detailed in Sections 3.2–3.6, we now present the complete, integrated workflow of the KGCAF framework. This end-to-end procedure, formalized in Algorithm 3, clarifies how profile-based decomposition, logic-guided GAT construction, and time-budgeted heuristic refinement are orchestrated to produce a final solution. The detailed sub-routines for the GAT construction and heuristic refinement stages are provided in Algorithms 1 and 2, respectively.

Input Specification. The framework takes as input a problem instance $I = (\mathcal{D}, \mathcal{C}, \mathcal{K}, \mathcal{Q}, \dots, \gamma, \delta, \alpha, \beta)$, a trained policy network π_θ, a chosen heuristic operator H from a family $\mathcal{H}$, and a global time budget B_T. The execution mode can be set to either sequential or parallel.

Phase 1: Profile-Based Decomposition. As described in Sections 3 and 3.2, the global set of customers C is partitioned into disjoint subsets $\{C_d\}_{d\in D}$, each assigned to a single depot. For each depot d, an initial profile $\mathbf{p}_d^{(0)} = (\mathbf{X}_d^{(0)}, \mathbf{V}_d^{(0)}, \mathbf{s}_d^{(0)})$ is instantiated, which encodes all static instance data and defines the logical feasibility predicates Φ_d that will govern the subsequent steps.

Phase 2: Logic-Guided GAT Route Initialization. For each depot d, operating either sequentially or in parallel, a set of feasible initial routes S_d^{init} is constructed by the trained GAT policy. This stage, detailed in Algorithm 1, begins by encoding the local graph G_d with a feasibility-aware GAT encoder. A multi-agent decoder then autoregressively builds routes for each vehicle. At every decision step t, the feasible action set $\mathcal{F}_k^t$ is computed from the depot profile $\mathbf{p}_d^{(t)}$ and used to mask infeasible nodes, strictly enforcing all logical predicates Φ_d. The output of this phase is a complete, logically feasible initial solution for each depot, $S_d^{\mathrm{init}} = \{r_k\}_{k\in\mathcal{K}_d}$.

Phase 3: Knowledge-Guided Heuristic Refinement. Each initial solution S_d^{init} serves as a starting point for a time-budgeted local search, as formalized in Algorithm 2. The chosen heuristic H proposes modifications to the routes within a local neighborhood. A candidate solution S'_d is evaluated by the composite cost function defined in Section 3 and is accepted only if it both reduces the total cost and satisfies every logical feasibility predicate $\phi_n^d \in \Phi_d$. The refinement continues until the allocated time budget $T_d^{(H)}$ is exhausted, producing a refined solution S_d^* for each depot.

Global Output. The final global solution S^* is the union of all independently optimized depot solutions, $S^* = \bigcup_{d\in D} S_d^*$. The profile-based decomposition and explicit feasibility masking at every stage guarantee that S^* covers every customer exactly once and respects all vehicle capacity, time window, and regional constraints of the original problem.

Table 2
Hyperparameter settings for the KGCAF framework.

Parameter	Value	Rationale
Optimization weights		
Distance weight (γ)	1.0	Unit weight for the primary cost component.
Time-window penalty weight (δ)	1.0	Balances distance and temporal penalties equally.
Earliness penalty (α)	0.1	Discourages unnecessary waiting without dominating routing.
Lateness penalty (β)	0.2	Moderate penalty; higher values distort learning (see sensitivity analysis).
KGAT architecture		
GAT layers (L)	3	Balances capacity and cost; 1–2 underfit, 4 gives marginal gains.
Attention heads (H)	8	Standard choice; captures most benefit of multi-head attention.
Embedding dimension (d_h)	32	Sufficient expressiveness without overfitting.
Clipping constant (C)	10	Prevents gradient explosion in decoder compatibility scores.
Masking constant (M)	10^6	Approximates $-\infty$ in Softmax to ensure zero probability for infeasible actions.
Training		
Optimizer	Adam	Standard, robust choice for policy gradient methods.
Learning rate	10^{-4}	Ensures stable convergence.
Epochs	20	Sufficient for convergence on the training set.
Batch size (B)	64	Balances training speed and stability under memory constraints.

Algorithm 3 Knowledge-Guided Collaborative Attention Framework (KGCAF)

Require:
I: Problem instance with depots $\mathcal{D}$, customers $\mathcal{C}$, parameters $\gamma, \delta, \alpha, \beta$.
π_θ: Trained GAT policy network with parameters θ.
$\mathcal{H}$: Family of available heuristic refinement operators.
B_T: Global time budget (optional, for time-budgeted execution).
$mode \in \{\text{Sequential}, \text{Parallel}\}$: Execution mode.
Ensure: Feasible solution set $S^* = \{S_d^* \mid d \in \mathcal{D}\}$ covering all customers.
// **Phase 1: Profile-Based Decomposition**
Partition customers: $\{\mathcal{C}_d\}_{d\in\mathcal{D}} \leftarrow$ ProfileDecomposition(I) {See Definition 1.}
for all $d \in \mathcal{D}$ **do**
  Initialize depot profile $\mathbf{p}_d^{(0)} \leftarrow (\mathbf{X}_d^{(0)}, \mathbf{V}_d^{(0)}, \mathbf{s}_d^{(0)})$
end for
// **Phase 2: GAT-Based Route Initialization**
if $mode$ = Parallel **then**
  for all $d \in \mathcal{D}$ **in parallel do**
    $S_d^{\text{init}} \leftarrow$ GATRouteConstruction($d, \mathcal{C}_d, \mathbf{p}_d^{(0)}, \pi_\theta$) {See Algorithm 1.}
  end for
else
  for all $d \in \mathcal{D}$ **sequentially do**
    $S_d^{\text{init}} \leftarrow$ GATRouteConstruction($d, \mathcal{C}_d, \mathbf{p}_d^{(0)}, \pi_\theta$)
  end for
end if
// **Phase 3: Heuristic Refinement**
Select a heuristic operator $H \in \mathcal{H}$.
$T_d^{(H)} \leftarrow$ AllocateBudget($B_T, |\mathcal{D}|, mode$). {Allocate time budget per depot.}
if $mode$ = Parallel **then**
  for all $d \in \mathcal{D}$ **in parallel do**
    $S_d^* \leftarrow$ HeuristicRefinement($S_d^{\text{init}}, \mathbf{p}_d^{(t)}, H, T_d^{(H)}$)
  end for
else
  for all $d \in \mathcal{D}$ **sequentially do**
    $S_d^* \leftarrow$ HeuristicRefinement($S_d^{\text{init}}, \mathbf{p}_d^{(t)}, H, T_d^{(H)}$)
  end for
end if
// **Phase 4: Inter-Depot Synchronization (if needed)**
$S^* \leftarrow$ SyncBestSolutions($\{S_d^*\}_{d\in\mathcal{D}}$) {Merge depot solutions, resolving any boundary inconsistencies.}
return S^*

3.8. Hyperparameter settings and sensitivity analysis

This section details the key hyperparameter settings and presents a summary of the sensitivity analysis conducted to validate them. To ensure reproducibility, Table 2 lists the principal hyperparameters of the KGCAF framework, their chosen values, and the rationale for each.

A sensitivity analysis on the most impactful parameters, GAT layers ($L \in \{1,2,3,4\}$), attention heads ($H \in \{4,8,16\}$), earliness penalty ($\alpha \in \{0.05, 0.1, 0.5\}$), and lateness penalty ($\beta \in \{0.2, 0.5, 1.0, 2.0\}$), is provided in full in the supplementary material. The key findings are:

- **Architecture** (L, H): The framework is robust, with final solution quality varying by less than 3% across all configurations. Our setting ($L = 3, H = 8$) achieved the largest refinement gain (23.8% distance reduction), indicating strong amenability to local search.
- **Penalty weights** (α, β): The lateness penalty β is the most sensitive parameter. Our chosen values ($\alpha = 0.1, \beta = 0.2$) guide the GAT toward spatially efficient routes while relying on heuristic refinement for final time-window feasibility, a deliberate division of labor central to KGCAF's design.

No reasonable alternative parameter setting reverses the relative ranking of KGCAF against the baselines, confirming that the reported improvements reflect the framework's algorithmic contributions rather than hyperparameter tuning.

4. Experiments and discussions

4.1. Experimental setup

We evaluate the proposed hybrid framework on mid- and large-scale multi-depot VRPTW instances, focusing on (i) the effectiveness of *profile-based decomposition*, (ii) the quality of GAT-initialized routes refined via heuristics, (iii) computational gains from *time-budgeted parallel execution*, (iv) the impact of Local (Depot-Specific) vs. Global (Depot-Flexible) heuristic refinements, and (v) comparison with best-known solutions. Experiments use benchmark *Cordeau* and *Vidal* datasets with 50–10,000 customers distributed across 3–12 depots. Each instance was tested under multiple time budgets (1, 5, 15, 20, and 30 min) and across five random seeds to ensure statistical consistency. Noteworthy, the results for the total time budget are reported, except for the time-budget evaluation analysis. To analyze the contribution of each component, we performed ablation studies by comparing KGCAF with two reduced variants: (1) *Decomposition+Heuristics*, which isolates the effect of problem decomposition without learning-based guidance; and (2) *Heuristics Only*, where no decomposition or learning component is used. All variants employ identical heuristic modules and are evaluated under both sequential and parallel modes, using the same objective function as the proposed framework. This setup enables a fair comparison of scalability, convergence, and solution quality across hybrid and purely heuristic paradigms. We compare our framework against heuristic, metaheuristic, and RL-based baselines. All experiments are executed on a 24-core CPU windows system, using Python programming language. Each configuration is averaged over five random seeds ($[1001 - 1005]$) to ensure statistical robustness.

4.2. Scaling VRP instances

Neural networks, particularly attention-based architectures, require input features to lie within a consistent and bounded numerical range to ensure stable gradient propagation and effective training. For the MDVRPTW benchmarks, the raw features — coordinates, demands, time windows, service times, and vehicle capacity — span several orders of magnitude (e.g., coordinates in the hundreds, time windows in the thousands). To address this, we apply instance-level normalization to all inputs before they are passed to the GAT encoder. Customer and depot coordinates are scaled to $[0, 1]$ via the transformation $(x + 100)/200$. Customer demands and service times are divided by 50, and time window parameters (ready time and due date) are divided by 1000. These normalizing constants are derived from the observed upper bounds of each feature across the benchmark datasets, ensuring all values fall approximately within the unit interval. We evaluate two variants of vehicle capacity scaling, resulting in two experimental configurations. In the scaled configuration, the vehicle capacity is set to a constant value of 1 for all instances, which treats capacity constraints as a purely relative relationship between normalized demand and a fixed capacity ceiling. This configuration is used in the main scaled experiments where the primary focus is on comparing heuristic refinement performance under a consistent internal representation. In the unscaled configuration, the vehicle capacity is scaled proportionally to the demand normalization by dividing the original dataset capacity by 50. This preserves the true ratio between vehicle capacity and customer demands, ensuring that the logical feasibility predicates ϕ^d_{cap} operate on the correct geometric relationship rather than a simplified abstraction. The unscaled configuration is employed specifically for the final evaluation against best-known solutions (BKS), as it reflects the actual problem parameters and enables a fully consistent comparison with results reported in the established benchmark literature. In the unscaled configuration, after route construction and heuristic refinement are completed, all solution costs are computed in the original unscaled problem space using the true objective function, guaranteeing that reported gaps are meaningful and directly comparable to published results.

4.3. Heuristic and metaheuristic baselines

Each heuristic is applied individually to GAT-initialized or decomposition-based solutions to ensure consistent evaluation. The list below summarizes all heuristic operators and their key parameters:

- **B2Opt:** Best-improvement 2-opt intra-route search, iteratively reversing segments with maximal gain. Maximum run time 1800s or until no improvement is found.
- **F2Opt:** First-improvement 2-opt, stopping after 200 consecutive non-improving iterations or the time limit.
- **FRelocate:** First-improvement inter-route relocate operator, moving a single customer between routes, with 200 non-improving iteration tolerance.
- **Sh2OptHC:** Shuffled 2-opt hill-climbing with randomized restarts (3 shuffles per round, up to 50 rounds).
- **SA2Opt:** Simulated Annealing using 2-opt neighborhood; initial temperature $T_0 = 1.0$, cooling rate $\alpha = 0.995$, maximum 5000 iterations. Uphill moves accepted with probability $\exp(-\Delta/T)$.
- **Tabu:** Tabu Search combining relocate and swap neighborhoods. Tabu tenure $\tau = \max(5, \sqrt{n})$, maximum 200 iterations or 1800s.
- **RCLInsertion:** Restricted Candidate List insertion heuristic with $\alpha = 0.4$; removal size $\max(1, 0.15 \times N)$; reinserts nodes based on cost-ranked candidate lists.
- **GreedyRelocate:** Greedy relocate heuristic selecting the single move yielding the best immediate improvement, iterated until no further gain.
- **GreedySwap:** Greedy intra- and inter-route customer swap heuristic, choosing the best gain move per iteration until convergence.
- **GRASP:** Greedy Randomized Adaptive Search Procedure combining RCL-based construction ($\alpha = 0.4$) and local 2-opt, relocate, and swap refinements. Performed over 50 iterations.
- **EALNS:** Enhanced Adaptive Large Neighborhood Search with 5 destroy (random, worst-cost, Shaw, route, time-window) and 4 repair (greedy, regret-2, regret-3, noisy greedy) operators, guided by lexicographic simulated annealing (distance primary, penalty secondary; $T_0 = 100$, $\alpha = 0.9995$, $T_{\min} = 0.1$) with segment-based weight adaptation (segment size 100, reaction factor 0.5). Uses 15%–40% removal rate, Pareto front tracking for bi-objective analysis (distance vs. penalty), and a three-phase escape mechanism triggered after 100 segments without improvement: Phase 1 increases removal to 30%–50%, Phase 2 resets to the best solution and reheats to $T_0/2$, and Phase 3 terminates the search. Maximum 100,000 iterations or 3600 s time limit.

These heuristics collectively span both local and global search paradigms, enabling a balanced evaluation of exploitation-oriented (like 2-opt, relocate), exploration-oriented (GRASP, SA, Tabu), and hybrid adaptive (EALNS) strategies for solution refinement.

4.4. RL-based baselines and advanced VRP baselines

We compare our proposed framework with state-of-the-art RL methods, including adaptive large neighborhood search methods SGVNSALS (Bezerra, Souza, & de Souza, 2023) and ALNS (Wang et al., 2024) to assess performance. To further contextualize the performance of the proposed hybrid framework against pure learning-based paradigms, we include the RL-based PDDQN studied in various works like (Ahamed et al., 2021; Bdeir et al., 2021), as well as a standard policy-gradient soft actor-critic (SAC), a rollout value networks based RVN-Rollout, and a variant of GAT network without decomposition and heuristic refinement, denoted as GAT_RL in our evaluations.

