CARD: COUPLED ASYMMETRIC ROLE-AND-DISTANCE IN REAL-WORLD VEHICLE ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning has achieved strong performance in vehicle routing problems (VRPs), yet existing methods mostly ignore both distance and role asymmetries in real-world VRPs, with a few addressing only one. We identify that these two asymmetries are inherently coupled: existing solvers that handle only one degrade more under combined shifts than under each shift separately. We propose CARD, a **C**oupled **A**symmetric **R**ole-and-**D**istance framework that eliminates this coupling by jointly handling both asymmetries. For asymmetric distance, CARD uses an attention mechanism that explicitly takes both the original and transposed cost matrices as input, while implicitly encoding role distinctions via its MLP scorer. For asymmetric roles, CARD introduces a role-aware training paradigm that operates on the role-reversed view alongside the original view, regularizing the policy through shared gradients. CARD consistently outperforms state-of-the-art neural solver RADAR across four VRP variants, with larger gains under more severe asymmetry, and achieves 5× training speedup to the same performance level. Crucially, CARD remains robust on real-world routing benchmarks across a wide range of distribution shifts without retraining.

1 INTRODUCTION

Real-world vehicle routing problems (VRPs) often involve asymmetric travel costs and heterogeneous service roles Christofides et al. (1979). These practical challenges have been formulated as several well-studied VRP variants. The Asymmetric VRP (AVRP) captures distance asymmetry, while the Pickup and Delivery Problem (PDP) and the Vehicle Routing Problem with Backhauls (VRPB) are concerned with role asymmetry Savelsbergh & Sol (1995); Deif & Bodin (1984). Such problems are NP-hard and require efficient solutions to reduce operational costs Parragh et al. (2008). We observe that these two asymmetries are not independent but coupled: directional travel costs and service roles mutually amplify each other's difficulty, making their joint treatment essential for realistic routing Li et al. (2021). Traditional operations research (OR) solvers can produce high-quality solutions but lack real-time adaptability and large-instance scalability Li & Lim (2003). Recent neural solvers offer greater flexibility and faster inference, yet they often suffer from slow training and poor solution quality Bello et al. (2016). They typically require tens of millions of training instances to converge, yet often settle into suboptimal local optima Kool et al. (2019). This issue is especially pronounced when both asymmetries are present Wang et al. (2025). Recent neural solvers have begun to address real-world asymmetries, but existing efforts tackle only one of them: RADAR targets distance asymmetry via SVD-based initialization and Sinkhorn normalization Yi et al. (2026), while PD-SNO addresses role asymmetry through symmetric node partitioning Li et al. (2024b). Consequently, while both approaches have advanced their respective directions, they each address only one type of asymmetry. They either ignore role structures or overlook directional costs, often producing inadequate solutions when both asymmetries coexist Liu et al. (2023); Cai et al. (2023). In problems where both are present, such as asymmetric PDP and VRPB variants, addressing only one type inevitably fails regardless of how well the individual case is handled Ma et al. (2022); Kong et al. (2024). To the best of our knowledge, no method handles both simultaneously, nor has any recognized or exploited their coupling, motivating us to treat them jointly.

In this paper, we propose CARD, a **C**oupled **A**symmetric **R**ole-and-**D**istance framework that jointly addresses both distance and role asymmetries. For distance asymmetry, we propose a distance-aware attention mechanism that leverages the fact that the transpose conveys complementary directional

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

information. This mechanism explicitly incorporates the transposed cost matrix C^T alongside the original cost matrix C and dot-product scores as input to attention, while implicitly capturing role sensitivity through the MLP fusion. Together, these two aspects couple distance and role processing within a single attention layer. For role asymmetry, we propose a role-aware training paradigm that exploits the observation that swapping the role assignment of all nodes produces an equivalent instance, i.e., $(d_i, p_i) \mapsto (p_i, d_i)$. This paradigm explicitly optimizes on both the original and role-reversed views, while implicitly stabilizing gradient updates through shared gradients. Role swapping (e.g., pickup $\leftrightarrow$ delivery or linehaul $\leftrightarrow$ backhaul) yields a structurally equivalent instance, without implying a learnable invertible mapping. Together, these two aspects couple role-specific optimization with cross-view gradient regularization during training. Together, these two components form a complementary framework that couples distance and role processing end-to-end. This joint design yields five empirical advantages: (1) it is substantially faster to train, reaching competitive performance in 100 epochs where RADAR needs 1000; (2) it yields consistently better solutions, outperforming prior methods across all tested VRP variants with both asymmetries; (3) it generalizes reliably across scales, handling instances 10× larger (e.g., from 100 to 1000 nodes) without retraining; (4) it adapts robustly to shifts, maintaining performance under diverse cost and demand distributions; (5) it transfers effectively to real-world networks, delivering reliable solutions under realistic travel costs.

Our main contributions are summarized as follows:

- We identify that distance and role asymmetries are inherently coupled in practical routing, and provide empirical evidence of a super-additive degradation under combined shifts: handling either asymmetry alone fails, while joint treatment eliminates this degradation.
- We propose CARD, a coupled asymmetric role-and-distance framework that eliminates this coupling through two complementary components: a distance-aware attention mechanism for directional costs and a role-aware training paradigm for cross-view consistency.
- Extensive experiments on four representative VRP variants demonstrate that CARD consistently outperforms state-of-the-art neural solvers across synthetic and real-world benchmarks, while achieving faster wall-clock training convergence.

2 Related Work

Despite growing interest in AVRP, PDP, and VRPB variants, existing OR and DRL efforts have largely progressed along separate tracks, focusing on different facets of asymmetry.

Classical Operations Research Methods. Early works formalized these problems with subsequent metaheuristics, though these methods are generally limited to small instances and require extensive tuning for new constraints Savelsbergh & Sol (1995); Parragh et al. (2008); Deif & Bodin (1984); Christofides et al. (1979); Li & Lim (2003).

Neural Solvers for Canonical Routing. The Attention Model first introduced Transformer-based architectures with REINFORCE training for TSP and CVRP Kool et al. (2019); Williams (1992). POMO significantly improved AM through multi-start decoding and a shared baseline Kwon et al. (2020). These methods have advanced neural combinatorial optimization for canonical symmetric routing, but they address neither distance nor role asymmetry Luo et al. (2023); Huang et al. (2025); Cao et al. (2026); Meng et al. (2026).

