

Multi-Relation Knowledge Graph-Guided Transformer with Objective-Aware Route Selection for Vehicle Routing Problems with Time Windows

Somkiat Kosolsombat¹, Chiabwoot Ratanavilisagul^{2*}

Data Science and Innovation Program, College of Interdisciplinary Studies, Thammasat University, Thailand¹
Faculty of Applied Science-Department of Computer and Information Science,
King Mongkut's University of Technology North Bangkok, Thailand²

Abstract—Vehicle routing problems with time windows require route construction methods that can reason jointly about spatial distance, vehicle capacity, service time, and customer time windows. Recent neural combinatorial optimization studies show that Transformer and graph-based models can learn routing policies, but many models still rely mainly on node features and do not expose which routing relations are responsible for feasibility and route quality. This study proposes a multi-relation knowledge graph guided Transformer framework for the vehicle routing problem with time windows. For each instance, a pairwise knowledge graph is built from spatial, temporal, capacity, slack, and depot identity relations. These relation vectors are projected into head-specific attention biases inside a pointer-style Transformer, while a feasibility mask enforces capacity and time window constraints during decoding. The framework is trained by imitation from OR-Tools teacher routes and evaluated on the 56 Solomon benchmark instances and on synthetic Solomon-style extensions. The original Transformer baseline decoded feasible routes for 54 of 56 Solomon instances, whereas the full knowledge graph guided model decoded feasible routes for all 56 instances. Additional controlled AM style, POMO style multistart, and edge-aware Transformer baselines improve route quality gaps on feasible instances but still leave one Solomon instance infeasible, highlighting the proposed framework's feasibility-oriented role. On three 120 instance synthetic runs, objective-aware route selection over relation-specialized models reached a mean feasible rate of 0.997 and reduced the mean distance gap by 125.418 Solomon coordinate distance units compared with the relation-free Transformer baseline. The contribution is therefore a diagnostic and extensible framework that integrates explicit routing knowledge into neural attention, evaluates relation ablations, and combines specialized route candidates according to the target operational objective.

Keywords—Vehicle routing problem with time windows; knowledge graph; Transformer; neural combinatorial optimization; relation aware attention; route selection

I. INTRODUCTION

The vehicle routing problem with time windows (VRPTW) is a central problem in logistics optimization. A fleet must serve geographically distributed customers while satisfying vehicle capacity and customer service intervals. The Solomon benchmark remains a standard controlled testbed for this setting [1]. Constraint and operations research solvers,

including OR-Tools, provide strong references because they represent routing constraints directly [2].

Learning-based solvers are attractive when repeated route construction, fast policy reuse, or learned route priors are required. Recent surveys describe rapid growth in neural combinatorial optimization, reinforcement learning, and learning to route methods [3]-[5]. However, a practical gap remains: many neural VRP models learn from node features without explicitly separating the spatial, temporal, capacity, and slack relations that drive VRPTW feasibility.

In this study, a pointer Transformer refers to an attention-based encoder-decoder route construction model whose output pointer distribution selects the next customer from the current set of feasible nodes. The model, therefore, predicts a sequence of customer visits rather than a fixed class label.

This work studies whether a knowledge graph (KG) can operate as a structured control layer for a Transformer routing model. Instead of treating the KG as a post hoc explanation, the proposed framework injects pairwise relation information directly into Transformer attention. The study is positioned as a framework and diagnostic analysis rather than as a claim that a compact neural model outperforms OR-Tools.

The contributions are: a multi-relation KG representation for VRPTW instances; a KG-guided pointer Transformer that converts relation vectors into attention bias; relation ablations and adaptive variants that reveal how relation groups affect decoded routes; and objective-aware route selection with multi-seed robustness evaluation.

The rest of the study is organized as follows: Section II reviews neural routing and knowledge guided routing studies. Section III states the research gap. Section IV presents the KG-guided framework, relation definitions, attention bias, feasibility mask, and implementation settings. Section V describes the experimental design, Section VI reports the results, Section VII discusses implications, and Section VIII and Section IX summarize limitations and conclusions.

II. RELATED WORK

Hybrid learning and optimization methods have become common in vehicle routing. Deep reinforcement learning has been used for column generation [6], neural construction heuristics have been embedded in large neighborhood search

*Corresponding author

[7], and combinatorial optimization enriched machine learning has been applied to dynamic VRPTW [8]. Multi-objective and multi-agent variants further expand the decision space beyond distance minimization [9]-[12].

Recent time window and electric vehicle studies use reinforcement learning and graph-structured data to model feasible actions, resource use, and operational constraints [13]-[15]. Transformer-based routing models use attention to represent customer interactions, including collaborative Transformer approaches, EFormer edge-based Transformers, and local global feature fusion Transformers [16]-[18]. GASE graph attention with edge fusion and symmetry preserving graph attention also suggest that edge information can improve routing representations [19], [20]. These models motivate relation-aware routing, but they do not isolate VRPTW spatial, temporal, capacity, slack, and identity relations as separate ablation controls inside a head-specific attention bias.