Noteworthy, the runtime for ALNS, AC, PDDQN, and SGVNSALS refers to the total time the method required to produce the final solutions, which for the learning-based methods (AC, PDDQN) includes the full training episodes. In contrast, the runtime for GAT_RL, RVN-Rollout, and our proposed KGCAF refers to their pure solution time (i.e., inference or construction time) after any offline training is complete. This distinction is critical for a fair evaluation of computational efficiency during deployment.

4.5. Experimental results

4.5.1. Performance evaluation on benchmarking datasets

In presented reports, the heuristics are restricted to 15 min time-budget per depot, except for EALNS which has an overall 60 min time-budget for all experiments.

Cordeau. Table 3 reports the comparative performance of various local and metaheuristic refinements integrated into the proposed KGCAF framework. The results consistently demonstrate substantial post-optimization gains over the GAT-initialized solutions ($\bar{D} = 33.10$), confirming the effectiveness of heuristic enhancement within KGCAF. Among the evaluated methods, the Greedy approaches achieve the best overall average performance with $\bar{D} = 17.90$ (GS) and $\bar{D} = 17.99$ (GR), while Tabu Search attains the lowest objective total distance values in 10 out of 20 instances. The EALNS method, with $\bar{D} = 19.57$, ranks third overall and achieves best-known performance on instances pr02, pr07, and pr12, demonstrating its particular strength on medium-scale problems. These results indicate that both memory-based (Tabu) and greedy-improvement strategies effectively exploit the structured embeddings produced by KGCAF. The superior performance of greedy methods stems from their aggressive descent behavior, enabling rapid convergence without the overhead of escape mechanisms. Tabu Search complements this by systematically exploring beyond immediate local optima through its recency-based memory, which proves particularly effective when the embedding space contains shallow local minima that would otherwise trap purely greedy approaches. EALNS achieves competitive results through its adaptive layer with dynamic operator selection, allowing it to navigate diverse landscape structures across different instances. However, its substantial computational cost ($\bar{T}$ = 2660.14s, over 20× that of GS at 51.19s) limits its practical efficiency, and its slightly higher average distance compared to greedy methods suggests that the KGCAF solution space is already sufficiently structured that simpler, more focused search strategies can match or exceed the performance of sophisticated adaptive schemes.

The performance patterns reveal important insights about the synergy between learned embeddings and heuristic search. Although B2 and SA2 offer moderate improvements over the initial solutions ($\bar{D}$ = 28.39 and 28.37 respectively), their impact remains limited to intra-route refinements, as these 2-Opt-based methods cannot restructure the route assignments inherited from the KGCAF initialization. In contrast, RCL and Sh2 show higher computational overheads with marginal quality gains ($\bar{D}$ = 33.06 and 32.63), reflecting their weaker guidance under the KGCAF solution space. The stochastic construction in RCL and the hill-climbing acceptance criterion in Sh2 fail to leverage the structured embeddings as effectively as the deterministic greedy or memory-guided search strategies. Notably, the F2 and FR variants show minimal improvement over the initial solutions ($\bar{D}$ = 33.09 and 32.92), indicating that basic insertion heuristics without a strong improvement phase are insufficient to capitalize on the embedding quality. The EALNS results are particularly instructive: while its adaptive mechanism successfully identifies effective operator combinations for specific instances (achieving best-known solutions on three benchmarks), the method's broad exploration strategy occasionally degrades solution quality relative to the focused exploitation of greedy methods, as evidenced on instances pr04–pr06 and pr10 where GS or GR outperform EALNS. This suggests that the KGCAF embeddings inherently encode high-quality structural information, making the solution space relatively smooth and amenable to exploitation-dominant search strategies rather than requiring the exploration–exploitation balancing that EALNS is designed to provide.

Overall, the hybridization of KGCAF with advanced neighborhood exploration yields superior convergence and robustness across the Cordeau benchmark, with the Greedy (GS), Greedy Randomized (GR), and Tabu Search methods emerging as the most effective refinements in terms of the quality-efficiency trade-off. The strong performance of greedy strategies substantiates a key finding: the KGCAF framework's ability to learn structured representations of the problem space produces initial solutions that lead to near-optimal solutions, enabling simple yet highly effective local improvement heuristics to achieve state-of-the-art results. The EALNS method, while achieving competitive quality ($\bar{D}$ = 19.57), highlights an important efficiency-quality tension, its adaptive multi-operator framework provides robustness across heterogeneous instance characteristics but incurs higher runtime without adjusting quality gains over simpler methods. This reinforces the conclusion that the primary contribution of the KGCAF framework lies in its generalization capability across problem instances of varying scale and complexity, producing an embedding space where computationally lightweight heuristics can operate with remarkable effectiveness.

Vidal (a,b). The results on the Vidal (a) and (b) datasets (Table 4) reaffirm the robustness of the KGCAF framework under larger and more complex instances. In both cases, significant post-optimization improvements are observed compared to the GAT-initialized solutions. Among all heuristics, GRASP consistently achieves the lowest average distances ($\bar{D}$ = 114.24 for Vidal(a) and $\bar{D}$ = 109.48 for Vidal(b)), outperforming all other methods on average by a notable margin. This overall superior performance stems from GRASP's multi-start constructive mechanism, which effectively explores the solution space by combining greedy randomized construction with local search, thereby escaping local optima that trap deterministic methods. The EALNS method delivers the second-best results overall ($\bar{D}$ = 131.40 and 121.10), outperforming Tabu Search, Simulated Annealing (SA2) variants, and all single-operator improvement heuristics. EALNS's strong performance can be attributable to its navigating the structured KGCAF landscape by applying context-appropriate perturbations. This adaptability proves particularly valuable on the heterogeneous Vidal instances, where instance-specific characteristics favor different search strategies. However, EALNS incurs the highest average runtime, reflecting the computational cost of maintaining multiple operators and adaptive weight updates. The performance gap between EALNS and GRASP suggests that while adaptive large neighborhood search provides robust diversification, the constructive phase of GRASP better exploits the high-quality structural priors embedded by the KGCAF initialization.

The GS and GR operators also deliver strong results with moderate computational effort, achieving the best or near-best performance on several instances (e.g., pr11, pr12, pr17, pr21 in both subsets). Their effectiveness lies in the greedy acceptance criterion, which aggressively exploits improving moves within the KGCAF-induced solution topology, rapidly converging to high-quality local optima. Notably, on instances where these greedy operators outperform GRASP, the solution landscape around the initial solution likely contains well-defined improvement trajectories that greedy descent can efficiently traverse without the overhead of multi-start diversification. Conversely, Tabu Search incurs higher runtimes ($\bar{T}$ = 3308.87 and 2938.57) with limited marginal benefits, underperforming even the simple B2 heuristic on Vidal(a), indicating that its tabu tenure mechanisms may excessively restrict revisiting productive regions of the search space that the structured KGCAF solutions naturally inhabit. The 2-opt variants (B2, F2, Sh2, SA2) provide modest intra-route refinements, with B2 achieving reasonable improvements at minimal cost ($\bar{T}$ = 0.12 and 0.15 s), while SA2 offers competitive solution quality at a higher but still moderate runtime, illustrating a favorable balance between exploration and exploitation for this temperature-guided 2-opt procedure. Overall, the findings demonstrate that the hybridization of KGCAF with GRASP or EALNS yields superior scalability and solution quality by leveraging complementary strengths; KGCAF provides high-quality structural priors, GRASP supplies effective multi-start diversification with targeted local search, and EALNS contributes adaptive operator selection that robustly handles instance heterogeneity, together demonstrating effective generalization across both Vidal benchmark subsets.

4.5.2. Performance comparison with state-of-the-art methods

Cordeau. Table 5 compares the best-performing variant of the proposed framework KGCAF(GS) against six SOTA baselines and learning-based methods on the 20 Cordeau benchmark instances. The proposed model consistently achieves the best objective values across all instances, with an average distance of $\bar{D}$ = 17.90 and average runtime of $\bar{T}$ = 51.19s. In contrast, the competing methods yield substantially higher average distances: ALNS ($\bar{D}$ = 33.47, +86.98%), GAT_RL ($\bar{D}$ = 66.09, +269.21%), PDDQN ($\bar{D}$ = 67.47, +276.92%), and SGVNSALS ($\bar{D}$ = 44.64, +149.38%). The learning-based methods of AC and RVN-Rollout also fall short, achieving average distances of $\bar{D}$ = 69.49 and $\bar{D}$ = 44.34, respectively. Moreover, KGCAF demonstrates superior computational efficiency, running approximately 70× faster than ALNS,

Table 3
Performance comparison of heuristic refinements within the KGCAF framework on the Cordeau benchmark, reporting average distance ($\bar{D}$) and runtime ($\bar{T}$).

Cordeau	Initial	B2	F2	FR	GR	GS	Sh2	Tabu	SA2	RCL	GRASP	EALNS
pr01	8.62	8.07	8.62	8.62	6.29	6.39	8.59	**6.19**	8.07	9.05	6.46	6.37
pr02	18.06	15.57	18.06	18.06	11.70	11.09	17.89	10.74	15.48	16.82	12.10	**10.66**
pr03	30.40	24.95	30.40	30.40	17.13	16.56	29.53	**15.88**	24.96	30.71	18.49	16.25
pr04	41.90	34.12	41.90	41.90	21.49	**19.52**	40.24	22.34	34.12	39.80	25.07	22.29
pr05	45.94	39.81	45.94	45.94	**24.40**	24.98	45.55	30.94	39.65	47.70	30.18	28.62
pr06	54.38	47.87	54.38	54.38	28.16	**28.09**	54.29	38.94	47.86	54.94	33.99	30.21
pr07	12.33	10.46	12.10	11.59	8.12	8.23	11.91	8.06	10.45	11.76	7.99	**7.88**
pr08	25.90	23.08	25.90	25.90	15.95	14.61	25.78	**14.38**	23.18	26.33	15.80	15.39
pr09	36.37	30.62	36.37	36.37	20.08	19.74	35.90	**18.38**	30.75	38.22	21.78	22.26
pr10	56.21	47.15	56.21	56.21	28.84	**29.94**	55.64	30.17	47.26	52.85	32.69	33.22
pr11	8.78	8.21	8.78	8.78	6.31	6.49	8.73	**6.07**	8.18	8.62	6.10	6.37
pr12	19.14	16.71	19.14	18.31	11.44	11.49	18.45	**10.50**	16.51	19.25	12.00	**10.65**
pr13	31.47	26.72	31.47	31.47	16.26	16.65	31.40	**15.46**	27.06	32.19	18.95	16.16
pr14	40.74	34.73	40.74	40.74	20.65	**19.40**	40.02	22.41	34.97	42.08	25.24	22.30
pr15	45.31	39.08	45.31	45.31	**24.00**	24.50	44.82	27.82	38.86	46.91	29.91	28.38
pr16	57.24	48.86	57.24	57.24	**27.46**	27.81	56.49	41.20	48.44	54.23	35.26	30.97
pr17	12.09	11.03	12.09	11.71	8.28	**8.14**	11.84	7.85	10.91	12.30	8.41	7.95
pr18	27.77	24.57	27.77	27.77	14.82	15.52	27.58	**14.43**	24.55	27.13	15.34	15.41
pr19	35.43	30.42	35.43	33.81	19.85	19.96	35.05	**19.05**	30.35	36.69	20.99	22.31
pr20	53.86	45.73	53.86	53.86	**28.53**	28.90	52.96	29.48	45.79	53.60	32.51	32.75
$\bar{D}$	33.10	28.39	33.09	32.92	**17.99**	**17.90**	32.63	19.51	28.37	33.06	20.46	19.57
$\bar{T}$		0.02	30.02	954.49	113.17	51.19	21.73	464.26	2.06	0.98	45.40	2660.14

Table 4
Performance comparison of heuristic refinements within the KGCAF framework on the Vidal (a) and Vidal (b) benchmarks, reporting average distance ($\bar{D}$) and runtime ($\bar{T}$).

Vidal (a)	Initial	B2	F2	FR	GR	GS	Sh2	Tabu	SA2	RCL	GRASP	EALNS
pr11	142.55	113.98	142.55	142.55	**70.02**	**68.75**	139.58	116.69	113.68	135.63	73.47	85.61
pr12	188.32	150.44	188.32	188.32	97.22	**90.35**	185.49	162.95	150.74	177.62	93.15	113.67
pr13	229.81	187.46	229.81	229.81	145.70	**114.72**	229.46	213.18	187.81	222.22	119.43	136.42
pr14	293.58	246.94	293.58	293.58	218.13	164.94	292.84	280.44	247.71	274.01	**141.54**	163.28
pr15	326.62	260.87	326.62	326.62	261.02	197.93	326.26	316.54	261.90	312.93	**163.63**	189.95
pr16	364.48	294.69	364.48	364.48	323.20	257.28	364.48	356.28	297.95	350.19	**180.24**	211.13
pr17	124.81	103.71	124.81	124.81	**63.94**	64.58	124.84	93.73	103.86	118.88	70.16	78.90
pr18	178.84	146.02	178.69	178.84	102.79	**89.08**	178.05	153.43	146.46	180.26	96.39	113.91
pr19	243.58	197.03	243.58	243.58	169.19	**133.97**	242.15	222.70	197.26	232.90	127.05	148.90
pr20	292.21	231.25	292.21	292.21	223.34	190.38	290.44	268.33	233.54	280.82	**150.60**	162.63
pr21	103.70	87.22	103.64	103.62	61.21	**59.81**	102.58	68.20	87.81	103.24	62.94	71.50
pr22	155.08	124.56	155.10	155.08	84.22	**82.59**	153.39	115.52	124.78	146.40	88.70	99.69
pr23	208.94	167.89	208.94	208.97	116.64	**109.75**	207.72	170.08	167.79	194.94	117.78	132.71
$\bar{D}$	219.42	177.85	219.41	219.42	148.97	124.93	218.25	195.24	178.56	210.00	**114.24**	131.40
$\bar{T}$		0.12	351.63	3907.66	2611.15	2101.85	201.87	3308.87	6.81	35.71	1736.29	3623.17

Vidal (b)	Initial	B2	F2	FR	GR	GS	Sh2	Tabu	SA2	RCL	GRASP	EALNS
pr11	135.90	106.54	135.90	135.90	62.87	**60.09**	134.10	105.40	107.15	136.42	67.95	75.02
pr12	181.55	139.57	181.55	181.55	100.53	**80.82**	179.41	156.35	139.53	177.92	88.54	97.34
pr13	224.75	179.07	224.75	224.75	157.09	122.38	222.24	207.40	180.28	218.69	**113.61**	120.87
pr14	275.40	214.23	275.40	275.40	216.74	167.67	275.64	259.47	217.87	268.34	**134.08**	146.90
pr15	326.07	255.92	326.07	326.07	279.12	229.20	325.85	313.45	256.50	315.43	**155.73**	173.37
pr16	355.37	278.36	355.37	355.37	314.65	275.84	353.80	346.17	281.66	358.27	**176.10**	189.19
pr17	121.60	96.11	121.60	121.60	**58.73**	59.64	120.90	89.92	96.21	115.40	62.76	72.99
pr18	172.99	139.06	172.99	172.99	102.39	**87.91**	172.99	144.73	138.89	171.19	92.17	102.06
pr19	230.97	186.50	230.97	230.97	161.10	133.45	229.07	204.19	188.14	228.54	**118.92**	131.82
pr20	292.21	230.39	292.21	292.21	222.83	189.89	290.00	268.94	233.07	284.80	**147.98**	159.80
pr21	104.06	87.22	104.06	103.92	**60.82**	58.44	103.65	69.63	87.57	101.32	61.69	71.65
pr22	155.26	124.38	155.26	154.77	**82.37**	83.20	151.80	112.26	125.04	149.37	89.07	100.03
pr23	208.89	167.44	208.80	208.89	113.71	**107.93**	207.72	168.02	168.53	194.76	114.61	133.33
$\bar{D}$	214.23	169.60	214.23	214.19	148.69	127.42	212.86	188.15	170.80	209.26	**109.48**	121.10
$\bar{T}$		0.15	389.74	3631.44	2337.96	1845.59	315.11	2938.57	6.17	41.58	2067.78	3622.03

8× faster than GAT_RL, 20× faster than PDDQN, and 42× faster than SGVNSALS on average, with an exception of RVN-Rollout which produces results in 3× less time. Overall, KGCAF(GS) achieves substantial quality improvements with acceptable computational costs, confirming its robustness and superiority across all Cordeau instances.