Neural Solvers for Asymmetric Routing. MatNet introduced a matrix encoding architecture to process matrix-form input, enabling neural solvers to handle asymmetric routing problems such as ATSP Kwon et al. (2021). RRNCO provides real-world routing benchmarks with asymmetric distance, bridging the gap between synthetic and practical settings Son et al. (2026). RADAR introduces SVD-based initialization and Sinkhorn normalization to encode directional cost information, achieving strong performance on asymmetric VRPs Yi et al. (2026). Despite these advances, these methods focus exclusively on distance asymmetry and do not consider the role asymmetry inherent in pickup-and-delivery and backhaul routing problems.

Neural Solvers for Structural Symmetries. Sym-NCO explicitly incorporates geometric symmetries such as rotation, reflection, and permutation as regularization terms to improve generalization Kim et al. (2022). MD-MTA exploits depot-level symmetry in multi-depot VRP Li et al. (2024a).

PD-SNO applies role-level symmetry in PDP by swapping pickup and delivery nodes during inference Li et al. (2024b). However, these methods focus exclusively on structural or role symmetries and do not consider directional cost asymmetries. Existing role-aware methods rely on coordinate-based symmetries or cost-preserving route reversal, both of which break down under asymmetric distance matrices, making direct comparison infeasible. Meanwhile, the distance-aware methods discussed above address only asymmetric travel costs without modeling role structures. Neither line of work handles both simultaneously, motivating the need for a framework that jointly addresses distance and role asymmetries.

3 Problem Formulation and Preliminaries

We consider four representative asymmetric VRP variants: the Asymmetric Multi-commodity Pickup and Delivery TSP (AMPDTSP), the Asymmetric Pickup and Delivery TSP (APDTSP), the Asymmetric VRP with Mixed Backhauls (AVRPMB), and the Asymmetric VRP with Simultaneous Delivery and Pickup (AVRPSDP). Let $V = \{0, 1, \ldots, n\}$ denote the set of nodes, where 0 is the depot, and let $C \in \mathbb{R}^{(n+1)\times(n+1)}$ denote the asymmetric cost matrix with $C_{ij} \neq C_{ji}$. Each node is assigned at least one role (pickup, delivery, linehaul, or backhaul); in some variants, a node may serve both pickup and delivery roles. Demands are denoted as $d_i \geq 0$ for delivery and $p_i \geq 0$ for pickup, and vehicles have capacity Q when applicable. Let $x_{ij} \in \{0, 1\}$ indicate whether a vehicle travels directly from node i to node j. The objective is to minimize the total travel cost:

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} C_{ij} x_{ij} \tag{1}$$

AMPDTSP has paired pickup-delivery requests with precedence, commodity weights, and capacity; **APDTSP** is the special case without weights or capacity. **AVRPMB** classifies each customer as linehaul or backhaul with no ordering restrictions. **AVRPSDP** requires each customer to be served with both delivery and pickup in a single visit, with load kept within $[0, Q]$. Formal definitions of all four variants are provided in Appendix A.1 (Table 6).

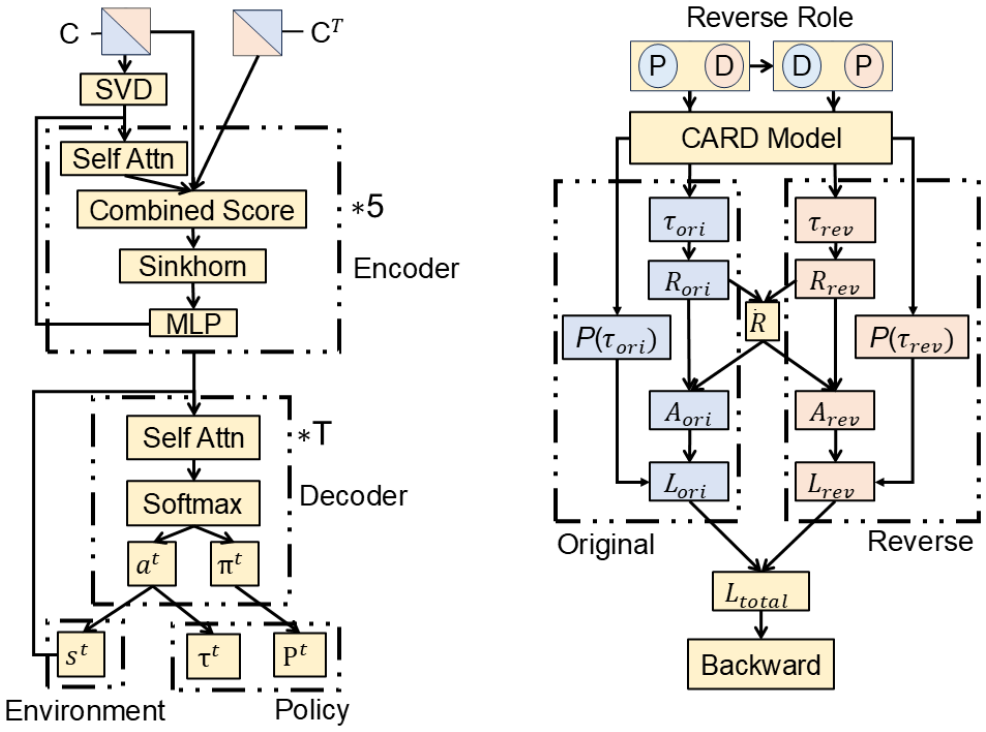
MDP Formulation. We formulate each variant as a Markov Decision Process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ Bellman (1957). A state $s^t \in \mathcal{S}$ captures the vehicle's current position and load, the partial visitation sequence, the remaining unvisited nodes, and the remaining demand. An action $a^t \in \mathcal{A}$ selects the next feasible node. The transition $\mathcal{P}(s^{t+1} \mid s^t, a^t)$ updates the vehicle's position, load, visited set, and remaining demand according to the selected node. The reward is $r^t = -C_{i,j}$, and the total reward of a trajectory is $R = \sum_0^{t-1} r^t$. This unified formulation covers all four variants and enables end-to-end policy learning without hand-crafted heuristics.