Broader generalization has been explored through equity-aware Transformers, cross-problem learning, foundation-style route models, mixture of experts, large-scale reinforcement learning, and route foundation models [21]-[26]. Knowledge-guided metaheuristics and hybrid reinforcement learning show that explicit domain knowledge can improve search behavior [27], [28]. Other recent routing research includes GCN-assisted search, hyperheuristic and VAE-based approaches, edge-based learning for VRPTW, and backhaul routing [29]-[32]. Closely related Thai VRPTW studies by Kosolsombat and Ratanavisagul examined modified ant colony optimization and applied deep reinforcement learning for time window routing [33], [34]. The present work differs by using a pairwise KG as a direct attention bias inside a Transformer and by testing relation groups through ablation and route selection.

III. RESEARCH GAP AND DESIGN RATIONALE

The literature indicates three limitations that motivate this study. First, many neural VRP models encode customers

primarily as node sequences. This is convenient for attention-based models, but VRPTW feasibility is inherently pairwise: whether customer j can follow customer i depends on travel time, service completion, waiting time, residual capacity, and both time windows. A model that only receives node attributes must infer these pairwise constraints indirectly from training examples.

Second, recent neural routing studies often report final route quality but provide less diagnostic evidence about which kinds of routing knowledge help. For a logistics decision maker, it is useful to know whether spatial proximity, temporal feasibility, capacity compatibility, or slack information is responsible for improved decoded routes. A relation ablation design, therefore, gives more insight than a single black box comparison.

Third, no single learned construction policy is expected to dominate all instance classes. Clustered, random, and mixed Solomon-style distributions have different structural properties. A relation setting that is useful for clustered customers may not be best for dispersed customers with tight windows. This motivates a framework in which relation-specialized models are treated as candidate route generators, and route validation is used before objective-aware selection.

IV. PROPOSED FRAMEWORK

Fig. 1 summarizes the proposed pipeline. Each VRPTW instance is converted into node features and a pairwise KG. OR-Tools teacher routes provide imitation labels. A pointer Transformer receives node features, while the KG contributes relation-specific attention bias. The decoder applies a hard feasibility mask, and a candidate pool of relation-specialized models is filtered and ranked by the chosen routing objective.

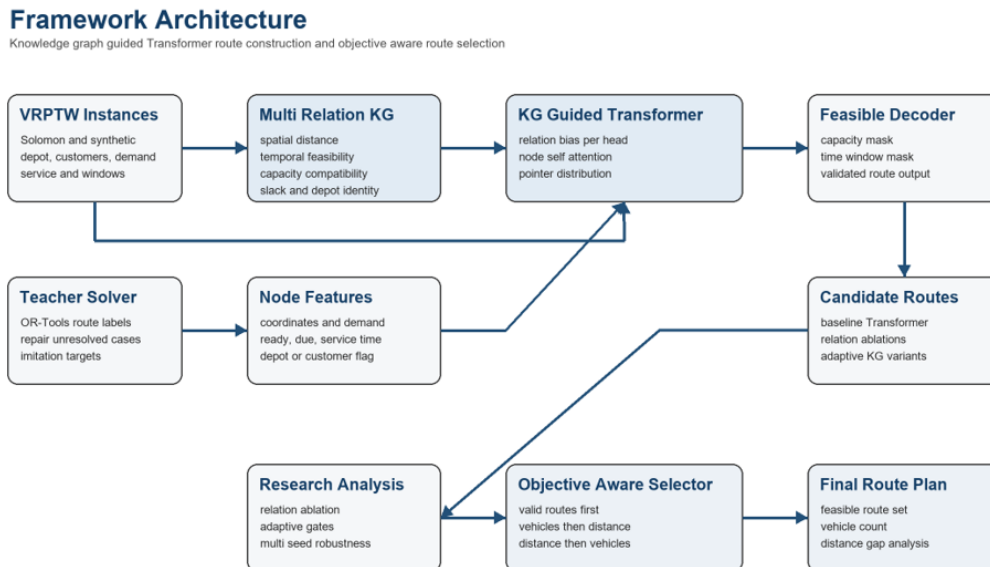


Fig. 1. Proposed multi-relation KG-guided Transformer framework with objective-aware route selection.

A. Problem Setting

Each instance contains one depot and n customers. Customer i has coordinates, demand, service duration, and a time window. A feasible route starts and ends at the depot, respects vehicle capacity, and begins service within every customer's time window. Evaluation reports the feasibility, the number of vehicles, total distance, and the gap relative to the OR-Tools reference.

B. Multi-Relation Knowledge Graph

For each ordered node pair, the KG stores relation features. The implemented groups are spatial relations, temporal feasibility, time window overlap, capacity compatibility, waiting time and slack, and depot or identity indicators. Ablation variants include spatial, temporal, capacity, feasible, slack, no slack, and full relation settings.