Vidal. Table 6 presents the performance of the proposed KGCAF framework (best variant (GS)) against state-of-the-art and learning-based methods on the Vidal (a) and (b) benchmark instances. As findings show, KGCAF(GS) consistently achieves the best total distance across all instances in both datasets (all bold entries), with average distances

Table 5
Comparison of the best-performing KGCAF variant (GS) against SOTA baselines, including ALNS, GAT_RL, PDDQN, SGVNSALS, AC, and RVN-Rollout, on the Cordeau instances. The table reports average distance $\bar{D}$ and runtime by $\bar{T}$; the best results are presented in bold. The percentage gap ($Gap|D|$) shows the relative improvement of KGCAF(GS) over each baseline's average distance.

Cordeau	$\|C\|$	$\|D\|$	ALNS		GAT_RL		PDDQN		SGVNSALS		AC		RVN-Rollout		KGCAF(GS)	
Instance			D	T	D	T	D	T	D	T	D	T	D	T	D	T
pr01	48	4	7.60	1190.83	16.63	107.09	18.55	208.44	11.44	35.00	18.08	683.59	11.14	0.55	**6.39**	**0.12**
pr02	96	4	19.11	2228.56	34.53	209.28	34.23	425.41	23.34	480.98	32.32	2287.00	24.95	2.72	**11.09**	**2.90**
pr03	144	4	31.32	2959.75	65.21	321.58	67.12	767.58	39.51	1320.99	64.42	4183.09	41.56	**7.28**	**16.56**	11.25
pr04	192	4	37.93	3269.61	74.05	469.58	72.06	1191.03	51.02	2280.65	73.07	5846.10	38.36	**13.81**	**19.52**	57.27
pr05	240	4	44.37	2221.50	77.94	599.89	82.00	1454.76	64.8	3480.59	80.28	7881.38	47.12	**23.05**	**24.98**	129.89
pr06	288	4	51.04	6154.87	105.26	684.84	110.30	1758.07	73.86	3480.92	105.71	7463.16	82.25	**37.26**	**28.09**	249.22
pr07	72	6	15.39	3152.89	29.07	187.76	31.30	387.06	17.31	480.16	31.73	1356.53	24.75	**1.09**	**8.23**	0.17
pr08	144	6	26.17	1911.45	59.00	326.15	55.99	884.08	38.84	2640.78	55.09	3446.62	29.61	6.34	**14.61**	**3.24**
pr09	216	6	43.54	6002.02	87.59	476.69	89.61	1254.95	52.23	2880.95	88.67	5417.60	69.66	16.44	**19.74**	**15.85**
pr10	288	6	57.09	6604.08	108.10	640.93	116.19	1802.96	75.53	3480.23	120.22	7682.88	65.19	**32.59**	**29.94**	63.09
pr11	48	4	8.55	2865.59	18.12	111.45	18.40	266.72	10.77	120.89	18.86	867.19	11.88	0.56	**6.49**	**0.11**
pr12	96	4	19.63	2877.52	34.68	216.90	32.67	551.49	24.29	360.98	33.94	1975.74	25.52	**2.38**	**11.49**	2.77
pr13	144	4	31.44	1762.72	61.24	316.60	63.86	861.31	40.8	2400.89	66.28	3409.51	49.37	**6.87**	**16.65**	**11.07**
pr14	192	4	39.17	3388.81	73.47	417.13	76.45	1147.33	52.52	3480.84	77.15	4838.94	52.85	**12.46**	**19.40**	48.51
pr15	240	4	46.83	5617.36	78.09	582.03	80.18	1549.31	59.72	3480.69	85.58	6399.16	41.91	**22.90**	**24.50**	112.95
pr16	288	4	50.99	6071.86	109.59	712.06	106.88	1818.05	72.42	3480.80	116.93	8001.52	72.69	**34.36**	**27.81**	233.80
pr17	72	6	15.02	4227.66	30.71	183.96	32.51	402.34	18.5	240.91	29.41	1428.02	22.77	1.20	**8.14**	**0.18**
pr18	144	6	27.14	4990.00	56.31	359.27	57.17	916.15	37.98	1920.92	57.57	3504.55	31.15	6.62	**15.52**	**3.36**
pr19	216	6	42.66	2518.38	88.46	538.65	88.94	1333.28	50.28	3000.91	93.24	5791.29	51.77	17.74	**19.96**	**16.13**
pr20	288	6	54.43	2471.23	113.66	724.18	115.03	1830.79	77.58	3480.91	117.03	8261.67	60.13	**31.81**	**28.90**	61.95
Average			33.47	3624.33	66.09	409.30	67.47	1040.56	44.64	2126.50	69.49	4280.56	44.34	**13.90**	**17.90**	51.19
$Gap\|D\|$			+86.98%		+269.21%		+276.92%		+149.38%		+288.21%		+147.71%			

of $\bar{D}$ = 124.93 for Vidal (a) and $\bar{D}$ = 127.42 for Vidal (b), significantly outperforming all competing methods. The relative improvements over the primary baselines are substantial: +74.28%, +216.40%, +385.19%, and +141.25% for Vidal (a), and +57.28%, +203.97%, +383.22%, and +117.32% for Vidal (b) over ALNS, GAT_RL, PDDQN, and SGVNSALS (restricted to 3600 s runtime), respectively. In terms of the other learning-based baselines AC and RVN-Rollout, AC yields average distance of $\bar{D}$ = 335.13 for Vidal (a) and $\bar{D}$ = 334.39 for Vidal(b), representing gaps of +168.25% and +162.40% over KGCAF(GS), respectively. RVN-Rollout performs markedly better, achieving average distance of $\bar{D}$ = 207.24 for Vidal(a) and $\bar{D}$ = 203.39 for Vidal(b), with corresponding gaps of +65.87% and +59.61%. While RVN-Rollout shows competitive performance with superior time efficiency relative to the hand-crafted heuristics, it still falls well short of KGCAAF(GS)'s solution quality. While the competing methods often require hours of runtime, KGCAF(GS) delivers high-quality solutions within tens to hundreds of seconds, highlighting its efficiency and scalability. Overall, KGCAF effectively leverages the structured solution space provided by the framework, achieving superior solution quality and computational efficiency across both Vidal benchmark subsets.

4.5.3. Performance across multiple seeds and execution modes

Table 7 presents the performance of the proposed framework on Vidal (a) instances over 5 runs with different random seeds (1001–1005), reporting the average and standard deviation for each depot (0–3). Results are shown for both parallel and sequential executions. Across all depots and instances, KGCAF(GS) consistently achieves the lowest average distances with small standard deviations, indicating robust performance and low variability. The results show that the proposed KGCAF framework not only provides superior solution quality but also demonstrates stable performance across seeds and execution modes. Parallel execution slightly accelerates runtime while preserving solution quality, confirming the framework's efficiency and scalability for large multi-depot instances.

4.5.4. Impact of time budget on heuristic performance

Table 8 presents the sequential performance of the proposed framework and several baseline heuristics on large-scale Vidal (a) instances under five different time budgets ranging from 1 to 30 min. As expected, solution quality generally improves with increasing time, with GRASP consistently achieving the lowest distances across all instances and budgets, demonstrating both efficiency and robustness. FR and GS show competitive performance in short budgets for smaller instances but are less effective on larger ones. Tabu Search maintains stable results but does not match GRASP in longer runs. Notably, GRASP shows effectiveness from the starting time budget (1 min), without improving over time. This shows quick convergence to an optimal solution, compared to the other heuristics.

In contrast, the EALNS heuristic exhibits a more gradual improvement trajectory, with its solution quality steadily enhancing as the time budget increases from 1 to 30 min. For instance, on the pr16 instance, EALNS reduces its distance from 231.65 at 1 min to 203.26 at 30 min, reflecting a consistent downward trend that leverages the additional computational time effectively. While EALNS does not surpass GRASP in absolute terms, it demonstrates competitive performance against FR and GS.

These results highlight the framework's ability to leverage additional computation time effectively, achieving near-optimal solutions while preserving scalability across varying problem sizes. The contrasting convergence behaviors, GRASP's rapid saturation versus EALNS's sustained improvement, suggest that EALNS benefits more explicitly from extended time budgets through its adaptive destroy-and-repair mechanisms, which continue to explore the solution space productively. It is noteworthy that 15 min often provide a good balance between solution quality and the run time, providing high quality solutions within a reasonable time.

4.5.5. Ablation study

Table 9 presents the ablation study comparing two simplified variants of the proposed framework: *Heuristics Only* (without decomposition and GAT learning) and *Decomposition + Heuristics* (without GAT). The *Heuristics Only* variant produces substantially higher total distances (D) across all instances, accompanied by elevated computation times (T) and time penalties (TP), indicating poor scalability and limited adherence to time windows. Notably, the impact of the time budget is evident in the performance of EALNS under the *Heuristics Only* configuration: constrained to a fixed 30 min budget identical to other heuristics, EALNS achieves a remarkably low time penalty (e.g., 43.18 for pr11) but suffers from a severely degraded total distance (123.34 for pr11), exposing a fundamental trade-off where strict time limits

Table 6
Performance comparison of the proposed KGCAF framework (best variant KGCAF(GS)) against SOTA methods, including ALNS, GAT_RL, PDDQN, SGVNSALS, AC, and RVN-Rollout, on the Vidal (a, b) datasets, showing average distance ($\bar{D}$) and runtime ($\bar{T}$); best results are presented in bold. The percentage gap ($Gap|D|$) shows the relative improvement of KGCAF(GS) over each baseline's average distance.

Vidal(a)	$\|C\|$	$\|D\|$	ALNS		GAT_RL		PDDQN		SGVNSALS		AC		RVN-Rollout		KGCAF(GS)	
Instance			D	T	D	T	D	T	D	T	D	T	D	T	D	T
pr11a	360	4	132.69	2443.19	231.29	83.43	226.93	2226.81	190.86	3600	226.98	12126.08	116.46	55.62	**68.75**	720.04
pr12a	480	4	183.03	7256.64	309.03	110.20	300.38	2945.86	242.53	3600	313.14	15524.92	149.93	123.09	**90.35**	1948.13
pr13a	600	4	218.40	7709.00	390.10	150.90	313.09	3246.89	303.51	3600	313.91	15363.82	190.53	229.00	**114.72**	2864.15
pr14a	720	4	263.71	8083.16	453.76	194.05	315.55	3479.54	354.99	3600	313.72	16429.01	235.24	346.84	**164.94**	3120.62
pr15a	840	4	294.95	8480.17	523.22	234.18	309.79	3598.03	406.64	3600	309.03	17059.28	257.84	574.14	**197.93**	3639.59
pr16a	960	4	330.15	4339.50	544.44	272.68	311.80	3727.47	464.37	3600	312.13	17680.15	313.44	749.81	**257.28**	3638.16
pr17a	360	6	129.46	2694.58	228.07	81.47	218.38	1952.01	174.00	3600	227.33	8275.94	140.59	60.19	**64.58**	524.89
pr18a	520	6	187.71	5747.19	327.32	126.75	326.26	2931.15	256.91	3600	338.53	12452.51	210.95	119.64	**89.08**	2030.69
pr19a	700	6	241.19	5886.77	440.58	187.20	430.63	4143.33	325.64	3600	447.05	28070.20	257.46	278.61	**133.97**	2827.14
pr20a	880	6	256.92	4326.69	550.48	243.91	469.07	4730.46	371.55	3600	470.40	32848.19	216.29	616.25	**190.38**	2847.81
pr21a	420	12	127.93	9766.02	260.50	93.93	266.64	2326.49	182.99	3600	271.57	13654.60	181.87	72.41	**59.81**	237.85
pr22a	600	12	175.55	4760.64	374.56	151.21	374.57	3437.46	255.72	3600	375.75	21599.76	192.80	174.39	**82.59**	936.16
pr23a	780	12	228.02	11102.03	505.45	208.41	493.04	4775.64	311.02	3600	499.09	30185.87	231.69	370.06	**109.75**	1988.75
Average			217.74	6252.61	395.29	164.49	606.15	6214.74	301.40	3600	335.13	18559.28	207.24	331.35	124.93	1406.51
$Gap\|D\|$			+74.28%		+216.40%		+385.19%		+141.25%		+168.25%		+65.87%			
Vidal(b)	$\|C\|$	$\|D\|$	ALNS		GAT_RL		PDDQN		SGVNSALS		Ac		RVN-Rollout		KGCAF(GS)	
Instance			D	T	D	T	D	T	D	T	D	T	D	T	D	T
pr11b	360	4	125.35	2443.19	215.48	59.45	220.96	1856.90	168.39	3600	229.87	9156.45	100.92	63.08	**60.09**	932.76
pr12b	480	4	155.77	7256.64	293.03	88.72	292.08	2620.35	221.07	3600	294.47	11483.23	113.36	116.32	**80.82**	2016.04
pr13b	600	4	195.36	7709.00	377.53	126.45	315.21	2925.01	272.75	3600	311.83	13471.46	208.82	203.24	**122.38**	2324.79
pr14b	720	4	249.69	8083.16	438.91	163.81	310.82	3076.12	329.12	3600	316.12	13820.93	231.27	355.27	**167.67**	2800.28
pr15b	840	4	291.80	8480.17	523.45	201.41	316.74	3129.41	381.08	3600	309.61	14339.64	314.48	474.53	**229.20**	2897.88
pr16b	960	4	300.84	4339.50	525.06	244.32	318.74	3262.44	440.68	3600	302.75	25652.04	362.17	740.27	**275.84**	3087.24
pr17b	360	6	118.37	2694.58	219.73	66.79	226.50	1882.80	155.71	3600	220.80	14020.24	119.99	51.77	**59.64**	726.57
pr18b	520	6	167.93	5747.19	321.48	105.76	332.24	2879.30	234.69	3600	326.12	22672.07	198.13	120.06	**87.91**	1480.32
pr19b	700	6	217.92	5886.77	440.07	154.48	432.15	4099.34	307.42	3600	430.39	30474.86	219.77	322.28	**133.45**	2203.76
pr20b	880	6	251.91	4326.69	552.00	215.21	476.61	4663.49	372.82	3600	474.62	42397.35	213.88	670.36	**189.89**	2850.64
pr21b	420	12	126.67	9766.02	254.82	75.43	263.13	2206.34	179.36	3600	258.23	17414.95	134.41	73.16	**58.44**	128.23
pr22b	600	12	172.73	4760.64	376.08	125.98	380.98	3513.71	232.58	3600	380.91	26332.57	155.11	212.82	**83.20**	504.35
pr23b	780	12	231.04	11102.03	497.57	180.01	498.37	4655.35	304.22	3600	491.45	34553.69	252.32	368.43	**107.93**	2039.79
Average (Vidal b)			200.41	6353.51	387.32	139.06	615.73	6298.76	276.91	3600	334.39	21983.78	203.39	290.12	127.42	1845.59
$Gap\|D\|$			+57.28%		+203.97%		+383.22%		+117.32%		+162.40%		+59.61%			