SVD-based Initialization. Following RADAR Yi et al. (2026), we preprocess C via low-rank SVD $C \approx U\Sigma V^\top$ with rank $k = 10$. The left and right singular vectors are projected to form initial node embeddings, capturing both out-bound and in-bound structural information.

Sinkhorn Normalization. Following RADAR Yi et al. (2026), we replace softmax with Sinkhorn normalization Sander et al. (2022) using $T = 10$ iterations, enforcing both row and column sums of the attention matrix to be 1. This doubly stochastic normalization allows the model to capture global context from all nodes jointly.

4 Methodology

The overall architecture and training pipeline of CARD is illustrated in Figure 1. Here "ori" and "rev" denote the original and role-reversed views, respectively. The core challenge is that distance and role asymmetries are coupled: handling either alone leads to super-additive degradation under combined shifts. To eliminate this coupling, CARD builds on the SVD-based initialization and Sinkhorn normalization introduced in Section 3 and consists of two complementary components. We first present a distance-aware attention mechanism that incorporates directional cost information into the encoder. We then introduce dual-view rollout and a role-aware loss that enforces policy consistency across the original and role-reversed views. During inference, the same dual-view decoding is applied and the better of the two solutions is selected.



(a) CARD Model (b) CARD Training

Figure 1: CARD framework. Left: distance-aware model with explicit bi-directional cost inputs. Right: role-aware training with explicit dual-view rollout.

4.1 Distance-Aware Attention with Role-Sensitive Scoring

Our attention mechanism primarily captures asymmetric distance patterns by fusing directional costs with query-key similarities via an MLP, while also encoding role sensitivity implicitly. For each pair of nodes i, j, the score is:

$$\omega_{ij} = \text{MLP}([Q_i K_j^\top / \sqrt{d_k},\ C_{ij},\ C_{ji}]) \tag{2}$$

where Q_i, K_j are query-key projections, and C_{ij}, C_{ji} are the forward and reverse costs. This score is computed by a two-layer MLP with ReLU activation and normalized by Sinkhorn iterations to enforce doubly stochasticity. This MLP-based formulation differs from standard additive attention, which would compute a linear combination $Q_i K_j^\top + \alpha C_{ij} + \beta C_{ji}$ with fixed scalars α, β. Such a linear form fails to capture interactions between directional costs and node roles, as the weight of C_{ij} is globally constant across all node pairs regardless of their role compositions.

By contrast, the MLP in our scorer can approximate any continuous function of $[Q_i K_j^\top, C_{ij}, C_{ji}]$, enabling input-dependent weighting of directional costs. Thus we include both C_{ij} and C_{ji}, allowing the MLP to emphasize forward cost when roles are complementary and suppress it otherwise. To illustrate this MLP-based scorer, consider a simplified decomposition:

$$\omega_{ij} \approx \underbrace{\phi(Q_i K_j^\top, C_{ij} + C_{ji})}_{\text{Symmetric base}} + \underbrace{\psi(Q_i K_j^\top) \cdot (C_{ij} - C_{ji})}_{\text{Direction bias}} + \underbrace{\xi(Q_i K_j^\top, C_{ij}, C_{ji}) \cdot \mathbb{I}[\text{role}(i) \neq \text{role}(j)]}_{\text{Role-modulated direction term}}, \tag{3}$$

where $\mathbb{I}[\text{role}(i) \neq \text{role}(j)]$ indicates complementary roles. This conceptual decomposition shows what the scorer can learn. In contrast, a linear formulation without C_{ji} reduces to $Q_i K_j^\top + \alpha C_{ij}$, which lacks such flexibility. We empirically validate the contribution of including both directional costs in Table 5. Appendix A.2 (Fig. 4) visualizes the role-modulated behavior: complementary pairs exhibit 20–30$\times$ higher sensitivity to directional costs than same-role pairs, confirming that the scorer differentiates role compositions.

4.2 DUAL-VIEW ROLLOUT

We perform dual-view rollout on original and role-reversed views. Both views evaluate on the same cost matrix C; details on how the role-reversed solution maps back to the original problem are provided in Appendix A.3. For each view, start nodes are restricted to role-valid ones: pickup/linehaul in the original view, delivery/backhaul in the role-reversed view, except AVRPSDP where both views allow all nodes. Let N_{ori} and N_{rev} denote the corresponding numbers of valid starts, which may differ across views and reflect the role-specific structure of each view.

Algorithm 1 Dual-View Rollout in CARD Training

Require: Environment E (with terminal state s^f), policy π_θ

for each epoch **do**
  $s^0_{\text{ori}} \leftarrow E.\text{reset}(); \quad P_{\text{ori}} \leftarrow 0; \quad \tau_{\text{ori}} \leftarrow \emptyset$
  while $s^t_{\text{ori}} \neq s^f$ **do**
    $a^t_{\text{ori}}, \pi^t_{\text{ori}} \leftarrow \pi_\theta(s^t_{\text{ori}}); \quad s^{t+1}_{\text{ori}} \leftarrow E(a^t_{\text{ori}}); \quad \tau_{\text{ori}} \leftarrow \tau_{\text{ori}} \cup a^t_{\text{ori}}; \quad P_{\text{ori}} \leftarrow P_{\text{ori}} + \log \pi^t_{\text{ori}}$
  end while
  $s^0_{\text{rev}} \leftarrow E.\text{reset}(); \quad P_{\text{rev}} \leftarrow 0; \quad \tau_{\text{rev}} \leftarrow \emptyset$
  while $s^t_{\text{rev}} \neq s^f$ **do**
    $a^t_{\text{rev}}, \pi^t_{\text{rev}} \leftarrow \pi_\theta(s^t_{\text{rev}}); \quad s^{t+1}_{\text{rev}} \leftarrow E(a^t_{\text{rev}}); \quad \tau_{\text{rev}} \leftarrow \tau_{\text{rev}} \cup a^t_{\text{rev}}; \quad P_{\text{rev}} \leftarrow P_{\text{rev}} + \log \pi^t_{\text{rev}}$
  end while
  Update θ using L_{total} from Eq. (8)
end for

During training, Algorithm 1 generates one trajectory per valid start node in each view. For each trajectory, we record its total cost and the log-likelihood of the selected actions. Let T be the number of decision steps. For the original view, each trajectory τ_{ori} yields a total reward $R_{\text{ori}} = -\text{cost}(\tau_{\text{ori}})$, where a higher reward indicates a lower travel cost. Its log-likelihood is:

$$P(\tau_{\text{ori}}) = \sum_{t=0}^{T-1} \log \pi(a^t_{\text{ori}} \mid s^t_{\text{ori}}). \tag{4}$$

The role-reversed view produces R_{rev} and $P(\tau_{\text{rev}})$ analogously. These quantities are then used to compute the advantages and the total loss L_{total} as defined in Sec. 4.3.