Let $V=\{0,\dots,n\}$ denote the depot and customer nodes, where node 0 is the depot. Customer i has coordinates x_i , demand q_i , service time s_i , ready time e_i , and due time l_i . Vehicle capacity is Q , the depot horizon is H , and d_{ij} is the Euclidean distance. For every ordered pair (i,j) , the KG relation vector r_{ij} contains normalized distance $d_{ij}/\max(d)$, a near-neighbor indicator $1[d_{ij} \leq q \cdot 0.20(d)]$, temporal feasibility $1[e_i+s_i+d_{ij} \leq l_j]$, reverse feasibility $1[e_j+s_j+d_{ij} \leq l_i]$, time-window overlap $1[\max(e_i,e_j) \leq \min(l_i,l_j)]$, and capacity compatibility $1[q_i+q_j \leq Q]$.

Slack and waiting relations are computed as $\text{wait}_{ij}=\max(0, e_j-(e_i+s_i+d_{ij}))/H$ and $\text{slack}_{ij}=\text{clip}((l_j-(e_i+s_i+d_{ij}))/H, -1, 1)$. Identity relations are $\text{same_node}=1[i=j]$, $\text{source_is_depot}=1[i=0]$, and $\text{target_is_depot}=1[j=0]$. These formulas make the KG deterministic and reproducible from a Solomon-style input file without external semantic resources.

The spatial group represents geometric closeness through normalized distance and near-neighbor indicators. These relations give the model a direct signal about route compactness. The temporal group represents whether one customer can precede another under service and travel time, including reverse feasibility and time window overlap. Slack and waiting relations capture how much scheduling flexibility remains when a customer pair is visited consecutively. The capacity group encodes whether two customer demands can coexist under the vehicle capacity, and the identity group distinguishes depot-related transitions from customer-to-customer transitions.

This formulation is intentionally simple and deterministic. The KG does not require external semantic knowledge or manually curated ontologies. Instead, it transforms the mathematical constraints of VRPTW into pairwise relation channels that can be consumed by the neural model. This makes the framework reproducible on any Solomon-style instance and allows each relation group to be removed or retained in ablation experiments.

C. KG-Guided Pointer Transformer

The relation-free baseline is a pointer-style Transformer that predicts the next customer from node embeddings. The KG-guided version projects relation vectors into head-specific

attention terms. These terms bias self attention before the pointer distribution is computed. During decoding, infeasible customers are masked to prevent capacity and time window violations.

Formally, node features X are projected to hidden states h_i . For attention head k , relation vector r_{ij} is mapped to a scalar bias $b_{ij}^k=\tanh(w_k^T r_{ij}+c_k)$ gamma, where gamma is a learned scale. The self-attention logit becomes $a_{ij}^k=(Q_i^k K_j^k)/\sqrt{d_h}+b_{ij}^k$ before the softmax over candidate keys j . In the implementation, the same relation-bias tensor is supplied to each relation-aware encoder layer, allowing different heads to learn different uses of spatial, temporal, capacity, slack, and depot relations.

The attention bias design is important because it lets each attention head learn a different use of the KG. One head may emphasize spatial closeness, another may emphasize temporal feasibility, and another may attend to depot-related transitions. The model, therefore, does not concatenate relation features to node features only once at the input layer; it uses relation information whenever pairwise attention scores are computed.

D. Training and Decoding

Training is formulated as imitation learning from teacher routes. For each partial route state, the teacher provides the next customer in the sequence, and the model is optimized to assign high probability to that teacher action. This design avoids the high variance of pure reinforcement learning and makes the experiment feasible on a modest GPU. It also creates a clear reference point: the neural model is learning to approximate the route construction behavior induced by the OR-Tools teacher.

During inference, the decoder constructs routes greedily under a feasibility mask. The mask removes customers who would violate capacity or time windows from the next action set. When no feasible customer remains, the vehicle returns to the depot, and a new route is started. After decoding, each route plan is validated again before it is used in evaluation or route selection. This validation step is essential because the study measures decoded operational feasibility, not only sequence prediction loss.

The feasibility mask is defined at each partial route state. A customer j can be selected only if j is unvisited, q_j is not larger than remaining capacity, and the service start time $\max(e_j, t_i+s_i+d_{ij})$ is not later than l_j . The depot action is allowed after at least one customer has been served on the current route, or when no customer satisfies the capacity and time-window tests. This hard mask prevents the decoder from selecting actions that are known to be infeasible before route validation.

E. Objective-Aware Route Selection

Different relation groups can be useful for different instance structures and objectives. Therefore, relation-specialized models generate route candidates. Valid candidates are ranked by either vehicles then distance or distance then vehicles. This route-level step converts relation diversity into an explicit selection mechanism.