force the algorithm to sacrifice solution quality for temporal feasibility. Incorporating decomposition with heuristics reduces total distances significantly; however, it leads to dramatically higher time penalties, reflecting excessive violations of time windows due to longer run times. Under the *Decomposition + Heuristics* variant, EALNS delivers drastically improved distances (e.g., 39.99 for pr11) while its time penalty, though elevated compared to the full framework, remains contained relative to other heuristics in this variant, demonstrating that decomposition effectively redirects the limited time budget toward more productive search rather than wasteful feasibility repairs.

The proposed full framework (KGCAF) achieves a well-balanced trade-off, maintaining low total distances while preserving time windows effectively and keeping run times reasonable, demonstrating the importance of integrating GAT-based learning with decomposition and heuristics for efficient, scalable performance. This progression across configurations — from the time-starved EALNS in *Heuristics Only* to its resource-efficient counterpart in the full KGCAF — confirms that the GAT-based initialization is critical for maximizing the value extracted from any given time budget, enabling the heuristic to converge rapidly toward high-quality, time-window-feasible solutions within practical computational limits.

4.5.6. Comparative discussion of Local (depot-specific) vs. Global (depot-flexible) heuristic refinement

Experimental results on scaled global refinement: Cordeau instances. The comparative analysis of Local (Table 3) and Global (Table 10) execution modes within the KGCAF framework reveals a fundamental trade-off between computational efficiency and solution quality. In the Local refinement mode, the heuristics operate under a restricted computational budget, resulting in extremely fast execution times. Excluding the EALNS, the average runtimes for constructive heuristics are negligible (e.g., B2 at 0.02s, SA2 at 2.06s), with even the more intensive Tabu Search completing in an average of 464.26 s. This mode produces a clear hierarchy where the Greedy Random family (GR, GS) dominates, achieving an average distance of $\bar{D}$ = 17.99 and 17.90, respectively, significantly outperforming the initial solution of 33.10. However, this local execution can cause a barrier for iterative improvement heuristics like Tabu Search, which only manages a mean distance of 19.51, losing to the faster, single-pass GR/GS heuristics. The performance of F2 and FR is nearly identical to the initial solution, indicating that local 2-opt and relocation moves alone are insufficient to escape deep local optima without a broader search strategy or a larger time allowance.

In contrast, the Global refinement mode operates with a generous 30-minute time budget (60-minutes for GRASP and EALNS), fundamentally altering the performance landscape. The most immediate and critical observation is the complete inversion of the heuristic hierarchy. The GR and GS heuristics, which were the undisputed leaders in the local mode, catastrophically collapse in the global mode. GR's average distance dramatically *increases* from 17.99 to 28.74, and GS's from 17.90 to 25.52, performing worse than a simple, fast 2-opt (B2 at 28.26). This degradation demonstrates that greedy randomized construction, when iterated for an extended period without a robust, long-memory improvement mechanism, suffers from over-optimization and convergence to deep, narrow local optima from which it cannot escape. Conversely, the true global optimizers come to the fore. The powerful metaheuristics EALNS and GRASP completely dominate, achieving average distances of $\bar{D}$ = 20.35 and 22.65, respectively, marginally outperformed by the local mode. Crucially, the B2 and F2 heuristics, which were ineffective locally, produce solutions with lower total distance in global mode ($\bar{D}$ = 28.26, $\bar{D}$ = 28.29), proving that

Table 7
Stability and scalability analysis of KGCAF heuristic variants on representative large-scale Vidal (a) instances. The table reports the mean distance and standard deviation (mean±std) for each depot (0–3) over five random seeds, for both parallel and sequential execution modes. The best mean distance per depot is in bold, demonstrating the robustness of the framework across different execution modes.

	#	B2	F2	FR	GR	GS	Sh2	Tabu	SA2	RCL	GRASP	EALNS
Parallel												
pr11	0	30.03 ± 0.0	29.93 ± 0.0	**14.18** ± 0.0	18.09 ± 0.0	18.92 ± 0.0	29.92 ± 0.01	26.91 ± 0.0	29.97 ± 0.12	29.93 ± 0.01	15.29 ± 0.68	24.24 ± 0.15
(360)	1	36.51 ± 0.0	36.72 ± 0.0	**19.93** ± 0.87	20.83 ± 0.11	22.29 ± 0.0	36.69 ± 0.14	41.99 ± 0.59	36.83 ± 0.14	36.69 ± 0.14	18.64 ± 1.04	29.61 ± 0.39
	2	31.68 ± 0.0	31.72 ± 0.0	**13.27** ± 0.0	15.61 ± 0.0	18.41 ± 0.0	31.71 ± 0.05	29.02 ± 0.0	31.63 ± 0.02	31.71 ± 0.05	13.51 ± 0.71	20.27 ± 0.20
	3	14.57 ± 0.0	14.45 ± 0.0	8.05 ± 0.0	8.11 ± 0.0	9.09 ± 0.0	14.45 ± 0.0	11.66 ± 0.0	14.52 ± 0.06	14.45 ± 0.00	**7.4** ± 0.14	9.21 ± 0.04
pr13	0	49.81 ± 0.0	49.83 ± 0.0	29.08 ± 0.70	33.40 ± 0.63	31.06 ± 0.01	49.83 ± 0.0	57.20 ± 0.26	49.64 ± 0.19	38.41 ± 1.27	**24.30** ± 0.60	38.65 ± 0.39
(600)	1	63.85 ± 0.0	63.40 ± 0.0	41.88 ± 0.56	51.69 ± 1.08	38.64 ± 0.48	63.38 ± 0.07	73.08 ± 0.22	63.86 ± 0.27	52.36 ± 1.13	**28.93** ± 0.79	44.85 ± 0.21
	2	49.62 ± 0.0	49.85 ± 0.0	32.12 ± 0.78	31.60 ± 1.39	26.78 ± 0.09	49.85 ± 0.01	57.02 ± 0.15	49.90 ± 0.21	38.06 ± 1.76	**23.80** ± 0.48	35.65 ± 0.54
	3	22.78 ± 0.0	22.91 ± 0.0	**10.89** ± 0.0	13.24 ± 0.0	14.03 ± 0.0	22.98 ± 0.09	21.87 ± 0.0	23.02 ± 0.14	12.95 ± 0.75	11.26 ± 0.44	14.98 ± 0.30
pr15	0	62.99 ± 0.0	63.41 ± 0.0	49.03 ± 1.01	60.73 ± 2.69	45.48 ± 1.65	63.32 ± 0.10	77.35 ± 0.37	63.29 ± 0.13	46.85 ± 1.86	**33.08** ± 1.43	49.47 ± 0.24
(840)	1	81.25 ± 0.0	81.27 ± 0.0	69.21 ± 0.93	86.98 ± 1.94	64.84 ± 3.01	81.27 ± 0.0	100.57 ± 0.52	81.78 ± 0.20	67.67 ± 2.26	**41.41** ± 1.44	59.97 ± 0.66
	2	77.85 ± 0.0	77.95 ± 0.0	59.81 ± 1.70	77.98 ± 2.55	55.89 ± 3.23	77.90 ± 0.08	95.64 ± 0.09	77.79 ± 0.13	60.05 ± 1.47	**36.55** ± 1.23	52.29 ± 1.04
	3	35.80 ± 0.0	36.16 ± 0.0	21.06 ± 0.65	22.58 ± 1.13	21.02 ± 0.02	36.13 ± 0.05	41.33 ± 0.42	36.01 ± 0.40	20.48 ± 0.81	**16.02** ± 0.30	25.49 ± 0.20
pr16	0	73.14 ± 0.0	72.90 ± 0.0	62.59 ± 1.39	77.88 ± 1.49	56.67 ± 2.28	72.91 ± 0.01	89.17 ± 0.39	73.51 ± 0.27	55.86 ± 1.76	**38.39** ± 0.89	54.82 ± 1.04
(960)	1	95.67 ± 0.0	96.43 ± 0.0	76.45 ± 0.88	106.31 ± 1.23	86.68 ± 3.05	96.45 ± 0.06	115.41 ± 0.07	96.89 ± 0.32	77.96 ± 1.32	**48.90** ± 1.29	65.05 ± 1.69
	2	84.83 ± 0.0	83.95 ± 0.0	73.30 ± 1.34	93.22 ± 1.41	68.53 ± 3.31	83.95 ± 0.00	104.94 ± 0.36	85.80 ± 0.19	67.95 ± 2.01	**40.51** ± 1.53	57.13 ± 0.66
	3	38.98 ± 0.0	39.34 ± 0.0	24.68 ± 0.57	28.87 ± 1.25	23.33 ± 0.26	39.39 ± 0.06	44.19 ± 0.23	39.09 ± 0.35	22.57 ± 0.90	**18.01** ± 0.33	31.35 ± 0.26
Sequential												
pr11	0	30.04 ± 0.0	29.93 ± 0.0	**14.19** ± 0.0	18.09 ± 0.0	18.93 ± 0.0	29.92 ± 0.03	26.91 ± 0.0	30.03 ± 0.17	21.94 ± 1.37	15.17 ± 0.37	23.93 ± 0.16
(360)	1	36.52 ± 0.0	36.72 ± 0.0	**19.20** ± 0.0	20.78 ± 0.0	22.29 ± 0.0	36.76 ± 0.16	41.73 ± 0.0	36.90 ± 0.29	33.06 ± 0.49	18.59 ± 1.17	28.86 ± 0.43
	2	31.68 ± 0.0	31.73 ± 0.0	**13.28** ± 0.0	15.61 ± 0.0	18.41 ± 0.0	31.73 ± 0.01	29.02 ± 0.0	31.63 ± 0.02	22.64 ± 1.04	13.75 ± 0.43	20.19 ± 0.16
	3	14.57 ± 0.0	14.45 ± 0.0	8.05 ± 0.0	8.11 ± 0.0	9.09 ± 0.0	14.49 ± 0.08	11.66 ± 0.0	14.51 ± 0.09	8.47 ± 0.62	**7.26** ± 0.22	8.50 ± 0.06
pr13	0	49.81 ± 0.0	49.83 ± 0.0	29.32 ± 0.58	33.82 ± 0.59	31.11 ± 0.14	49.86 ± 0.04	57.22 ± 0.27	49.80 ± 0.20	38.24 ± 0.96	**24.18** ± 0.89	38.56 ± 0.41
(600)	1	63.85 ± 0.0	63.40 ± 0.0	42.30 ± 0.62	52.13 ± 1.21	38.85 ± 0.61	63.37 ± 0.07	73.13 ± 0.24	63.77 ± 0.30	53.21 ± 1.42	**29.79** ± 1.04	43.93 ± 0.49
	2	49.62 ± 0.0	49.85 ± 0.0	32.34 ± 0.48	32.35 ± 0.38	26.73 ± 0.05	49.86 ± 0.01	57.03 ± 0.10	49.91 ± 0.23	38.12 ± 1.45	**23.47** ± 0.62	34.86 ± 0.36
	3	22.78 ± 0.0	22.91 ± 0.0	**10.89** ± 0.0	13.24 ± 0.0	14.03 ± 0.0	22.90 ± 0.03	21.87 ± 0.00	23.07 ± 0.16	13.13 ± 0.47	11.02 ± 0.33	14.77 ± 0.13
pr15	0	62.99 ± 0.0	63.41 ± 0.0	48.59 ± 0.13	59.67 ± 1.39	44.68 ± 0.80	63.33 ± 0.09	77.13 ± 0.21	63.41 ± 0.17	48.12 ± 0.60	**35.94** ± 2.61	48.77 ± 0.23
(840)	1	81.25	81.27 ± 0.0	68.73 ± 0.49	86.12 ± 0.78	62.93 ± 1.37	81.27 ± 0.0	100.30 ± 0.05	81.86 ± 0.34	67.33 ± 1.04	**41.72** ± 1.06	59.67 ± 0.70
	2	77.85 ± 0.0	77.95 ± 0.0	58.81 ± 0.54	75.86 ± 1.33	54.01 ± 0.84	77.94 ± 0.02	95.51 ± 0.06	77.79 ± 0.23	60.87 ± 2.25	**37.53** ± 2.91	51.35 ± 0.78
	3	35.80 ± 0.0	36.16 ± 0.0	20.32 ± 0.40	21.87 ± 0.51	21.01 ± 0.0	36.12 ± 0.08	41.29 ± 0.67	36.14 ± 0.31	20.50 ± 0.71	**16.08** ± 0.56	25.10 ± 0.19
pr16	0	73.14 ± 0.0	72.90 ± 0.0	62.08 ± 0.21	77.19 ± 0.23	55.82 ± 0.27	72.89 ± 0.06	89.00 ± 0.04	73.19 ± 0.37	55.19 ± 1.38	**39.20** ± 2.47	53.56 ± 1.18
(960)	1	95.67 ± 0.0	97.09 ± 0.70	76.13 ± 0.16	105.74 ± 0.29	85.27 ± 0.62	96.40 ± 0.06	115.43 ± 0.05	97.07 ± 0.40	78.74 ± 3.16	**50.09** ± 4.52	63.52 ± 0.59
	2	84.83 ± 0.0	84.64 ± 0.76	72.85 ± 0.06	92.89 ± 0.00	67.03 ± 0.51	83.96 ± 0.03	104.94 ± 0.26	85.64 ± 0.34	66.26 ± 1.70	**43.05** ± 4.23	56.29 ± 1.14
	3	38.98 ± 0.0	39.34 ± 0.0	24.54 ± 0.10	28.46 ± 0.08	23.16 ± 0.07	39.32 ± 0.03	44.06 ± 0.05	39.08 ± 0.17	21.98 ± 1.13	**17.94** ± 0.58	30.56 ± 0.41

exploration breadth can be more effective than exploitation depth for these simple operators when given sufficient time.