During inference, we apply the same dual-view rollout procedure, but replace stochastic sampling with greedy decoding ($a^t = \arg\max \pi(\cdot \mid s^t)$) and omit the log-likelihood accumulation. For each view, we generate one greedy trajectory per valid start, and select the better of the two decoded solutions based on the total cost. No parameter update is performed.

4.3 ROLE-AWARE LOSS WITH GRADIENT INTERACTION

During training, the two views are optimized jointly with a role-aware loss. We first compute a cross-view baseline as the global mean reward over all trajectories from both views, giving a baseline for each view; the advantages, policy gradient losses, and total loss are then defined as follows:

$$\bar{R} = \frac{1}{N_{\text{ori}} + N_{\text{rev}}} \left(\sum_{i=1}^{N_{\text{ori}}} R_{\text{ori},i} + \sum_{j=1}^{N_{\text{rev}}} R_{\text{rev},j} \right) \tag{5}$$

$$A_{\text{ori}} = R_{\text{ori}} - \bar{R}, \quad A_{\text{rev}} = R_{\text{rev}} - \bar{R} \tag{6}$$

$$L_{\text{ori}} = -A_{\text{ori}} \cdot P(\tau_{\text{ori}}), \quad L_{\text{rev}} = -A_{\text{rev}} \cdot P(\tau_{\text{rev}}) \tag{7}$$

$$L_{\text{total}} = \frac{N_{\text{ori}} \cdot L_{\text{ori}} + N_{\text{rev}} \cdot L_{\text{rev}}}{N_{\text{ori}} + N_{\text{rev}}} \tag{8}$$

Gradient Decomposition. The total loss in Eq. 8 is a weighted average of the two view losses, and the cross-view baseline $\bar{R}$ introduces an interaction between the two views. Let $N_{\text{total}} = N_{\text{ori}} + N_{\text{rev}}$. The gradient of the total loss with respect to the shared policy parameters θ can be decomposed as:

$$\nabla_\theta L_{\text{total}} = \underbrace{\frac{1}{2}\nabla_\theta(L_{\text{ori}} + L_{\text{rev}})}_{\text{Shared policy update}} + \underbrace{\frac{N_{\text{ori}} - N_{\text{rev}}}{2N_{\text{total}}}\nabla_\theta(L_{\text{ori}} - L_{\text{rev}})}_{\text{Gradient balancing term}} \tag{9}$$

This decomposition reveals two key insights. First, the *shared policy update* ensures that both views contribute to a common descent direction, preventing the model from overfitting to the bias of a single view. Second, the *gradient balancing term* reweights the difference between the two view-specific gradients, balancing their contributions when $N_{\text{ori}} \neq N_{\text{rev}}$. Together, these two terms couple the two views through a shared optimization objective. Full derivation is provided in Appendix A.4.

Coupling Mechanism. In this work, we define *coupling* operationally as the super-additive effect between the two asymmetry types: a method that addresses only one type suffers a performance drop under combined shifts that exceeds the sum of the drops under each individual shift. The gradient balancing term in Eq. (9) provides a mechanism for this coupling: it forces the shared policy to reconcile two structurally distinct views. More importantly, substituting the global mean $\bar{R}$ into the advantage computation:

$$A_{\text{ori},i} = R_{\text{ori},i} - \bar{R} \quad (i = 1, \ldots, N_{\text{ori}}), \qquad A_{\text{rev},j} = R_{\text{rev},j} - \bar{R} \quad (j = 1, \ldots, N_{\text{rev}}). \tag{10}$$

Although the two views may have different numbers of valid starts ($N_{\text{ori}} \neq N_{\text{rev}}$), summing the advantages within each view gives

$$\sum_{i=1}^{N_{\text{ori}}} A_{\text{ori},i} = -\sum_{j=1}^{N_{\text{rev}}} A_{\text{rev},j}, \tag{11}$$

since both sums are relative to the global mean $\bar{R}$ over all trajectories. The two views thus provide negatively correlated advantages, forming a mutual regularization: minimizing L_{total} forces the policy to perform well on both views. This reduces gradient variance and accelerates convergence (Fig. 2). Removing this interaction (replacing the cross-view baseline with a view-specific one) causes a substantial drop (Table 5).

5 Experimental Results and Analysis

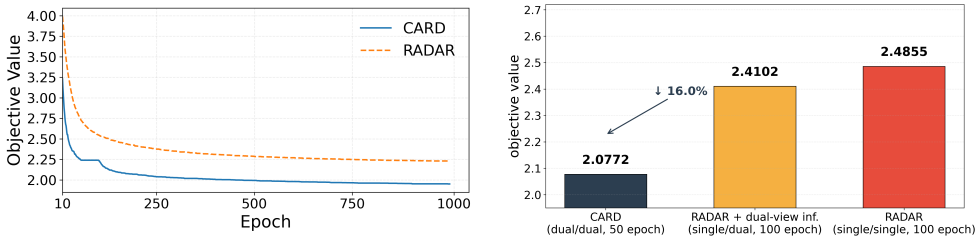
5.1 Experimental Setup

We evaluate CARD across five dimensions: training efficiency, in-distribution, cross-scale generalization, OOD generalization, and real-world routing, followed by ablation studies. We report obj. (average cost on 1000 instances), gap (relative to CARD), and inference time (minutes, "m") per 1,000 instances, using greedy decoding. Averages use unrounded values; minor discrepancies may appear from displayed figures.

For synthetic instances, we generate asymmetric cost matrices and demands following the standard procedure in MatNet and RADAR; full details are provided in Appendix A.5. For cross-scale evaluation, we train on 20/50/100 nodes and test on 200/500/1000 without fine-tuning. For OOD, we apply biased cost (low/high asymmetry) and biased demand shifts, as well as their combinations. For real-world evaluation, we use the real-world distance matrices from the RRNCO benchmark Son et al. (2026), on which we construct two of our problem variants (AMPDTSP and AVRPMB).