Two ranking choices are considered because routing objectives are not always equivalent. A logistics operation that pays a fixed cost per vehicle may prefer vehicles then distance, while an operation with a fixed fleet may prefer distance then vehicles. Reporting both objectives prevents an overly broad claim of superiority and shows how the same candidate pool can be adapted to different operational priorities.

F. Implementation Details

The implementation settings used in the reported experiments are summarized in Table I. The local neural comparison uses the relation-free pointer Transformer as a direct baseline and keeps the same teacher-label generation and decoded feasibility validation as the KG-guided models.

TABLE I. IMPLEMENTATION AND TRAINING SETTINGS

Setting	Reported value
Relation-free model	d_model=128, 8 heads, 4 layers, FFN=256
KG-guided model	d_model=128, 4 heads, 3 relation-aware layers, FFN=256
Regularization	dropout=0.1, weight decay=1e-4, gradient clipping=1.0
Optimizer	AdamW; lr=1e-3 for relation-free baseline, 8e-4 for KG variants
Training	imitation learning from OR-Tools labels; batch size 32; Solomon 80 epochs; synthetic 50 epochs
Decoding	greedy pointer decoding with hard capacity and time-window mask
Controlled neural baselines	AM style attention pointer; POMO style 16-start selection; edge-aware Transformer with full relation features
Hardware	Remote GPU: NVIDIA RTX 3050, CUDA 12.4, PyTorch 2.6.0+cu124

G. Competitive Benchmarking Scope

Table II positions the proposed framework against established neural routing families and adds controlled neural baselines that were run under the same Solomon data, teacher label, decoding, and validation protocol. The AM style, POMO style multistart, and edge-aware Transformer rows are locally controlled baselines rather than official reproductions of AM, POMO, EFormer, GASE, or L2D codebases. This design directly strengthens the neural comparison while keeping the claim limited to verified same-pipeline results.

TABLE II. NEURAL BASELINE SCOPE AND CONTROLLED SOLOMON RESULTS.

Method/scope	Run type	Feas.	D-gap	V-gap
Official AM	literature only	--	--	--
Official POMO	literature only	--	--	--
L2D style solvers	literature only	--	--	--
Official EFormer/GASE	literature only	--	--	--
AM style attention pointer	controlled	55/56	288.059	3.836
Edge aware Transformer	controlled	55/56	275.804	3.764
POMO style multistart	controlled	55/56	223.276	2.636
Relation-free Transformer	controlled	54/56	318.654	4.056
Proposed KG full	controlled	56/56	304.202	3.464

V. EXPERIMENTAL DESIGN

The experiments use the 56 original Solomon 100 customer instances and synthetic Solomon-style extensions containing clustered, random, and mixed customer distributions. Teacher routes are generated with OR-Tools, including repair runs for initially unresolved cases. Models are trained by imitation and evaluated by decoded feasibility, distance gap, and vehicle gap.

The synthetic stage provides a robustness view beyond the small original benchmark. Three independent 120 instance synthetic runs are summarized with mean values and 95 percent confidence intervals. Because OR-Tools is the teacher and reference, the reported gaps are conservative measures of how far the learned route construction remains from the solver output.

The evaluation uses four complementary indicators. Feasibility measures whether the decoded solution satisfies all route constraints. Vehicle gap measures how many additional vehicles the learned route uses relative to OR-Tools. Distance gap is the extra travel distance relative to OR-Tools. Maximum gap captures the worst-case deterioration. Together, these metrics distinguish constraint satisfaction, fleet size, average route quality, and tail risk.

The original Solomon instances are used to evaluate compatibility with a standard benchmark. The synthetic extensions are not intended to replace the benchmark; they provide repeated instance samples so that route selection and relation ablation can be tested across multiple seeds. This distinction is important because a framework claim should not rely only on a single small benchmark split.

VI. RESULTS AND ANALYSIS

Table II adds controlled neural baselines requested for the competitive context. The POMO style multistart baseline obtains the lowest Solomon mean distance gap among the local neural baselines (223.276) and a lower mean vehicle gap than the single policy models, while the AM style and edge aware baselines also improve route quality over the original relation-free Transformer. However, all three additional controlled baselines leave one Solomon instance infeasible. The proposed KG full model is therefore not claimed to dominate every quality metric; its main advantage in the original Solomon benchmark is complete decoded feasibility with explicit relation diagnostics.

TABLE III. ORIGINAL SOLOMON INSTANCE RESULTS

Method	Feas.	D-gap	Max gap	V-gap
Transformer baseline	54/56	318.654	621.406	4.056
KG-guided Transformer	56/56	304.202	683.267	3.464
KG-guided Transformer + repair	56/56	295.435	671.015	3.464

Table III reports the original Solomon results. KG guidance improves feasibility over the relation-free Transformer: the baseline decoded feasible routes for 54 of 56 instances, whereas the full KG-guided model decoded feasible routes for all 56 instances and reduced the mean vehicle gap. The repair variant further reduced the mean distance gap while retaining complete feasibility.