The core impact of the execution mode lies in its interaction with a heuristic's exploration–exploitation balance and memory mechanism. The Local mode acts as a sprint, favoring construction heuristics (GR, GS) that build a single, high-quality solution quickly by exploiting immediate greedy choices without the burden of escaping subsequent local optima. The Global mode is a marathon that harshly penalizes the memoryless nature of GR/GS, which end up cycling or stagnating, while richly rewarding metaheuristics (EALNS, GRASP) that systematically destroy and rebuild solutions, or even simple restart mechanisms (B2), to traverse the solution space broadly. The dramatic increase in total distance of GR and GS under a time budget is the most significant finding; it provides strong empirical evidence that for depot-flexible MDVRPTW variants, investing time in a weak heuristic without an adaptive memory or diversification strategy is not just inefficient but actively detrimental to final solution quality compared to a rapid restart approach. The Global refinement framework therefore acts as a crucial litmus test, clearly separating scalable, long-horizon metaheuristics from myopic greedy strategies.

Experimental results on scaled global refinement: Vidal instances. The comparative analysis of Local and Global execution modes on Vidal benchmarking datasets reveal a fundamental shift in heuristic efficacy, directly tied to the availability of depot-flexible refinement. Under the Local mode, where each route is improved independently against a fixed depot assignment, the performance hierarchy is relatively compressed, with the GRASP metaheuristic achieving the lowest average distances of 114.24 and 109.48 for Vidal (a) and (b), respectively. Notably, the simpler, greedier heuristics like GS and GR also deliver strong results in this setting, while the more complex EALNS framework struggles, yielding average distances that are noticeably higher (131.40 and 121.10). This suggests that in a constrained, route-by-route context, the intensive search of EALNS is less effective without the ability to re-optimize the depot choice, and the aggressive local search of GRASP is better suited to fully exploit a fixed-depot neighborhood.

The transition to the Global depot-flexible execution mode fundamentally inverts this performance ranking. By allowing the simultaneous reassignment of customers to different depots during the refinement process, the Global mode unlocks the true potential of the EALNS heuristic. EALNS decisively outperforms all other methods, achieving average distances of 137.58 and 128.89 on Vidal (a) and (b) respectively, representing a marginal increase compared to its Local-mode results and establishing a clear dominance over GRASP by a margin of more than 30 average distance units. This indicates that EALNS's adaptive, ruin-and-recreate search paradigm is more powerful at navigating the expanded, multi-depot solution landscape, successfully coordinating inter-depot customer movements to find structurally superior configurations that GRASP's constructive approach cannot reach. Similar to findings on Cordeau instances, the simplest heuristics like B2, F2, and FR, experience an improvement in their relative performance in the Global mode, which shows their greedy logic can be effective in large-scale routing problems re-optimization task.

The impact of the execution mode is therefore not merely a uniform improvement but an alteration of the heuristic landscape. Methods like SA2 and Tabu Search exhibit consistent but unspectacular performance across both modes, scaling roughly proportionally with the increased search potential. However, the differential between Local GRASP and

Table 8
Impact of time budget on heuristic refinement performance. The table presents the sequential execution results for selected heuristics on large-scale Vidal (a) instances under a range of time budgets. The best distance achieved within each budget and instance is shown, illustrating the convergence behavior and time-solution quality trade-off of each method.

	1 min					
Vidal (a)	FR	GR	GS	Tabu	GRASP	EALNS
pr11	90.57	104.38	83.61	135.74	54.47	84.77
pr13	173.28	214.65	195.42	227.54	89.22	139.33
pr15	271.20	319.92	303.34	326.02	124.98	197.48
pr16	307.37	359.87	350.02	362.95	145.99	231.65
	5 min					
pr11	67.82	71.54	69.91	126.83	54.38	84.45
pr13	147.67	187.78	151.87	221.80	89.22	135.27
pr15	240.65	304.06	258.85	323.53	124.98	191.76
pr16	280.74	350.27	318.57	360.37	145.99	209.33
	15 min					
pr11	58.90	65.73	69.91	119.47	54.38	82.96
pr13	126.70	155.96	123.44	214.62	89.22	132.23
pr15	212.34	275.01	210.97	319.42	124.98	187.34
pr16	254.99	328.55	270.22	357.73	145.99	205.00
	20 min					
pr11	56.27	65.31	69.91	118.60	54.38	82.96
pr13	121.23	146.62	117.94	212.77	89.22	132.23
pr15	205.51	263.00	196.64	316.31	124.98	187.34
pr16	247.02	319.82	255.58	356.61	145.99	205.00
	30 min					
pr11	54.73	62.60	68.73	109.31	54.38	82.29
pr13	113.49	128.75	109.81	208.69	89.22	131.75
pr15	196.01	241.60	179.49	313.52	124.98	186.48
pr16	236.34	305.43	234.01	354.04	145.99	203.26

Table 9
Ablation study quantifying the contribution of core KGCAF components. The table compares the full framework against a "Heuristics Only" baseline (no decomposition or GAT) and a "Decomposition + Heuristics" variant (no GAT) on select Vidal (a) instances in parallel mode. For each configuration, it reports the total distance (D), runtime (T), and time-window penalty (TP), highlighting the crucial role of the learning-based initialization.

	Heuristics Only												KGCAF (Best)
Vidal (a)		B2	F2	FR	GR	GS	Sh2	Tabu	SA2	RCL	GRASP	EALNS	
pr11	D	48.38	91.38	107.45	90.84	73.63	78.27	145.01	73.59	107.77	43.33	123.34	68.73
	T	1820.23	1820.55	1807.66	1830.02	1817.15	1822.69	1470.01	32.31	25.84	4537.40	1801.83	686.60
	TP	1065.12	1272.03	1434.12	1317.85	1245.89	1212.39	2403.26	1273.26	1540.86	969.11	43.18	26.06
pr13	D	238.62	244.70	265.71	290.02	229.91	247.52	290.02	171.12	171.12	103.36	196.82	110.58
	T	1870.67	1852.46	1810.56	1920.23	1896.68	1870.58	1817.92	92.16	92.16	5538.85	1803.28	1812.95
	TP	4994.34	5140.95	5654.97	6398.78	5291.90	5188.01	7306.14	3977.52	4258.84	2898.25	77.27	47.08
pr15	D	425.83	386.66	404.57	448.10	414.18	385.37	433.24	231.88	244.97	190.37	276.94	128.96
	T	1904.64	1836.75	1833.29	2107.11	2040.19	1838.58	1819.13	54.33	240.17	5744.93	1825.31	3933.36
	TP	12415.63	11403.64	12064.73	13769.85	12745.63	11224.48	14361.68	14361.68	8465.82	6507.31	112.89	84.43
pr16	D	502.07	443.61	468.13	517.76	489.62	451.71	505.37	281.61	281.41	227.80	310.38	147.62
	T	2120.34	1850.71	1847.26	2214.15	4532.49	1892.04	1825.57	58.66	377.18	5853.26	1841.92	3864.40
	TP	16837.42	15027.44	16108.61	18280.57	17239.15	15351.81	18896.47	11167.24	10721.52	8707.09	129.10	94.24
Decomposition + Heuristics													
		B2	F2	FR	GR	GS	Sh2	Tabu	SA2	RCL	GRASP	EALNS	
pr11	D	41.00	41.31	40.65	37.70	40.39	40.89	39.50	41.70	49.29	37.15	39.99	68.73
	T	2.51	1106.43	1829.96	1863.83	1835.69	2146.27	1864.62	36.63	65.14	4259.03	1801.30	686.60
	TP	81.53	79.59	76.68	76.48	79.68	80.04	91.14	80.81	98.07	68.18	97.37	26.06
pr13	D	61.07	62.39	60.59	58.53	60.37	62.48	59.19	61.97	74.35	59.49	59.38	110.58
	T	4.20	1824.95	1877.59	1854.95	1904.44	1840.29	2232.65	56.23	323.53	5795.46	1807.86	1812.95
	TP	127.88	127.01	136.70	141.72	137.73	127.18	150.62	128.42	154.92	120.50	153.93	47.08
pr15	D	78.02	79.52	75.72	74.98	76.40	79.48	75.13	81.38	105.82	84.74	75.04	128.96
	T	5.11	1814.95	1877.12	2522.53	2220.07	2120.81	2492.37	57.48	615.35	6077.37	1808.72	3933.36
	TP	171.92	173.82	204.95	210.73	200.78	174.74	216.59	175.63	225.59	173.56	217.80	84.43
pr16	D	89.60	90.57	86.00	84.06	85.91	90.69	84.92	93.52	117.65	98.79	84.83	147.62
	T	11.53	1824.43	1881.45	2667.09	2218.04	2190.55	2869.74	69.02	761.31	6284.29	1841.41	3864.40
	TP	190.00	194.97	224.32	230.76	225.39	196.17	235.84	194.49	238.88	202.28	236.29	94.24

Table 10
Performance of the KGCAF framework with depot-flexible global heuristic refinement on the Cordeau benchmark. The table reports the best total distance and average distance ($\bar{D}$) and runtime ($\bar{T}$), with the best instance-wise and overall average results in bold.

Instance	B2	F2	FR	GR	GS	Sh2	SA2	Tabu	RCL	GRASP	EALNS
pr01	8.17	8.17	**6.16**	6.42	6.56	8.17	8.35	6.23	8.95	6.40	6.61
pr02	15.71	15.68	11.68	**10.42**	11.29	15.81	16.36	12.46	17.18	10.56	11.06
pr03	24.81	24.80	24.05	22.92	17.65	24.81	26.90	24.16	28.87	17.47	**17.32**
pr04	34.21	34.21	35.76	36.76	28.73	34.19	33.48	35.24	37.17	27.59	**22.63**
pr05	39.27	39.22	39.86	42.68	37.95	39.22	41.33	39.93	42.24	33.24	**30.66**
pr06	47.13	47.45	50.89	52.56	49.00	47.45	48.87	50.56	49.07	42.12	**32.40**
pr07	10.71	10.73	8.34	**7.82**	8.03	10.71	11.68	8.12	11.44	8.30	8.25
pr08	22.82	22.82	20.93	19.66	**16.53**	22.82	23.30	20.30	24.37	16.55	16.79
pr09	30.68	31.00	32.36	34.11	29.28	30.90	33.78	32.28	35.04	26.09	**23.77**
pr10	47.09	47.09	49.88	54.72	51.03	47.09	51.93	50.29	52.96	39.99	**32.57**
pr11	8.23	8.32	6.29	6.28	6.60	8.32	7.95	**6.07**	8.58	6.33	6.51
pr12	16.48	16.47	12.41	10.85	11.23	16.47	16.55	11.96	17.94	**10.39**	10.96
pr13	26.73	26.60	23.68	23.86	17.94	26.60	26.49	24.26	29.20	**17.50**	18.66
pr14	34.68	34.72	34.88	36.30	29.68	34.72	34.72	34.73	37.26	28.79	**23.26**
pr15	38.97	39.05	39.56	42.27	38.63	39.05	43.53	38.98	41.79	31.60	**29.47**
pr16	48.35	48.17	51.77	54.65	50.67	48.18	49.19	50.57	52.53	42.05	**32.19**
pr17	10.95	10.93	7.76	7.93	7.95	10.87	11.29	**7.64**	11.35	7.66	8.04
pr18	24.54	24.49	22.33	20.31	16.41	24.50	24.77	20.18	26.13	**16.30**	16.61
pr19	30.15	30.39	31.66	32.19	27.69	30.39	33.48	31.96	34.65	25.16	**23.98**
pr20	45.46	45.39	48.63	52.02	47.61	45.38	50.72	47.82	50.98	38.98	**35.24**
$\bar{D}$	28.26	28.29	27.94	28.74	25.52	28.28	29.73	27.69	30.88	22.65	**20.35**
$\bar{T}$	0.23	358.77	1709.06	1607.31	1398.35	637.46	9.96	1629.40	29.32	3742.46	3376.16

Table 11
Performance of the KGCAF framework with depot-flexible global heuristic refinement on the large-scale Vidal (a) and Vidal (b) datasets. The table shows best total distance and the average distance ($\bar{D}$) and runtime ($\bar{T}$), highlighting the efficacy of powerful metaheuristics in this unrestricted setting.