All models are trained for 1000 epochs with 10000 new instances per epoch. We compare CARD against RADAR, MatNet, and RRNCO, and additionally evaluate two OR baselines: LKH-3 Helsgaun (2017) on APDTSP (synthetic, n=20/50, MAX_TRIALS=1000), and OR-Tools on AMPDTSP (RRNCO, 10s per instance). LKH-3 is not applied to larger or role-constrained variants, as it rarely yields feasible solutions at 100 nodes, and OR-Tools is not applied to AVRPMB, whose multi-trip reloading structure its routing model cannot represent. For AVRPSDP100, we do not use Sinkhorn normalization because it diverges completely on this problem variant; we apply the same softmax substitution to both CARD and RADAR to ensure a fair comparison. For inference, we apply input shuffling as augmentation (Appendix A.6). The gap and obj. results use fixed augmentation counts per method (CARD aug10, RADAR aug20, MatNet aug40); the time row in Table 1 is measured under a unified time budget by adjusting augmentation counts per method. All experiments are run on

a single NVIDIA RTX 5090 GPU with 32GB memory; hyperparameters are listed in Appendix A.7 (Table 8). Training converges substantially faster than RADAR, reaching its 1000-epoch performance by epoch 106, and outperforms RADAR with dual-view inference by 16.0% under matched wall-clock time (Fig. 2; Appendix A.8).



(a) Convergence Curves

(b) Comparison at Equal Training Time

Figure 2: Training efficiency on AMPDTSP100. Left: convergence curves. Right: comparison of CARD, RADAR, and RADAR with dual-view inference under matched wall-clock time.

5.2 Main Result

Table 1: In-distribution performance on AMPDTSP, APDTSP, AVRPMB, and AVRPSDP.

	n=20	n=50	n=100	n=20	n=50	n=100
	AMPDTSP			APDTSP		
MatNet (aug40) gap	3.66%	16.94%	21.20%	2.55%	12.82%	18.69%
RRNCO (aug20) gap	3.89%	11.53%	18.63%	3.35%	1.67%	4.51%
RADAR (aug20) gap	4.96%	9.05%	14.35%	1.41%	1.21%	1.22%
CARD (aug10) obj.	1.8066	1.8496	1.9086	1.7747	1.8214	1.8681
time	0.07m	0.20m	0.38m	0.07m	0.20m	0.38m
	AVRPMB			AVRPSDP		
MatNet (aug40) gap	7.03%	11.97%	34.74%	3.85%	6.61%	9.37%
RRNCO (aug20) gap	5.51%	6.52%	12.52%	3.09%	3.16%	5.08%
RADAR (aug20) gap	4.03%	3.26%	5.30%	1.54%	1.80%	4.03%
CARD (aug10) obj.	1.6364	1.6280	1.6323	2.3396	2.3646	2.3673
time	0.07m	0.20m	0.38m	0.06m	0.22m	0.40m

Table 1 reports the in-distribution performance. CARD consistently achieves the lowest cost across all variants and scales. The largest gains over RADAR are on AMPDTSP100, AVRPMB100, and AVRPSDP100, with the gap generally growing with instance size. On APDTSP, where role reversal is not applicable, CARD consistently outperforms RADAR, confirming that the distance-aware design remains effective. For reference, LKH-3 achieves 1.750 on APDTSP50 with a 100% success rate but drops to 47% success at APDTSP100, and its runtime is orders of magnitude higher than neural solvers; we therefore exclude it from the tables. The full table with absolute objective values is provided in Appendix Table 11. Multi-seed stability on AMPDTSP20 is reported in Appendix A.9 (Table 10), showing a standard deviation below 0.5%.

5.3 Cross-Scale Generalization

Table 2 reports the cross-scale generalization gaps. MatNet is omitted for evaluation scales exceeding 256 nodes (i.e., 50-500, 100-500, and 100-1000) due to its fixed one-hot embedding dimension, which caps the maximum number of nodes it can handle. MatNet's Average covers only the two scales it can handle. RADAR shows large gaps on role-asymmetric problems, with gaps up to 33.48% on AMPDTSP, 11.71% on AVRPMB, and 16.93% on AVRPSDP. CARD consistently outperforms both RADAR and the other baselines (MatNet and RRNCO) across all variants. Complete cross-scale results are in Appendix Table 12, including APDTSP and absolute values.

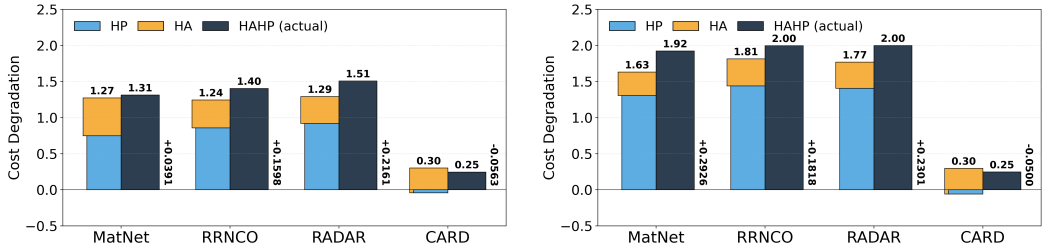
Table 2: Cross-scale generalization gaps on AMPDTSP, APDTSP, and AVRPMB.