Table IV shows the Solomon relation ablation. Relation groups contribute differently: temporal and slack-oriented variants are competitive on the distance gap, while the full setting gives complete feasibility. The no slack setting has a low distance gap but loses one feasible instance, showing that relation selection changes the tradeoff between feasibility, distance, and vehicle count.

TABLE IV. SOLOMON RELATION ABLATION SUMMARY

Relation	Feas.	D-gap	V-gap	Relation specialization
spatial	55/56	295.447	3.455	spatial
temporal	56/56	272.667	4.268	temporal
capacity	55/56	276.861	3.964	capacity
feasible	55/56	295.612	3.909	feasible
slack	56/56	272.708	3.643	slack
no_slack	55/56	267.915	3.309	no_slack
full	56/56	304.202	3.464	full

Table V reports the synthetic medium route selection results, and Table VI summarizes the multi-seed synthetic robustness study. Objective-aware ensembling is stronger than any single learned model in terms of feasibility and distance gap. Across three 120 instance synthetic runs, the distance-oriented ensemble reaches a mean feasible rate of 0.997 and reduces the mean distance gap by 125.418 Solomon coordinate distance units relative to the relation-free Transformer. Table VII further shows that the hardest groups are R1 and RC1, where relation diversity and selection provide the largest feasibility improvements.

TABLE V. SYNTHETIC MEDIUM ROUTE SELECTION RESULTS

Method	Feas.	D-gap	V-gap
transformer_baseline	55/60	451.255	3.836
kg_ablation_full	55/60	404.900	3.218
kg_ablation_no_slack	59/60	445.001	3.153
adaptive_kg_transformer_l1	56/60	432.546	3.679
ensemble_all	60/60	344.374	1.867
ensemble_all_distance	60/60	311.390	2.633
router_ranker	57/60	437.149	3.263
router_nn	55/60	439.378	3.491

TABLE VI. MULTI-SEED SYNTHETIC ROBUSTNESS

Method	Feas. rate	D-gap	V-gap
transformer_baseline	0.858 +/- 0.062	427.827 +/- 22.013	4.557 +/- 0.960
kg_ablation_full	0.911 +/- 0.097	407.123 +/- 19.786	3.954 +/- 0.991
kg_ablation_no_slack	0.936 +/- 0.029	390.777 +/- 26.168	3.070 +/- 0.584
adaptive_kg_transformer_l1	0.914 +/- 0.082	393.310 +/- 13.080	3.769 +/- 1.181
router_nn	0.933 +/- 0.025	410.077 +/- 14.317	3.304 +/- 0.121
router_ranker	0.947 +/- 0.020	421.206 +/- 5.469	3.180 +/- 0.092

ensemble_all	0.997 +/- 0.005	336.183 +/- 9.882	2.031 +/- 0.099
ensemble_all_distanc	0.997 +/- 0.005	302.409 +/- 4.245	2.858 +/- 0.037

Table VIII reports paired t-tests across the three independent 120 instance synthetic runs. The distance-oriented ensemble significantly reduces the distance gap relative to the relation-free Transformer baseline ($p=0.0059$) and significantly increases the feasible rate ($p=0.0472$). The vehicle-oriented ensemble also significantly reduces the vehicle gap ($p=0.0327$). The full KG ablation improves the feasible rate, but the three-seed test does not show significance for that isolated comparison, which supports the interpretation that objective-aware route selection is the most robust component in the current experiments.

TABLE VII. HARD CLASS FEASIBILITY AND DISTANCE GAP

Group	Base feas.	Ens. feas.	Base gap	Ens. gap
C1	0.950	1.000	402.579	270.620
C2	1.000	1.000	343.069	211.234
R1	0.417	0.983	476.645	369.153
R2	1.000	1.000	464.693	323.813
RC1	0.783	1.000	485.504	334.981
RC2	1.000	1.000	430.212	305.588

TABLE VIII. PAIRED MULTI-SEED SIGNIFICANCE TESTS

Comparison	Metric	Delta	t	p
ensemble_all_distance vs baseline	distance gap reduction	125.418	12.979	0.0059
ensemble_all_distance vs baseline	feasible rate increase	0.139	4.437	0.0472
ensemble_all vs baseline	vehicle gap reduction	2.526	5.390	0.0327
kg_ablation_full vs baseline	feasible rate increase	0.053	1.142	0.3719

TABLE IX. INFERENCE TIME BENCHMARK ON SOLOMON INSTANCES

Model	Device	Repeats	Mean s/inst.	Min-Max s	Overhead
Relation-free Transformer	cuda	3	1.356	1.003-1.879	reference
Full KG-guided Transformer	cuda	3	1.668	1.259-2.152	+23.0%

The results also show that full relation information is not always the best distance-oriented configuration. On the original Solomon ablation, the no slack and slack variants produce lower mean distance gaps than the full relation model, while the full relation model gives complete feasibility. This suggests that relation richness can improve constraint awareness but may also change the model's route construction bias. Therefore, the ablation should be interpreted as evidence for controllability rather than as a simple ranking of relation sets.