Vidal(a)

Instance	B2	F2	FR	GR	GS	Sh2	SA2	Tabu	RCL	GRASP	EALNS
pr11	112.81	112.83	129.74	140.27	136.70	112.83	105.34	127.02	131.15	100.07	**89.33**
pr12	149.20	151.05	179.88	186.93	183.88	150.68	146.94	174.72	171.44	**133.99**	119.18
pr13	186.07	194.89	216.10	228.50	227.94	194.97	185.84	217.55	208.60	163.25	**137.63**
pr14	246.35	260.93	275.15	292.06	291.18	265.21	232.24	273.27	266.17	238.02	**172.30**
pr15	257.89	302.73	314.76	325.24	323.86	302.10	268.90	319.90	296.93	267.28	**199.47**
pr16	292.62	339.87	353.34	363.74	362.45	337.20	308.94	345.23	333.28	290.03	**219.13**
pr17	103.43	103.10	115.23	122.50	118.40	103.04	99.87	108.23	114.98	87.10	**81.04**
pr18	144.98	145.55	158.81	177.28	173.40	145.37	146.56	160.71	165.68	121.33	**116.87**
pr19	195.94	214.19	234.20	242.00	240.55	212.68	201.59	215.43	224.48	173.90	**153.72**
pr20	228.39	259.98	285.55	291.12	290.01	260.15	250.60	288.74	262.04	235.87	**174.10**
pr21	86.35	86.86	100.24	102.56	100.20	86.96	92.08	99.51	101.48	83.95	**75.32**
pr22	124.12	127.20	150.01	154.37	153.71	127.24	136.93	146.52	151.09	122.45	**108.11**
pr23	166.49	183.16	201.67	208.25	207.58	183.35	182.36	201.28	201.32	167.54	**142.37**
$\bar{D}$	176.51	190.95	208.82	218.06	216.14	190.91	181.40	206.01	202.20	168.06	**137.58**
$\bar{T}$	1.19	1795.46	1841.02	10041.59	4365.53	1876.44	39.94	1835.17	1204.68	6689.42	3674.90

Vidal(b)

Instance	B2	F2	FR	GR	GS	Sh2	SA2	Tabu	RCL	GRASP	EALNS
pr11	112.81	112.83	129.86	140.27	136.70	112.94	104.39	127.02	129.91	99.96	**76.57**
pr12	149.20	151.05	179.94	186.93	183.88	151.05	151.18	174.72	171.01	132.96	**101.23**
pr13	186.07	195.01	216.35	228.50	227.94	195.31	188.18	217.68	211.40	169.83	**124.87**
pr14	246.35	261.37	275.15	292.06	291.18	265.27	243.58	273.27	266.08	234.49	**153.76**
pr15	257.89	302.73	314.76	325.24	323.86	303.06	270.84	319.90	294.93	261.99	**190.99**
pr16	292.62	339.87	353.34	363.74	362.45	337.30	309.75	345.23	335.85	298.08	**204.01**
pr17	103.43	103.10	115.23	122.50	118.40	103.10	104.64	108.25	115.93	86.30	**75.98**
pr18	144.98	145.68	158.71	177.28	173.40	145.63	148.77	160.71	165.14	127.66	**106.91**
pr19	195.94	214.19	233.98	242.00	240.55	212.51	207.98	215.43	222.38	186.43	**140.35**
pr20	228.39	259.98	285.55	291.12	290.01	260.29	246.00	288.74	272.29	235.90	**172.48**
pr21	86.35	86.86	100.24	102.56	100.20	86.85	90.94	99.51	100.41	83.05	**75.96**
pr22	124.12	127.24	150.01	154.37	153.71	127.24	141.38	146.52	149.66	120.31	**105.69**
pr23	166.49	183.21	201.67	208.25	207.58	183.22	186.33	201.28	197.85	172.04	**146.75**
$\bar{D}$	176.51	191.01	208.83	218.06	216.14	191.06	184.15	206.02	202.53	169.92	**128.89**
$\bar{T}$	1.19	1796.18	1841.75	10050.71	4372.41	1880.60	40.12	1832.61	1221.70	6691.39	3600

Global EALNS is the defining result of this comparison. It demonstrates that the potency of a metaheuristic can be highly conditional on the decision scope. Particularly, GRASP's local search framework outperformed at optimizing fixed assignments, while EALNS's global search dominated when granted the authority to restructure the entire fleet-to-depot assignment. This reveals a critical design principle for large-scale multi-depot problems; the choice of refinement heuristic must be intrinsically linked to the execution mode, with the most powerful, globally-oriented search operators only becoming fully effective when empowered with depot-flexibility (see Table 11).

4.5.7. Comparative analysis: Unscaled local vs. Global heuristic refinement

Here, we present the results on unscaled vrp benchmarking instances, comparing the performance of the proposed framework with

Table 12
Comparative analysis of unscaled local vs. global heuristic refinement on the Cordeau benchmark. The table reports the total distance computed on the original, unscaled problem instances for both the depot-specific (Local) and depot-flexible (Global) execution modes. The average unscaled distance ($\bar{D}$) and runtime ($\bar{T}$) are provided, with the best result per instance in bold.

Instance	Unscaled_Local								Unscaled_Global							
	B2	F2	GR	GS	Sh2	RCL	GRASP	EALNS	B2	F2	GR	GS	Sh2	RCL	GRASP	EALNS
pr01	1139.90	1125.79	1159.63	1112.55	1134.67	1348.17	1126.47	**891.54**	1191.86	1124.46	1195.88	1244.63	1063.21	1632.43	1078.37	**861.32**
pr02	2217.83	2213.71	2160.59	2376.68	2127.01	2942.18	2158.85	**1386.63**	2363.05	2162.70	1853.92	1997.23	2127.49	3288.75	1839.57	**1275.82**
pr03	3366.22	3314.60	3262.21	3580.02	3353.91	4624.85	3346.39	**1861.22**	3811.69	3762.70	3765.95	3073.01	3659.54	5655.91	2759.13	**1799.37**
pr04	4221.26	4044.63	3921.39	4125.02	4063.24	6325.46	3800.32	**2102.97**	4722.89	4505.57	6326.95	4788.30	4470.43	7426.61	3629.29	**2124.34**
pr05	4660.25	4468.22	4443.70	4960.84	4506.87	6840.05	4495.11	**2495.74**	5452.10	5335.83	7686.21	6761.28	5366.97	8441.30	4691.79	**2435.07**
pr06	6005.09	6226.61	5451.67	6126.28	5857.01	9147.64	5697.32	**2770.43**	7068.51	6908.82	9788.66	9071.45	6860.45	10609.71	6384.50	**2752.75**
pr07	1644.65	1670.62	1616.84	1611.95	1666.92	1566.07	1554.14	**1138.01**	1395.31	1419.93	1431.30	1374.19	1427.25	2279.63	1362.54	**1018.37**
pr08	3183.05	3035.78	2967.15	3132.74	3033.28	3926.81	2944.60	**1709.45**	2949.83	2808.96	3097.33	2652.32	2811.62	4702.10	2309.77	**1658.62**
pr09	3971.08	4089.35	3968.92	4098.85	4021.51	5125.40	3843.12	**2207.06**	4268.60	4167.28	6179.26	5057.30	4085.25	6962.39	3757.76	**2243.65**
pr10	5551.41	5464.08	5506.90	5839.88	5558.58	7885.16	5319.00	**2943.20**	6139.95	6628.60	9829.59	9089.67	6619.32	10260.72	5695.94	**2899.19**
pr11	1127.39	1152.67	1200.32	1142.02	1111.86	1409.37	1125.56	**891.54**	984.58	964.25	989.80	1041.49	990.57	1766.20	962.24	**861.32**
pr12	2242.67	2223.54	2154.14	2335.31	2263.58	2661.23	2169.92	**1380.19**	2168.54	2122.81	1546.55	1799.79	2101.34	3511.08	1500.96	**1275.78**
pr13	3490.87	3415.41	3344.21	3528.94	3435.99	4905.01	3246.54	**1860.53**	3480.99	3459.07	3198.17	2929.54	3513.16	5383.95	2430.82	**1794.60**
pr14	4135.17	4039.91	3818.20	3899.73	3950.12	5291.29	3791.04	**2099.36**	4123.63	4064.52	5568.10	4307.44	4037.51	6999.95	3185.53	**2070.57**
pr15	4563.75	4760.56	4533.33	5080.78	4609.74	7497.34	4452.53	**2514.42**	5151.09	5120.20	7276.52	6311.59	5169.31	7973.97	4085.61	**2415.28**
pr16	5662.13	5889.39	5591.77	6075.68	5936.19	8995.79	5445.68	**2741.87**	6290.25	5969.28	9375.88	8677.72	5981.96	9752.54	5614.40	**2756.11**
pr17	1699.70	1674.95	1584.25	1739.59	1623.51	1698.91	1576.97	**1196.98**	1399.38	1377.62	1320.52	1463.30	1375.81	2324.16	1291.85	**1077.33**
pr18	2994.72	3040.39	2875.13	3020.23	3080.04	3761.12	2877.01	**1703.54**	2775.30	2853.61	2525.47	2345.25	2838.79	5200.75	2246.92	**1670.86**
pr19	3983.47	3850.72	3830.54	4260.49	3985.73	5389.49	3867.78	**2201.46**	3676.37	3683.82	5180.17	4242.82	3679.72	6334.89	3125.27	**2214.78**
pr20	5614.49	5633.80	5337.83	5511.33	5449.06	8628.06	5383.78	**2935.63**	5744.04	5494.58	9286.11	8319.52	5490.88	9527.60	5220.65	**2879.10**
$\bar{D}$	3573.76	3566.74	3436.44	3677.95	3538.44	4998.47	3411.11	**1951.59**	3757.90	3696.73	4871.12	4327.39	3683.53	6001.73	3158.65	**1904.21**
$\bar{T}$	35.16	323.48	76.44	33.59	484.54	0.32	1795.82	2809.80	2.68	906.94	1438.97	1302.62	1235.06	12.48	4107.36	3157.67

different heuristics in Local and Global heuristic refinement modes. The full set of results for all evaluated methods is provided in the supplementary material.

Cordeau instances. Table 12 reports the unscaled total distance and runtime for evaluated heuristics under Local (depot-specific) and Global (depot-flexible) refinement on the 20 Cordeau instances.

The most prominent result is the consistent dominance of EALNS, which achieves the lowest total cost on every single instance with average total distance of $\bar{D} = 1951.59$ (Local) and $\bar{D} = 1904.21$ (Global). This superiority stems from a powerful synergy between the GAT-based initialization and EALNS's adaptive destroy-and-repair mechanism. The GAT policy constructs initial routes that are topologically coherent and strictly constraint-feasible. This structured starting point allows EALNS to deploy its portfolio of removal and insertion heuristics to focus on cost-reducing reconfigurations rather than feasibility repair, exploiting large-scale perturbations that are guided by the search history and dynamically adapted to the instance structure.

Comparing execution modes, the Global mode yields a slightly better overall average, but this advantage is not uniform. Smaller instances and those with six depots benefit more from Global refinement, where cross-depot customer reassignments can correct suboptimal assignments from the initial decomposition. For certain mid-range instances (e.g., pr04, pr19), Local mode performs marginally better, indicating that the expanded search space of Global mode can occasionally introduce temporal penalties that outweigh distance savings when the penalty landscape is complex. The modest differences confirm that the profile-based decomposition provides a reliable structural prior. Among lighter-weight operators, an interesting performance inversion occurs: GreedyRelocate outperforms GreedySwap in Local mode ($\bar{D} = 43436.4$ vs. 3677.95), yet this reverses dramatically in Global mode ($\bar{D} = 4871.12$ vs. 4327.39). This behavior can be attributable to the cross-depot operator moves in Global mode creating a volatile neighborhood where swap operators, which maintain route cardinality, provide more stable exploration than the cascading effects of unconstrained relocations. The runtime analysis reveals a clear efficiency-quality frontier. Simpler operators like B2Opt ($\bar{T} = 35.16$s) and GreedyRelocate ($\bar{T} = 76.44$s) offer rapid solutions at a significant quality premium, while EALNS ($\bar{T} = 2809.80$s) delivers near-optimal quality for offline planning scenarios.

Vidal instances. The results in Table 13 demonstrate a clear and consistent hierarchy in solution quality for the large-scale Vidal instances. Most notably, the EALNS heuristic, when coupled with the KGCAF initialization, decisively outperforms all other methods in both Local and Global execution modes. Across all 26 Vidal (a) and (b) instances, EALNS achieves the lowest unscaled cost, yielding average distances of $\bar{D}$ of 8880.84 (Local) and 9537.14 (Global) for Vidal (a), and $\bar{D}$ of 8373.05 (Local) and 9643.95 (Global) for Vidal (b). This dominant performance underscores the efficacy of integrating a powerful metaheuristic with the structured solution space provided by the KGCAF's profile-based decomposition and initialization. The EALNS algorithm effectively exploits the initial solution's feasibility and quality, using large-scale destruction and repair operations to escape local optima and navigate the constrained search space in a way that simpler local search heuristics cannot.

A pivotal finding is the stark performance contrast between the two execution modes. The Local refinement mode consistently and significantly outperforms the Global mode for nearly all heuristics. This validates a core principle that the profile-based decomposition effectively captures the spatial and temporal structure of the problem, which is diluted or lost when making cross-depot moves in the Global mode. By relaxing depot assignments, the Global search faces an exponentially larger and less constrained neighborhood, making it harder for heuristics to converge to high-quality solutions, especially under time budgets. This effect is particularly pronounced for the Greedy Relocate (GR) and Greedy Swap (GS) heuristics, whose Global performance dramatically degrades (e.g., on pr11a, GR jumps from 10585.83 to 23306.72), indicating their greedy selection criteria are overwhelmed by the vast action space of an unrestricted multi-depot problem.

In contrast, the Local mode's success stems from the structured decomposition that confines each heuristic to a well-defined sub-problem. The profile-based initialization already establishes high-quality, feasible, and depot-centric routes, which creates a beneficial inductive bias. The local heuristics then operate as highly effective exploitation operators, making intra- and inter-route improvements within a depot's pre-allocated region without being confounded by computationally expensive and often fruitless inter-depot exchanges. The Local mode, therefore, represents a superior trade-off, leveraging the full capacity of advanced metaheuristics like EALNS within the KGCAF's scalable and parallelizable decomposition, leading to state-of-the-art results.

Table 13
Comparative analysis of unscaled local vs. global heuristic refinement on the Vidal (a) and Vidal (b) benchmark. The table reports the total distance computed on the original, unscaled problem instances for both the depot-specific (Local) and depot-flexible (Global) execution modes. The average unscaled distance ($\bar{D}$) and runtime ($\bar{T}$) are provided, with the best result per instance in bold.