	Method	20-200	50-500	100-200	100-500	100-1000	Average
AMPDTSP	MatNet gap	71.14%	–	73.11%	–	–	72.13%
	RRNCO gap	33.24%	37.63%	11.39%	35.65%	27.71%	29.12%
	RADAR gap	27.45%	33.48%	7.76%	33.43%	16.44%	23.71%
	CARD obj.	2.7887	2.8298	2.0587	2.3947	3.6219	–
AVRPMB	MatNet gap	98.36%	–	65.43%	–	–	81.90%
	RRNCO gap	5.28%	13.51%	7.68%	5.90%	4.08%	7.29%
	RADAR gap	4.26%	11.71%	2.65%	2.93%	2.01%	4.71%
	CARD obj.	2.4640	2.1474	1.6854	1.8059	2.0154	–
AVRPSDP	MatNet gap	75.64%	–	70.67%	–	–	73.16%
	RRNCO gap	8.67%	10.10%	6.58%	15.49%	24.89%	13.15%
	RADAR gap	6.92%	7.81%	5.33%	13.22%	16.93%	10.04%
	CARD obj.	2.6343	2.6264	2.1425	2.2248	2.7086	–

5.4 Out-of-Distribution Generalization

Table 3: Out-of-distribution (OOD) generalization gaps on four problems.

	n=20	n=50	n=100	n=20	n=50	n=100
	AMPDTSP LA,HA			APDTSP LA,HA		
MatNet	2.84%	19.81%	46.82%	4.90%	21.46%	42.89%
RRNCO	9.65%	26.74%	24.65%	3.27%	3.16%	2.99%
RADAR	10.23%	27.26%	13.70%	1.47%	1.85%	1.84%
CARD	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	AVRPMB LA,HA,HL,HB,HAHL,HAHB			AVRPSDP LA,HA,HD,HP,HAHD,HAHP		
MatNet	13.76%	18.99%	32.92%	13.76%	19.93%	28.85%
RRNCO	11.30%	8.72%	12.99%	11.47%	17.02%	26.50%
RADAR	9.58%	7.29%	8.21%	9.58%	16.77%	25.68%
CARD	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%



(a) AVRPSDP50 (b) AVRPSDP100

Figure 3: Super-additive effect on AVRPSDP50 and AVRPSDP100.

Table 3 reports averaged OOD gaps. LA/HA denote low/high cost asymmetry (scaling long edges by 1.2/1.5); HL/HB (AVRPMB) and HD/HP (AVRPSDP) denote demand shifts (reducing the non-dominant role demand range from [1,10] to [1,5]). CARD remains stable across all shifts, while RADAR degrades significantly when cost and demand shifts are combined. The super-additive effect is consistently observed across all single-asymmetry baselines on both AVRPSDP50 and AVRPSDP100, while CARD shows no such super-additivity at either scale. Per-shift breakdowns and detailed analysis are provided in Appendix A.13 (Fig. 3).

5.5 REAL-WORLD ROUTING RESULT

Table 4: Real-world routing performance on the RRNCO benchmark.

		ID		OOD		cluster		Average
		obj.	gap	obj.	gap	obj.	gap	gap
AMPDTSP	OR-Tools	78.93	14.01%	85.39	22.24%	35.16	64.67%	33.64%
	MatNet	76.54	10.56%	77.16	10.46%	31.64	48.20%	23.07%
	RRNCO	73.22	5.76%	73.65	5.43%	28.17	31.92%	14.37%
	RADAR	72.97	5.40%	73.89	5.78%	27.43	28.49%	13.22%
	CARD	69.23	0.00%	69.85	0.00%	21.35	0.00%	0.00%
AVRPMB	MatNet	61.24	17.04%	62.45	18.80%	19.47	21.51%	19.11%
	RRNCO	57.32	9.55%	58.10	10.52%	17.24	7.61%	9.23%
	RADAR	55.26	5.60%	55.67	5.89%	16.92	5.62%	5.71%
	CARD	52.32	0.00%	52.57	0.00%	16.02	0.00%	0.00%

Table 4 evaluates CARD on the RRNCO benchmark under three distribution settings. CARD outperforms all baselines on both AMPDTSP and AVRPMB, including RADAR, the strongest existing neural solver. On all 1280 instances, CARD completes inference in about 20 seconds, whereas OR-Tools requires 10 seconds per instance. This combination of solution quality and inference speed makes CARD practical for real-world routing.

5.6 ABLATION STUDIES

Table 5: Ablation study on AMPDTSP and AVRPMB.

	AMPDTSP100		AVRPMB100	
	obj.	gap	obj.	gap
RADAR	2.2277	14.38%	1.7715	6.65%
CARD-A	2.0394	4.71%	1.6968	2.16%
CARD-R	2.1385	9.80%	1.7472	5.19%
CARD	1.9476	0.00%	1.6610	0.00%

Table 5 ablates the two core components of CARD. CARD-A removes the distance-aware mechanism from the encoder (Sec. 4.1); CARD-R removes the role-aware mechanism in both training and inference (Secs. 4.2 and 4.3). Both reduce the model to MatNet style, and both drops confirm their necessity, with CARD-R's larger drop identifying the role-aware design as the primary driver. To further isolate training from inference, we compare RADAR with dual-view inference (Appendix A.8), confirming that the gain comes from dual-view training rather than inference-time selection. The full CARD achieves the best performance on both problems.

6 CONCLUSION

We study real-world VRP variants where asymmetric distance and role asymmetry naturally coexist. We identify that addressing either asymmetry in isolation is insufficient, and propose CARD, a framework that handles both jointly. By introducing a distance-aware attention mechanism and a role-aware training algorithm, our method consistently outperforms the state-of-the-art neural solver RADAR, achieving superior solution quality with improved efficiency. More importantly, CARD remains robust under real-world distribution shifts where both asymmetries are present, with the largest gains over baselines occurring precisely when both are active. A limitation is that dual-view rollout doubles per-epoch training time, though this is offset by the far fewer epochs required. For future work, we plan to extend CARD to broader vehicle routing problems.

AI USE STATEMENT

In this work, we used generative AI tools for code debugging and LaTeX formatting. We have reviewed all AI-assisted work and take full responsibility for the final content, including text, claims, and experimental results.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or sensitive applications. All experiments are conducted on synthetic and publicly available benchmark datasets. We foresee no direct negative societal impacts; the proposed method aims to improve logistics efficiency, which can contribute to reduced fuel consumption and operational costs. There are no conflicts of interest to declare.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the problem formulation (Sec. 3), model architecture (Sec. 4.1), training algorithm (Sec. 4.3 and Algorithm 1), and experimental settings (Sec. 5 and Appendix Table 8). All datasets are generated using the procedure described in Sec. 5 with fixed random seeds. Source code is submitted as supplementary material and will be released anonymously upon publication. Complete numerical results with absolute objective values are provided in Appendix Tables 11–14.

AUTHOR CONTRIBUTIONS

Omitted for double-blind review.

ACKNOWLEDGMENTS

Omitted for double-blind review.

REFERENCES

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5): 679–684, 1957.

Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

Junchuang Cai et al. A survey of dynamic pickup and delivery problems. *Neurocomputing*, 554: 126631, 2023.

Rui Cao, Zhiguang Cao, Yihan Huang, Jiaqi Wang, Yuan Jiang, Yubin Xiao, and You Zhou. Learning to handle constrained routing problems from a decoupling perspective. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026.

N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi (eds.), *Combinatorial Optimization*, pp. 315–338. John Wiley and Sons, Chichester, 1979.

I. Deif and L. Bodin. Extension of the clarke and wright algorithm for the vehicle routing problem with backhauls. *Proceedings of the 1984 Annual Meeting of the Transportation Research Board*, 1984.

Keld Helsgaun. An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, 12, 2017.

Ziwei Huang, Jianan Zhou, Zhiguang Cao, and Yixin Xu. Rethinking light decoder-based solvers for vehicle routing problems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4pRwkYpa2u.

Minsu Kim, Junyoung Park, and Jinkyoo Park. Sym-nco: Leveraging symmetricity for neural combinatorial optimization. *Advances in Neural Information Processing Systems*, 35:1936–1949, 2022.

Detian Kong, Yining Ma, Zhiguang Cao, Tianshu Yu, and Jianhua Xiao. Efficient neural collaborative search for pickup and delivery problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11019–11034, 2024.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. Pomo: Policy optimization with multiple optima for reinforcement learning. *Advances in neural information processing systems*, 33:21188–21198, 2020.

Yeong-Dae Kwon, Jinho Choo, Iljoo Yoon, Minah Park, Duwon Park, and Youngjune Gwon. Matrix encoding networks for neural combinatorial optimization. *Advances in Neural Information Processing Systems*, 34:5138–5149, 2021.

Haibing Li and Andrew Lim. A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(2):173–186, 2003.

Jingwen Li, Liang Xin, Zhiguang Cao, Andrew Lim, Wen Song, and Jie Zhang. Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2306–2315, 2021.

Jinqi Li, Bing Tian Dai, Yunyun Niu, Jianhua Xiao, and Yaoxin Wu. Multi-type attention for solving multi-depot vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):17831–17840, 2024a.

Jinqi Li, Yunyun Niu, Guodong Zhu, and Jianhua Xiao. Solving pick-up and delivery problems via deep reinforcement learning based symmetric neural optimization. *Expert Systems with Applications*, 255:124514, 2024b.

Shengcai Liu, Yu Zhang, Ke Tang, and Xin Yao. How good is neural combinatorial optimization? a systematic evaluation on the traveling salesman problem. *IEEE Computational Intelligence Magazine*, 18(3):14–28, 2023.

Fu Luo, Xi Lin, Fei Liu, Qingfu Zhang, and Zhenkun Wang. Neural combinatorial optimization with heavy decoder: Toward large scale generalization. *Advances in Neural Information Processing Systems*, 36:8845–8864, 2023.

Yining Ma, Jingwen Li, Zhiguang Cao, Wen Song, Hongliang Guo, Yuejiao Gong, and Yeow Meng Chee. Efficient neural neighborhood search for pickup and delivery problems. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4776–4784, 2022. doi: 10.24963/ijcai.2022/662.

Dian Meng, Yaoxin Wu, Yaqing Hou, and Zhiguang Cao. PomtVRS: Preference-optimized multitask vehicle routing solver with preference gating. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vJsJPNRFIJ.

Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. A survey on pickup and delivery problems. part I: Transportation between customers and depot. *Journal für Betriebswirtschaft*, 58(1):21–51, 2008. doi: 10.1007/s11301-008-0033-7.

Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. *arXiv preprint arXiv:2110.11773*, 2022.

M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1):17–29, 1995.

Jiwoo Son, Zhikai Zhao, Federico Berto, Chuanbo Hua, Zhiguang Cao, Changhyun Kwon, and Jinkyoo Park. Towards real-world routing with neural combinatorial optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=sKvo9ZZfpe.

Conghui Wang, Zhiguang Cao, Yaoxin Wu, Long Teng, and Guohua Wu. Deep reinforcement learning for solving vehicle routing problems with backhauls. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):4779–4793, 2025. ISSN 2162-237X. doi: 10.1109/TNNLS.2024.3371781.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Hang Yi, Ziwei Huang, Zhiguang Cao, and Yining Ma. Radar: Learning to route with asymmetry-aware distance representations. In *International Conference on Learning Representations (ICLR)*, 2026.

A Appendix

A.1 Detailed Problem Definitions

Table 6 summarizes the four asymmetric VRP variants considered in this work, including their key constraints.

Table 6: Summary of problem variants.

	AMPDTSP	**APDTSP**	**AVRPMB**	**AVRPSDP**
Pickup/Delivery pairing	yes	yes	no	no
Capacity	yes	no	yes	yes
Linehaul/Backhaul roles	n/a	n/a	yes	no
Simultaneous pickup+delivery	no	no	no	yes
Load constraint	$[0, Q]$	n/a	$[0, Q]$	$[0, Q]$ throughout

AMPDTSP. Each request consists of a pickup node and a paired delivery node, with the precedence constraint that the pickup must be visited before its paired delivery. For each node i, let $p_i \geq 0$ denote the pickup demand (if node i is a pickup node) and $d_i \geq 0$ denote the delivery demand (if node i is a delivery node); nodes serving as pickups have $p_i > 0, d_i = 0$, and delivery nodes have $d_i > 0, p_i = 0$. The vehicle capacity Q restricts the total load at any point to $[0, Q]$, where the load changes by $+p_i$ at pickup nodes and $-d_i$ at delivery nodes. The precedence constraint $t_{\text{pickup}} < t_{\text{delivery}}$ is enforced for each request.

APDTSP. Special case of AMPDTSP with no capacity constraint ($Q = \infty$), i.e., only the precedence constraint remains.

AVRPMB. Each customer i is assigned a role: linehaul (delivery) or backhaul (pickup). Let $d_i > 0$ denote the delivery demand for linehaul customers and $p_i > 0$ denote the pickup demand for backhaul customers; each node has only one role. The vehicle load changes by $-d_i$ at linehaul stops and $+p_i$ at backhaul stops, and must remain within $[0, Q]$ throughout the route.