The hard class analysis is especially informative. R1 and RC1 are more difficult because customer locations are less clustered and time windows are tighter. The relation free Transformer has a low feasible rate in R1, whereas the distance-oriented ensemble reaches near complete feasibility.

This indicates that route validation and candidate diversity are most valuable when a single learned policy is brittle.

VII. DISCUSSION

The main finding is that explicit relation structure makes neural route construction more controllable. The KG is useful not only as an input feature source but also as an experimental mechanism for testing which routing knowledge supports feasibility and route quality. Objective-aware selection further shows that relation-specialized models can be combined without assuming that a single relation setting is best for every instance class.

The framework should be interpreted as a learned construction and diagnostic layer. It does not replace OR-Tools, which remains the reference solver and teacher. Instead, it provides a structured neural route generation pipeline that can be paired with local search, repair, or solver-based refinement in future work.

From a methodological perspective, the proposed KG has two roles. The first role is inductive bias: it exposes pairwise feasibility and routing structure to the Transformer before the model has to infer those relations from data. The second role is experimental transparency: relation groups can be added, removed, or selected, making it possible to explain performance differences in terms of routing knowledge rather than only model size or training loss.

From a practical perspective, the framework is most useful as a fast route construction component. A dispatcher or decision support system could use KG-guided decoding to generate feasible candidate routes quickly, then pass the best candidates to a local search, repair heuristic, or exact solver for improvement. In this setting, the neural model contributes speed and diversity, while the downstream optimizer contributes final route refinement decision support system could use KG-guided decoding to generate feasible candidate routes quickly, then pass the best candidates to a local search, repair heuristic, or exact solver for improvement. In this setting, the neural model contributes speed and diversity, while the downstream optimizer contributes final route refinement.

Table IX reports inference time on the RTX 3050 environment using the 56 Solomon instances and three repeats per instance. The relation-free Transformer averages 1.356 s per instance, while the full KG-guided Transformer averages 1.668 s per instance. Thus, explicit KG relation bias adds about 23.0 percent decoding overhead in this implementation, which is modest relative to the added feasibility and diagnostic benefits reported above.

VIII. LIMITATIONS AND FUTURE WORK

The current model uses greedy decoding, and the learned routes remain worse than OR-Tools in terms of distance. The original Solomon set is also small, so synthetic extensions are useful for robustness but cannot replace additional external benchmarks. The added AM style, POMO style, and edge-aware Transformer baselines strengthen the same-pipeline comparison, but official reproduction of AM, POMO, L2D, EFormer, GASE, and other stronger neural or hybrid VRP solvers remains future work.

Another limitation is that the relation schema is manually designed from known VRPTW constraints. This is appropriate for a first framework study, but future research could learn relation weights or discover additional relation channels automatically. The router component also needs more data before it can consistently outperform simple objective aware ensembling. Finally, computational time should be reported more fully in future comparisons so that route quality can be interpreted together with inference cost.

IX. CONCLUSION

This study presented a multi-relation KG-guided Transformer framework for VRPTW. The KG is projected into Transformer attention as pairwise relation bias, and the framework is evaluated through ablation, adaptive variants, route selection, and multi-seed synthetic robustness. The results show improved feasibility over a relation-free Transformer and stronger robustness through objective-aware route selection. The contribution is a framework for operationalizing explicit routing knowledge inside neural attention and for analyzing how relation groups affect learned VRPTW construction.