	Unscaled_Local							Unscaled_Global						
Vidal (a) Instance	B2	F2	GR	GS	Sh2	GRASP	EALNS	B2	F2	GR	GS	Sh2	GRASP	EALNS
pr11	9090.00	13003.38	10585.83	10404.97	12781.00	11200.64	**5560.70**	12321.61	14333.49	23306.72	21862.90	14308.12	12751.53	**5699.73**
pr12	16975.52	20311.24	21985.50	18173.57	20500.37	19140.73	**7169.80**	16952.68	24675.94	32085.63	31154.29	24548.68	15085.72	**7484.52**
pr13	26373.45	29138.50	31568.93	29535.46	28855.53	26358.19	**8923.97**	20429.81	33689.79	40411.19	39476.98	33513.45	27077.03	**9613.23**
pr14	39541.67	36630.28	40171.66	39306.15	36081.98	33426.01	**10685.01**	25377.18	42488.26	48351.29	47593.73	42543.62	34112.03	**11739.04**
pr15	49388.59	43791.48	50264.17	49813.02	43472.80	42194.19	**12802.81**	29712.47	51795.37	56350.61	56242.07	51899.49	43591.04	**13378.26**
pr16	59321.96	53326.22	59340.89	57673.89	52733.21	49613.04	**14259.11**	33781.18	60073.76	64627.46	64363.65	60186.16	49629.75	**16887.40**
pr17	9114.44	12014.35	10440.09	10159.73	11910.03	10269.93	**5294.95**	11171.28	14155.64	20545.95	19373.87	13954.37	11869.46	**5312.91**
pr18	16705.03	18639.44	20024.40	19410.44	19334.77	17983.03	**7376.37**	16238.56	24506.87	29393.85	28689.28	24387.71	16764.70	**7679.42**
pr19	34196.76	31198.11	34030.27	34679.15	30717.69	28066.16	**10062.71**	22661.07	36830.11	41353.68	40914.74	36460.12	26548.50	**10663.10**
pr20	40465.84	38078.27	41060.34	41293.02	38355.49	35170.22	**11876.10**	28384.95	45763.10	50920.94	50755.70	45707.25	36932.95	**13109.64**
pr21	8576.85	8807.19	8145.92	9087.60	8796.56	7846.51	**5169.27**	9701.64	12427.27	17027.92	16500.43	12406.63	12546.62	**5072.96**
pr22	11488.36	14131.99	12430.12	12030.84	13952.16	11923.50	**7055.48**	14025.70	21279.81	25633.41	25595.01	21385.68	20385.53	**7438.08**
pr23	18435.05	21748.72	19803.94	20195.40	20714.79	16669.39	**9214.64**	18822.77	28975.74	33343.84	33276.77	28964.31	30212.69	**9904.52**
$\bar{D}$	26128.73	26216.86	27680.93	27058.71	26015.87	23835.50	**8880.84**	19967.76	31615.01	37180.96	36599.95	31558.89	25962.12	**9537.14**
$\bar{T}$	784.02	1471.17	1068.71	785.10	1654.20	3713.05	3620.52	14.23	1804.40	3837.45	2601.63	1804.32	5900.00	3667.05
Vidal (b)														
pr11	16864.93	22054.55	21096.83	19023.50	23141.19	21542.96	**5010.48**	10885.08	13640.19	22953.10	21280.77	13608.75	10722.94	**5023.353**
pr12	36673.82	37185.02	39445.37	38162.91	36718.11	32951.72	**6419.57**	14262.43	23717.36	31797.33	30482.22	23619.16	17174.60	**6805.084**
pr13	52780.43	50686.52	56208.72	53380.65	49743.21	46540.64	**8061.30**	18254.10	32987.70	40901.43	40338.81	32908.07	29811.12	**8772.325**
pr14	67012.00	63261.52	70898.68	68968.07	60801.94	58542.85	**10078.78**	22165.50	42150.68	49441.71	48705.97	42534.58	28668.43	**11989.95**
pr15	83041.78	74097.40	84655.25	85802.46	76145.84	73038.78	**12011.38**	26834.01	52087.01	57504.68	57254.87	52372.00	37585.89	**14911.03**
pr16	96899.46	88843.59	96639.44	97485.40	89744.68	83401.36	**13098.30**	30026.27	59957.88	65370.76	65087.26	59899.71	47351.04	**18829.93**
pr17	17945.35	22327.13	23267.02	20896.26	21374.45	20317.26	**5009.76**	10194.25	12038.19	19332.38	17721.10	12292.62	11433.06	**5185.324**
pr18	34142.73	35825.97	38144.51	36494.89	35761.72	33068.11	**6947.64**	14133.34	21886.48	29794.65	29083.70	21485.78	14818.86	**7834.977**
pr19	56843.51	50963.53	58769.55	59892.22	53509.06	46739.94	**9090.58**	19558.36	34389.27	40228.72	39559.29	34061.53	26474.28	**10447.26**
pr20	40450.20	38042.93	41060.34	41275.81	38844.70	34384.55	**11717.04**	24939.91	45557.28	50905.08	50720.46	45647.73	40352.42	**12348.17**
pr21	8744.02	8599.96	8095.07	9185.79	9011.08	8020.88	**5150.28**	8794.29	10974.93	16776.74	16242.25	11098.72	11220.90	**5048.218**
pr22	11504.46	14143.12	12406.85	12010.11	13624.29	11870.99	**6896.72**	12374.11	20555.35	25534.32	25239.52	20640.08	19833.21	**7745.255**
pr23	18208.18	21300.87	19746.51	19982.65	21191.66	16198.29	**9357.83**	16151.66	28226.31	33240.38	33123.89	28245.54	26725.28	**10430.41**
$\bar{D}$	41623.91	40564.01	43879.55	43273.90	40739.38	37432.18	**8373.05**	17582.56	30628.36	37213.94	36526.16	30647.25	24782.46	**9643.946**
$\bar{T}$	671.94	1335.99	1006.56	706.35	1628.57	3592.35	3613.358	12.50	1804.22	3887.83	2572.82	1804.34	5898.30	3647.535

4.5.8. Comparison with best-known solutions

Cordeau instances. Table 14 presents a rigorous evaluation of the proposed KGCAF framework (GAT+EALNS variant), against the established BKS from the literature on the 20 Cordeau benchmark instances (Bezerra, de Souza, & Souza, 2023; Lim et al., 2025). The comparison employs the unscaled objective function under the original instance configurations, thereby providing a direct assessment of absolute solution quality and optimality gaps. Two execution modes are analyzed: the Local mode, in which heuristic refinement is confined to the depot-level decomposition, and the Global mode, in which refinement operates without depot restrictions, allowing cross-depot customer reassignments.

The findings show that the framework achieves an average total cost of 1904.21, marking a −14.32% average improvement over the mean BKS of 2222.43. Critically, KGCAF sets a new best-found solution on 19 of the 20 instances, and the magnitude of the gain frequently exceeds 15%, with improvements surpassing 27% on instances such as pr02 and pr07, which points to a fundamental architectural advantage.

The source of this advantage can be attributed to the synergy between representation learning and heuristic search, a coupling that mitigates the core weaknesses of each paradigm. Purely learned construction methods, such as GAT-based policy networks, can capture global graph topology but often suffer from myopic sequential decoding, leading to structurally suboptimal route skeletons. Conversely, even powerful heuristics like EALNS are highly sensitive to their initial solution; a poor starting point can trap them in a local optima. KGCAF bridges this gap through a two-stage process of structured initialization adhering to the embedded logical feasibility rules, and heuristic refinement as an exploitative refinement engine that iteratively destructs and repairs this intelligent prior. This tight coupling allows the heuristic's local search operators to efficiently fine-tune the route structures, achieving a level of local optimality that the learning-based component cannot reach alone, while avoiding the poor local optima that often trap heuristics initialized from scratch. The framework, therefore, unifies the global pattern-recognition capabilities of graph attention with the local exploitation power of a metaheuristic, creating a hybrid that is far more effective than the sum of its parts.

A deeper analysis of the Local versus Global execution modes further illuminates the interplay between decomposition and solution quality. The Global mode, which relaxes depot-assignment constraints during the heuristic refinement, achieves the superior average gap of −14.32% and outperforms the Local mode on 14 out of 20 instances. This result reveals a critical design principle: while depot-level profile decomposition is indispensable for scalable learning and parallel construction, it inherently introduces a degree of rigidity by assigning customers to depots a priori. The Global refinement stage acts as a crucial corrective layer, allowing the heuristic search to reassign customers across depot boundaries, thereby rectifying sub-optimal decomposition decisions. The effect is dramatic on instances like pr17, where the Global mode (−12.85%) improves upon the Local mode (−3.18%) by over nine percentage points. The utility of this mechanism is further evidenced by the sole instance where the Local mode failed to surpass the BKS (pr15, +3.34%). The Global mode successfully resolved this shortfall, reaching a −0.73% improvement. This demonstrates that for certain problem geometries, the ability to transcend the initial decomposition is not merely beneficial but essential.

In conclusion, the results demonstrate that KGCAF is not merely a competitive approach but represents the new leading methodology for the Cordeau benchmark. The framework's success originates from its ability to generate an intelligent, feasibility-aware initial solution prior and then leverage a depot-flexible, globally-oriented search mechanism to fully exploit this prior, consistently discovering superior optima that

Table 14
Comparison of the proposed KGCAF framework (GAT+EALNS variant) against Best-Known Solutions (BKS) from the literature on the 20 Cordeau benchmark instances. The table reports the unscaled total distance and percentage gap for both Local and Global GAT+EALNS configurations. The bold entry per row shows the best solution achieved by our framework.

Instance	Best Known Solution			GAT_EALNS			
	Approach	Year	BKS	Local	Gap (%)	Global	Gap (%)
pr01	ITS	2004	1074.12	891.54	−17.00	**861.32**	−19.81
pr02	ITS	2004	1762.21	1386.63	−21.31	**1275.82**	−27.60
pr03	ITS	2004	2373.65	1861.22	−21.59	**1799.37**	−24.19
pr04	ELS-LNS	2024	2811.63	**2102.97**	−25.20	2124.34	−24.44
pr05	ELS-LNS	2024	2962.25	2495.74	−15.75	**2435.07**	−17.80
pr06	ELS-LNS	2024	3578.99	2770.43	−22.59	**2752.75**	−23.09
pr07	ITS	2004	1418.22	1138.01	−19.76	**1018.37**	−28.19
pr08	VTNS	2021	2096.73	1709.45	−18.47	**1658.62**	−20.89
pr09	HGSADC	2013	2712.56	**2207.06**	−18.64	2243.65	−17.29
pr10	HGSADC	2013	3465.92	2943.2	−15.08	**2899.19**	−16.35
pr11	IFA	2022	981.98	891.54	−9.21	**861.32**	−12.29
pr12	VTNS	2021	1464.5	1380.19	−5.76	**1275.78**	−12.89
pr13	VTNS	2021	2001.81	1860.53	−7.06	**1794.6**	−10.35
pr14	VTNS	2021	2195.33	2099.36	−4.37	**2070.57**	−5.68
pr15	HGSADC	2013	2433.15	2514.42	3.34	**2415.28**	−0.73
pr16	HGSADC	2013	2836.67	**2741.87**	−3.34	2756.11	−2.84
pr17	ITS	2004	1236.24	1196.98	−3.18	**1077.33**	−12.85
pr18	VTNS	2021	1788.18	1703.54	−4.73	**1670.86**	−6.56
pr19	HGSADC	2013	2261.08	**2201.46**	−2.64	2214.78	−2.05
pr20	HGSADC	2013	2993.31	2935.63	−1.93	**2879.1**	−3.82
$\bar{D}$			2222.43	1951.589	−12.19	**1904.212**	−14.32

were previously unattainable by either learning or heuristic methods in isolation.

Vidal instances. Table 15 presents the performance of the KGCAF framework against the best-known solutions for the Vidal (a) and Vidal (b) benchmark instances (Bezerra, de Souza, & Souza, 2023). A striking and consistent pattern emerges from this comparison, showing that the Local (depot-specific) execution mode significantly outperforms the Global (depot-flexible) mode across both datasets, and this performance divergence grows noticeably with instance scale.

The findings show that the Local mode achieves exceptional results on Vidal (a). It discovers new best-known solutions on 11 out of 13 instances, as indicated by the negative percentage gaps. The average improvement over the BKS is −7.23%, with substantial gains on instances like pr11a (−17.26%) and pr17a (−16.01%). This can validate the core design principle of KGCAF; profile-based decomposition creates a structured, localized solution space where heuristics can exploit region-specific patterns with high efficiency. By confining the search to coherent, depot-centric sub-problems, the heuristic refinement avoids the noise and complexity of cross-depot interactions, allowing it to converge more rapidly to near-optimal or even previously undiscovered optima within the given time budget. Moreover, the GAT policy, trained under the same feasibility-aware, profile-guided constraints, initializes routes that are already well-adapted to their local region. The subsequent heuristic search therefore begins from a high-quality, logically consistent starting point, enabling it to find deep improvements that purely global or randomly initialized methods would likely miss.

In contrast, the performance of the Global mode degrades markedly, particularly as the problem size increases. On Vidal (a), the Global mode achieves a nearly neutral average gap of −0.37% against the BKS. Noteworthy, the mode performs competitively on smaller instances (e.g., pr11a, pr17a, pr21a) but generates large positive gaps on larger ones (e.g., +18.09% on pr16a, +9.58% on pr20a). This degradation is severely amplified on the Vidal (b) instances, where customer demands and time windows are tighter, making the problem more constrained. The Global mode's average gap increases to +27.18%, reaching an extreme +63.98% on the largest instance, pr16b.

This divergence highlights a key insight: while the Global mode has the theoretical advantage of inter-depot flexibility, this freedom becomes a liability in large, highly constrained problems. Without the guidance of depot profiles, the heuristic refinement in a unified search space must navigate a combinatorially explosive solution landscape. The search becomes computationally expensive and prone to getting trapped in suboptimal regions, particularly when tight constraints fragment the feasible space. Conversely, the Local mode's strength is precisely its ability to manage this complexity. The profiles act not just as a decomposition mechanism but as a knowledge-guided constraint firewall. They drastically reduce the effective problem size for each refinement process, transforming a single intractable global search into several tractable local ones. The fact that the Local mode maintains strong performance while the Global mode fails on Vidal (b) powerfully demonstrates that for complex, constrained MDVRPTW, intelligent problem decomposition is more effective than granting an optimizer complete freedom.

Overall, the comparison with the BKS reveals that the KGCAF local mode can set a new state-of-the-art on the Vidal (a) benchmark, proving the effectiveness of its hybrid learning-and-search paradigm over specialized heuristics. Additionally, it shows that the scalable decomposition is paramount for large-scale routing problems. The systematic and predictable failure of the Global mode on larger, more constrained instances underscores that the framework's principal source of strength is its hierarchical, profile-based decomposition. This design choice is not merely a matter of parallel convenience; it is fundamental to achieving scalable optimization in the face of complex real-world constraints.

5. Discussion and limitations

The experimental results demonstrate that KGCAF achieves superior solution quality and computational efficiency across diverse benchmark instances. Beyond reporting these empirical gains, this section provides a deeper analysis of the architectural principles underlying KGCAF's performance, contrasting its design choices with those of pure reinforcement learning and prior hybrid methods. We further examine the challenges inherent to multi-agent vehicle routing and how the proposed framework addresses them, and we conclude by identifying current limitations that outline directions for future work.

Table 15
Comparison of the proposed KGCAF framework (GAT+EALNS variant) against Best-Known Solutions (BKS) from the literature on the Vidal (a) and Vidal (b) instances. The table reports the unscaled total distance and percentage gap for both Local and Global GAT+EALNS configurations. The bold entry per row shows the best solution achieved by our framework.