REFERENCES

- [1] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254-265, 1987, doi: <https://doi.org/10.1287/opre.35.2.254>.
- [2] Google, "OR-Tools routing: vehicle routing," Google for Developers, Jan. 22, 2025. [Online]. Available: <https://developers.google.com/optimization/routing>. Accessed: May 17, 2026.
- [3] Aigerim Bogrybayeva; Meraryslan Meraliyev; Taukekhan Mustakhov; Bissenbay Dauletbayev, "Learning to Solve Vehicle Routing Problems: A Survey," arXiv (Cornell University), 2022. doi: <https://doi.org/10.48550/arxiv.2205.02453>.
- [4] Xuan Wu; Di Wang; Lijie Wen; Yubin Xiao; Chunguo Wu; Yuesong Wu; Chaoyu Yu; Douglas L. Maskell; You Zhou, "Neural Combinatorial Optimization Algorithms for Solving Vehicle Routing Problems: A Comprehensive Survey with Perspectives," arXiv (Cornell University), 2024. doi: <https://doi.org/10.48550/arxiv.2406.00415>.
- [5] K. T. Chung; C.K.M. Lee; Yung Po Tsang, "Neural combinatorial optimization with reinforcement learning in industrial engineering: a survey," *Artificial Intelligence Review*, 2025. doi: <https://doi.org/10.1007/s10462-024-11045-1>.
- [6] Cheng Chi; Amine Mohamed Aboussalah; Elias B. Khalil; Juyoung Wang; Zoha Sherkat-Masoumi, "A Deep Reinforcement Learning Framework For Column Generation," arXiv (Cornell University), 2022. doi: <https://doi.org/10.48550/arxiv.2206.02568>.
- [7] Jonas K. Falkner; Daniela Thyssens; Lars Schmidt-Thieme, "Large Neighborhood Search based on Neural Construction Heuristics," arXiv (Cornell University), 2022. doi: <https://doi.org/10.48550/arxiv.2205.00772>.
- [8] Leo Baty; Kai Jungel; Patrick S. Klein; Axel Parmentier; Maximilian Schiffer, "Combinatorial Optimization enriched Machine Learning to solve the Dynamic Vehicle Routing Problem with Time Windows," arXiv (Cornell University), 2023. doi: <https://doi.org/10.48550/arxiv.2304.00789>.
- [9] Jianjun Deng; Junjie Wang; Xiaojun Wang; Yiqiao Cai; Peizhong Liu, "Multi-Task Multi-Objective Evolutionary Search Based on Deep Reinforcement Learning for Multi-Objective Vehicle Routing Problems with Time Windows," *Symmetry*, 2024. doi: <https://doi.org/10.3390/sym16081030>.
- [10] Rixin Wu; Ran Wang; Jie Hao; Qiang Wu; Ping Wang; Dusit Niyato, "Multiobjective Vehicle Routing Optimization With Time Windows: A Hybrid Approach Using Deep Reinforcement Learning and NSGA-II,"

- IEEE Transactions on Intelligent Transportation Systems, 2024. doi: <https://doi.org/10.1109/tits.2024.3515997>.
- [11] Fahong Yu; Meijia Chen; Xiaoyun Xia; Dongping Zhu; Qiang Peng; Kuibiao Deng, "Logistics Distribution Route Optimization With Time Windows Based on Multi-Agent Deep Reinforcement Learning," *International Journal of Information Technologies and Systems Approach*, 2023. doi: <https://doi.org/10.4018/ijitsa.342084>.
- [12] Zixian Zhang; Geqi Qi; Wei Guan, "Coordinated multi-agent hierarchical deep reinforcement learning to solve multi-trip vehicle routing problems with soft time windows," *IET Intelligent Transport Systems*, 2023. doi: <https://doi.org/10.1049/itr2.12394>.
- [13] Arash Mozhdghi; Mahdi Mohammadzadeh; Xin Wang, "Edge-DIRECT: A Deep Reinforcement Learning-based Method for Solving Heterogeneous Electric Vehicle Routing Problem with Time Window Constraints," *arXiv (Cornell University)*, 2024. doi: <https://doi.org/10.48550/arxiv.2407.01615>.
- [14] Dimeth Nouicer; Ikkal Chammakhi Msadaa; Khaled Grayaa, "A Novel Routing Solution for EV Fleets: A Real-World Case Study Leveraging Double DQNs and Graph-Structured Data to Solve the EVRPTW Problem," *IEEE Access*, 2023. doi: <https://doi.org/10.1109/access.2023.3327324>.
- [15] Zefang Zong; Xia Tong; Meng Zheng; Yong Li, "Reinforcement Learning for Solving Multiple Vehicle Routing Problem with Time Window," *ACM Transactions on Intelligent Systems and Technology*, 2024. doi: <https://doi.org/10.1145/3625232>.
- [16] Zengjian Yang, "Learning to Solve Vehicle Routing Problem with Soft Time Window via Collaborative Transformer," *Online*, 2024. doi: <https://doi.org/10.65286/icic.v20i2.44243>.
- [17] Dian Meng; Zhiguang Cao; Yaixin Wu; Yaqing Hou; Hongwei Ge; Qiang Zhang, "EFormer: An Effective Edge-based Transformer for Vehicle Routing Problems," *Online*, 2024. doi: <https://doi.org/10.24963/ijcai.2024/954>.
- [18] Wei Li; Bing Tian Dai; Xueming Yan; Junying Zou; Zhijie Liang; Jingwen Li, "Learning to solving vehicle routing problems via local-global feature fusion transformer," *Complex & Intelligent Systems*, 2025. doi: <https://doi.org/10.1007/s40747-025-02018-0>.
- [19] Zhenwei Wang; Ruibin Bai; Fazlullah Khan; Ender Ozcan; Tiehua Zhang, "Gase: graph attention sampling with edges fusion for solving vehicle routing problems," *Memetic Computing*, 2024. doi: <https://doi.org/10.1007/s12293-024-00428-0>.
- [20] Cong Dao Tran; Thong Bach; Truong Son Hy, "Symmetry-preserving graph attention network to solve routing problems at multiple resolutions," *arXiv (Cornell University)*, 2023. doi: <https://doi.org/10.48550/arxiv.2310.15543>.
- [21] Jiwoo Son; Minsu Kim; Sanghyeok Choi; Hyeonah Kim; Jinkyoo Park, "Equity-Transformer: Solving NP-Hard Min-Max Routing Problems as Sequential Generation with Equity Context," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: <https://doi.org/10.1609/aaai.v38i18.30007>.
- [22] Zhuoyi Lin; Yaixin Wu; Bangjian Zhou; Zhiguang Cao; Wen Song; Yingqian Zhang; J. Senthilnath, "Cross-Problem Learning for Solving Vehicle Routing Problems," *arXiv (Cornell University)*, 2024. doi: <https://doi.org/10.48550/arxiv.2404.11677>.
- [23] Samuel J. K. Chin; Matthias Winkenbach; Akash Srivastava, "Learning to Deliver: a Foundation Model for the Montreal Capacitated Vehicle Routing Problem," *arXiv (Cornell University)*, 2024. doi: <https://doi.org/10.48550/arxiv.2403.00026>.
- [24] Jianan Zhou; Zhiguang Cao; Yaixin Wu; Wen Song; Yining Ma; Jie Zhang; Chi Xu, "MVMoE: Multi-Task Vehicle Routing Solver with Mixture-of-Experts," *arXiv (Cornell University)*, 2024. doi: <https://doi.org/10.48550/arxiv.2405.01029>.
- [25] Federico Berto; Chuanbo Hua; Nayeli Gast Zepeda; Andre Hottung; Niels A. Wouda; Leon Lan; Park, Junyoung; Kevin Tierney; Jinkyoo Park, "RouteFinder: Towards Foundation Models for Vehicle Routing Problems," *arXiv (Cornell University)*, 2024. doi: <https://doi.org/10.48550/arxiv.2406.15007>.
- [26] Zefang Zong; Hansen Wang; Jingwei Wang; Meng Zheng; Yong Li, "RBG: Hierarchically Solving Large-Scale Routing Problems in Logistic Systems via Reinforcement Learning," *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022. doi: <https://doi.org/10.1145/3534678.3539037>.
- [27] Zhao Yun; Xiaoning Shen; Zhongpei Ge, "A Knowledge-Guided Multi-Objective Shuffled Frog Leaping Algorithm for Dynamic Multi-Depot Multi-Trip Vehicle Routing Problem," *Symmetry*, 2024. doi: <https://doi.org/10.3390/sym16060697>.
- [28] Reza Shahbazian; Alessia Ciacco; Giusy Macrina; Francesca Guerriero, "Knowledge-guided hybrid deep reinforcement learning for the dynamic multi-depot electric vehicle routing problem," *Computers & Operations Research*, 2025. doi: <https://doi.org/10.1016/j.cor.2025.107217>.
- [29] Jorin Dornemann, "Solving the capacitated vehicle routing problem with time windows via graph convolutional network assisted tree search and quantum-inspired computing," *Frontiers in Applied Mathematics and Statistics*, 2023. doi: <https://doi.org/10.3389/fams.2023.1155356>.
- [30] Kassem Danach; Louai Saker; Hassan Harb, "Integrating Metaheuristics and Machine Learning for Enhanced Vehicle Routing: A Comparative Study of Hyperheuristic and VAE-Based Approaches," *World Electric Vehicle Journal*, 2025. doi: <https://doi.org/10.3390/wevj16050258>.
- [31] Ying Zhou; Lingjing Kong; Hui Wang, "A multiobjective edge-based learning algorithm for the vehicle routing problem with time windows," *Information Sciences*, 2025. doi: <https://doi.org/10.1016/j.ins.2025.122223>.
- [32] Conghui Wang; Zhiguang Cao; Yaixin Wu; Long Teng; Guohua Wu, "Deep Reinforcement Learning for Solving Vehicle Routing Problems With Backhauls," *IEEE Transactions on Neural Networks and Learning Systems*, 2024. doi: <https://doi.org/10.1109/tnnls.2024.3371781>.
- [33] S. Kosolsombat and C. Ratanavilisagul, "Modified ant colony optimization with selecting and elimination customer and re-initialization for VRPTW," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 6, pp. 3471-3482, Dec. 2022, doi: <https://doi.org/10.11591/eei.v11i6.3943>.
- [34] S. Kosolsombat and C. Ratanavilisagul, "Applied Deep Reinforcement Learning for Solving the Vehicle Routing Problem with Time Windows," in *2023 8th International Conference on Computational Intelligence and Applications (ICCIA)*, Haikou, China, Jun. 23-25, 2023, pp. 21-25, doi: <https://doi.org/10.1109/ICCIA59741.2023.00012>.