Instance	Vidal (a)					Vidal (b)				
		GAT_EALNS					GAT_EALNS			
	BKS	Local	Gap (%)	Global	Gap (%)	BKS	Local	Gap (%)	Global	Gap (%)
pr11	6720.71	**5560.70**	−17.26	5699.73	−15.19	**4839.44**	5010.48	3.53	5023.35	3.80
pr12	8179.80	**7169.80**	−12.35	7484.52	−8.50	**6063.26**	6419.57	5.88	6805.08	12.23
pr13	9667.20	**8923.97**	−7.69	9613.23	−0.56	**7254.17**	8061.30	11.13	8772.33	20.93
pr14	11124.01	**10685.01**	−3.95	11739.04	5.53	**8732.29**	10078.78	15.42	11989.95	37.31
pr15	13013.97	**12802.81**	−1.62	13378.26	2.80	**10439.72**	12011.38	15.05	14911.03	42.83
pr16	14299.87	**14259.11**	−0.29	16887.40	18.09	**11483.22**	13098.30	14.06	18829.93	63.98
pr17	6304.30	**5294.95**	−16.01	5312.91	−15.73	**4806.01**	5009.76	4.24	5185.32	7.89
pr18	8308.32	**7376.37**	−11.22	7679.42	−7.57	**6526.72**	6947.64	6.45	7834.98	20.04
pr19	10677.61	**10062.71**	−5.76	10663.10	−0.14	**8227.25**	9090.58	10.49	10447.26	26.98
pr20	11963.91	**11876.10**	−0.73	13109.64	9.58	**10325.80**	11717.04	13.47	12348.17	19.59
pr21	6260.53	5169.27	−17.43	**5072.96**	−18.97	**4866.57**	5150.28	5.83	5048.22	3.73
pr22	7985.37	**7055.48**	−11.64	7438.08	−6.85	**6488.50**	6896.72	6.29	7745.26	19.37
pr23	9937.43	9214.64	−7.27	**9904.52**	−0.33	**8523.41**	9357.83	9.79	10430.41	22.37
$\bar{D}$	9572.54	**8880.84**	−7.23	9537.14	−0.37	**7582.80**	8373.05	10.42	9643.95	27.18

5.1. Why KGCAF outperforms pure reinforcement learning approaches

The substantial and consistent performance gap between KGCAF and the pure RL baselines (GAT_RL, PDDQN, AC) originates from three architectural design choices that address fundamental limitations of end-to-end neural policies for constrained combinatorial optimization.

Explicit feasibility enforcement. Pure RL methods handle constraints implicitly through penalty terms in the reward function. This approach provides no feasibility guarantees during inference, can complicate credit assignment by confounding cost minimization with constraint satisfaction, and may require careful tuning of penalty coefficients that often vary across instance scales. KGCAF, in contrast, enforces capacity, time-window, and regional constraints through logical masking at every decision step. The feasible action set $\mathcal{F}_k^t$ acts as a constraint firewall, ensuring that the policy operates exclusively within the feasible solution space during both training and inference. This design eliminates the feasibility–optimality trade-off that penalty-based methods must navigate.

Structured decomposition for scalable representation learning. Pure attention-based architectures for VRPs typically attend over the full set of customers globally. As instance size grows, this global attention becomes both computationally expensive and representationally diffuse, the model must learn to ignore irrelevant long-range interactions while capturing local routing structure. KGCAF's profile-based decomposition partitions the problem into depot-centric subproblems, enabling each GAT instance to learn rich relational representations within a localized, semantically coherent customer subset. The resulting embeddings encode not just spatial proximity but also the shared constraint context (e.g., time-window compatibility, capacity interplay) that governs effective routing within a region.

Hybrid learning–search decoupling. Perhaps the most critical distinction is that KGCAF does not task the neural policy with producing final solutions. The GAT provides a structured, high-quality initialization; heuristic refinement then performs the local exploitation needed to reach a sharp local optimum. This decoupling acknowledges that neural networks excel at capturing global structural patterns but struggle with the precise local adjustments that define optimal VRP solutions. Prior hybrid approaches treat learning and search as loosely connected modules, with heuristics operating as black-box post-processors on neural outputs. KGCAF unifies them through shared logical predicates Φ_d and the knowledge-guided cost function, allowing the heuristic to operate on the same feasibility-aware representation space learned by the GAT. The ablation study (Table 9) confirms this synergy: removing the GAT initialization (*Decomposition + Heuristics*) causes a dramatic increase in time-window violations, while the full framework maintains both low distance and low temporal penalties.

5.2. Multi-agent coordination: Challenges and benefits

Formulating the MDVRPTW as a multi-agent system introduces well-known coordination challenges: non-stationarity from concurrent agent decisions, credit assignment across agents serving overlapping or adjacent regions, and the computational burden of centralized coordination at scale. KGCAF addresses these through a combination of structural decomposition and lightweight synchronization.

The profile-based decomposition assigns each customer to exactly one depot, transforming the multi-agent problem into largely independent single-depot subproblems. Within each depot, vehicles operate on a fixed customer pool with well-defined feasibility predicates. This eliminates agent-level non-stationarity during route construction, as no two agents from different depots compete for the same customer. Moreover, global consistency is maintained through asynchronous profile messages that communicate only modified state entries (e.g., updated regional boundaries, visited customer flags). This avoids the overhead of full state sharing while ensuring that cross-depot constraints, such as customer coverage, are satisfied. The experimental results validate this design: parallel execution achieves solution quality comparable to sequential execution with reduced wall-clock time (Table 7), and the small standard deviations across random seeds confirm stable, reproducible coordination without the training instability commonly reported in fully decentralized MARL systems.

Beyond theoretical soundness, the multi-agent architecture yields two practical advantages. First, it enables parallel heuristic refinement across depots, with the overall runtime governed by the critical path rather than the sum of depot-level efforts. Second, the modularity of the design allows individual depot policies to be updated or replaced without retraining the entire system, facilitating deployment in dynamic logistics environments where depot configurations or fleet compositions may change over time.

5.3. Limitations

Despite its strong performance, KGCAF has several limitations that motivate future research.

- **Sensitivity to decomposition quality.** The profile-based decomposition is a hard assignment of customers to depots. While the Global refinement mode can partially correct suboptimal assignments, the Local mode, which achieves the best results on large-scale Vidal (b) instances, is entirely dependent on the initial decomposition quality. Developing learned or adaptive decomposition strategies that can be refined jointly with routing remains an open direction.

- **Static depot profiles.** The current profile representation captures spatial and temporal customer features but does not learn to adapt the profile structure itself during training. Incorporating learned profile embeddings that evolve based on observed routing patterns could further improve the quality of the decomposition and the resulting initial solutions.
- **Heuristic selection and time-budget allocation.** The choice of heuristic operator and the allocation of the refinement time budget are currently set globally. The experimental results show that different heuristics excel on different instances and under different time budgets. An adaptive mechanism that selects the most promising heuristic and allocates runtime based on instance characteristics or online performance feedback could improve robustness and efficiency.
- **Extension to dynamic and stochastic settings.** The current framework is evaluated on static benchmark instances. Real-world logistics involve dynamic customer arrivals, stochastic travel times, and evolving constraints. Extending the profile-based architecture to handle online replanning, while maintaining the feasibility guarantees provided by logical masking, represents an important direction for practical deployment.

6. Conclusion and future directions

In this work, we introduced the Knowledge-Guided Collaborative Attention Framework (KGCAF), a hybrid methodology that tightly integrates depot-level profile-based decomposition, logic-guided graph attention, and time-budgeted heuristic refinement for large-scale multi-depot vehicle routing. Our experimental evaluation demonstrates that KGCAF consistently outperforms state-of-the-art baselines, achieving up to an 86% reduction in total distance on Cordeau instances and a 74% reduction on larger Vidal instances, while strictly maintaining feasibility. A key finding is the critical role of the profile-based decomposition, which acts not as a limitation but as a vital scalability mechanism. On large, highly constrained instances, local depot-specific refinement decisively outperforms a global, depot-flexible search, establishing new best-known solutions on Cordeau and 11 of 13 Vidal (a) instances. This confirms that intelligent problem decomposition is paramount for navigating the combinatorial complexity of real-world logistics, with the learned GAT-based structural priors serving as an indispensable component that shapes a favorable optimization landscape for subsequent heuristics. Looking ahead, several directions may further extend the proposed framework. One promising avenue is the deeper integration of dynamic feasibility knowledge into the representation learning process, enabling node embeddings to adapt continuously to evolving operational constraints such as charging availability, traffic conditions, or stochastic demand. In addition, extending the depot-profile abstraction to support inter-depot cooperation and load rebalancing could improve global efficiency in highly dynamic logistics networks. Finally, incorporating online learning and transfer mechanisms would allow KGCAF to generalize across heterogeneous routing scenarios and rapidly adapt to previously unseen environments, further enhancing its applicability to real-world, large-scale vehicle routing systems.

CRediT authorship contribution statement

Narges Movahedkor: Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization. **Reza Shahbazian:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Formal analysis, Conceptualization. **Francesca Guerriero:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis.

Declaration of Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to improve the grammar, language, and readability of the manuscript. After using, the authors reviewed the content and take full responsibility for the final content of the published article.

Acknowledgments

This research was supported by a grant awarded by the "European Union – Next Generation EU" under the "PRIN 2022" project: SMOTION (ID2022EAECWJ), CUP H53D23002000006.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cie.2026.112282.

Data availability

The datasets and the source code developed in this study are available on GitHub at https://github.com/ShahbazianR/Scalable-KGCAF.git.

References

Ahamed, T., Zou, B., Farazi, N. P., & Tulabandhula, T. (2021). Deep reinforcement learning for crowdsourced urban delivery. *Transportation Research Part B: Methodological*, *152*, 227–257.

Ajao, Q., & Oludamilare, O. (2023). Safety challenges and analysis of autonomous electric vehicle development: Insights from on-road testing and accident reports. arXiv preprint arXiv:2305.12665.

Alcaraz, J. J., Losilla, F., & Caballero-Arnaldos, L. (2022). Online model-based reinforcement learning for decision-making in long distance routes. *Transportation Research Part E: Logistics and Transportation Review*, *164*, Article 102790.

Bdeir, A., Boeder, S., Dernedde, T., Tkachuk, K., Falkner, J. K., & Schmidt-Thieme, L. (2021). RP-DQN: An application of Q-learning to vehicle routing problems. In *KI 2021: advances in artificial intelligence: 44th german conference on AI, virtual event, September 27–October 1, 2021, proceedings 44* (pp. 3–16). Springer.

Bezerra, S. N., de Souza, S. R., & Souza, M. J. F. (2023). A general VNS for the multi-depot open vehicle routing problem with time windows. *Optimization Letters*, *17*(9), 2033–2063.

Bezerra, S. N., Souza, M. J. F., & de Souza, S. R. (2023). A variable neighborhood search-based algorithm with adaptive local search for the Vehicle Routing Problem with Time Windows and multi-depots aiming for vehicle fleet reduction. *Computers & Operations Research*, *149*, Article 106016.

Bogyrbayeva, A., Yoon, T., Ko, H., Lim, S., Yun, H., & Kwon, C. (2023). A deep reinforcement learning approach for solving the traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, *148*, Article 103981.

Bouanane, K., Amrani, M. E., & Benadada, Y. (2022). The vehicle routing problem with simultaneous delivery and pickup: A taxonomic survey. *International Journal of Logistics Systems and Management*, *41*(1–2), 77–119.

Eßer, J., Bach, N., Jestel, C., Urbann, O., & Kerner, S. (2022). Guided reinforcement learning: A review and evaluation for efficient and effective real-world robotics [survey]. *IEEE Robotics & Automation Magazine*, *30*(2), 67–85.

Guerriero, F., Macrina, G., Mosca, V., & Scalzo, E. (2025). The vehicle routing problem and integrated challenges in the perishable product supply chain. *Computers & Industrial Engineering*, Article 111428.

Hildebrandt, F. D., Thomas, B. W., & Ulmer, M. W. (2023). Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research*, *150*, Article 106071.

Hua, C., Berto, F., Son, J., Kang, S., Kwon, C., & Park, J. (2025). CAMP: Collaborative attention model with profiles for vehicle routing problems. arXiv preprint arXiv:2501.02977.

Lim, T. Y., Ng, X. J., Tan, C. J., & Lim, C. P. (2025). Stepwise monogamous pairing genetic algorithm method applied to a multi-depot vehicle routing problem with time windows. *Neural Computing and Applications*, *37*(16), 9639–9668.

Liu, F., Lu, C., Gui, L., Zhang, Q., Tong, X., & Yuan, M. (2023). Heuristics for vehicle routing problem: A survey and recent advances. arXiv preprint arXiv:2303.04147.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., & Laganà, D. (2017). The vehicle routing problem with occasional drivers and time windows. In *Optimization and decision science: methodologies and applications: ODS, sorrento, Italy, September 4-7, 2017 47* (pp. 577–587). Springer.

Mazumder, S., Liu, B., Wang, S., Zhu, Y., Yin, X., Liu, L., & Li, J. (2022). Knowledge-guided exploration in deep reinforcement learning. arXiv preprint arXiv:2210.15670.

Movahedkor, N., Shahbazian, R., & Guerriero, F. (2026). Hybrid knowledge-guided reinforcement learning with adaptive variable neighborhood search for dynamic multi-depot electric vehicle routing problems. *Expert Systems with Applications*, Article 132238.

Nazari, M., Oroojlooy, A., Snyder, L., & Takác, M. (2018). Reinforcement learning for solving the vehicle routing problem. *Advances in Neural Information Processing Systems*, *31*.

Pan, W., & Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(1), 405–422.

Santiyuda, G., Wardoyo, R., Pulungan, R., & Yu, V. F. (2024). Multi-objective reinforcement learning for bi-objective time-dependent pickup and delivery problem with late penalties. *Engineering Applications of Artificial Intelligence*, *128*, Article 107381.

Shahbazian, R., Ciacco, A., Macrina, G., & Guerriero, F. (2025). Knowledge-guided hybrid deep reinforcement learning for the dynamic multi-depot electric vehicle routing problem. *Computers & Operations Research*, Article 107217.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., & Van Woensel, T. (2023). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, *304*(3), 865–886.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, *12*.

Wang, S., Sun, W., & Huang, M. (2024). An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time windows. *Computers & Industrial Engineering*, Article 110122.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.

Zhang, K., Lin, X., & Li, M. (2023). Graph attention reinforcement learning with flexible matching policies for multi-depot vehicle routing problems. *Physica A. Statistical Mechanics and its Applications*, *611*, Article 128451.

Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, 321–384.

Zhang, S., Zhou, T., Fang, C., & Yang, S. (2024). A novel collaborative electric vehicle routing problem with multiple prioritized time windows and time-dependent hybrid recharging. *Expert Systems with Applications*, *244*, Article 122990.

Zhao, J., Mao, M., Zhao, X., & Zou, J. (2020). A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*, *22*(11), 7208–7218.

Zhou, H., Li, Y., Ma, C., Long, K., & Li, X. (2025). Modular vehicle routing problem: Applications in logistics. *Transportation Research Part E: Logistics and Transportation Review*, *197*, Article 104022.

Zhou, Y., Qu, C., Wu, Q., Kou, Y., Jiang, Z., & Zhou, M. (2024). A bilevel hybrid iterated search approach to soft-clustered capacitated arc routing problems. *Transportation Research Part B: Methodological*, *184*, Article 102944.

Zhu, Z., Tang, X., Qin, Y., Huang, Y., & Hashemi, E. (2023). A survey of lateral stability criterion and control application for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*.