AVRPSDP. Each customer i has both delivery demand $d_i > 0$ and pickup demand $p_i > 0$, both served in a single visit. The vehicle load changes by $-d_i$ (delivery) and then $+p_i$ (pickup) at each visit, and must remain within $[0, Q]$ throughout the route.

A.2 MLP Attention Visualization

To provide empirical evidence for the conceptual decomposition in Eq. 3 of the main paper, we visualize the MLP-based attention scorer on 10 AMPDTSP20 instances, aggregating all node pairs

across instances. We extract the raw attention scores before Sinkhorn normalization for three types of node pairs: complementary roles (pickup→delivery), same pickup roles (pickup→pickup), and same delivery roles (delivery→delivery). These are encoder-side scores and are not subject to the decoder-side logit clipping. For each pair, we vary the forward cost C_{ij} while fixing other inputs.

The complementary pairs (P→D) contain 100 node pairs, while the same-role pairs (P→P and D→D) each contain 90 pairs. Figure 4 shows that the P→D curve has a slope whose magnitude is 21–29× larger than that of the same-role curves, indicating that the MLP is substantially more sensitive to directional cost when roles are complementary. This confirms the role-modulated behavior described in the main paper.

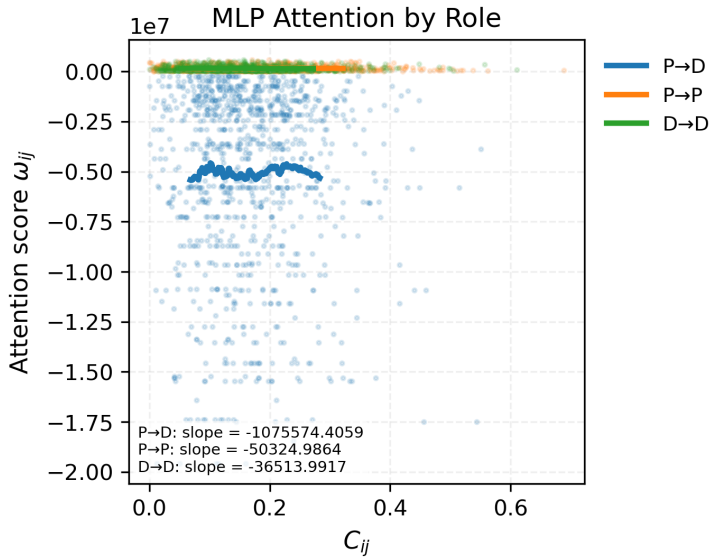


Figure 4: MLP attention scores vs. forward cost C_{ij} for different role pairs. The P→D curve exhibits a slope 21–29× steeper than same-role curves, confirming role-modulated attention.

A.3 Dual-View Rollout Details

Both the original and role-reversed views evaluate costs using the same cost matrix C traversed in the forward direction. Role reversal only swaps the role labels $(d_i, p_i) \mapsto (p_i, d_i)$, leaving the distance structure unchanged. The sequence produced under the role-reversed view, when read in reverse order, corresponds to a feasible route under the original problem, and its cost is obtained by reading C in the forward direction. This ensures that the rewards from both views are on a consistent scale, allowing the shared cross-view baseline to provide valid advantage estimates.

A.4 Derivation of Gradient Decomposition

This section provides the detailed derivation of Eq. (9) in the main paper.

Starting from the definition of L_{total} in Eq. (8), let $N_{\text{total}} = N_{\text{ori}} + N_{\text{rev}}$. The gradient with respect to the shared policy parameters θ is:

$$\nabla_\theta L_{\text{total}} = \frac{N_{\text{ori}}}{N_{\text{total}}} \nabla_\theta L_{\text{ori}} + \frac{N_{\text{rev}}}{N_{\text{total}}} \nabla_\theta L_{\text{rev}} \tag{12}$$

$$= \frac{1}{2} \nabla_\theta (L_{\text{ori}} + L_{\text{rev}}) + \frac{N_{\text{ori}} - N_{\text{rev}}}{2N_{\text{total}}} \nabla_\theta (L_{\text{ori}} - L_{\text{rev}}). \tag{13}$$

The first term encourages both views to update in a shared descent direction. The second term acts as an adaptive coupling regularizer: when $N_{\text{ori}} > N_{\text{rev}}$, it amplifies the gradient from the original view relative to the role-reversed view, and vice versa. This dynamically balances the influence of the two views during training, preventing either from dominating the update.

A.5 Instance Generation

Asymmetric cost matrices follow the standard generation procedure used in MatNet and RADAR. We first sample a random integer matrix with entries in $[0, 10^6)$ and set the diagonal to zero. We then apply Floyd-Warshall shortest-path closure to enforce the triangle inequality. Directional asymmetry is introduced by scaling one direction of each edge by a fixed factor α, so that $C_{ij} \neq C_{ji}$ for a controlled fraction of edges. The matrix is finally divided by 10^6.

For AMPDTSP, AVRPMB, and AVRPSDP, demands are sampled uniformly from $\{1, \ldots, 10\}$; the depot has zero demand. For APDTSP, no demands or capacity constraints are applied. Capacity settings by instance size are listed in Table 7. For the role-reversed view, the cost matrix remains unchanged. For AMPDTSP and AVRPMB, the demand vector is negated, i.e., $d_i \mapsto -d_i$; for AVRPSDP, the delivery and pickup channels are swapped.

For OOD evaluation, we consider two types of shifts: (1) cost shifts, which scale long edges by 1.2 (LA) or 1.5 (HA); and (2) demand shifts, which reduce the non-dominant role's demand range from $[1, 10]$ to $[1, 5]$ (HL/HB for AVRPMB, HD/HP for AVRPSDP). Combined shifts (HAHL, HAHB, HAHD, HAHP) apply both simultaneously.

Table 7: Vehicle capacity settings by instance size.

Instance size	20	50	100	200	500	1000
AMPDTSP	30	40	50	60	100	150
AVRPMB	30	40	50	60	150	500
AVRPSDP	30	40	50	150	500	1000

A.6 Augmentation Protocol

For inference, we apply input shuffling as augmentation: each instance is expanded into g shuffled variants with randomized node order. Shuffling preserves the optimal solution but reduces ordering bias by exposing the policy to different input orderings.

The count g is defined per view. CARD evaluates two views (original and role-reversed), so CARD aug10 corresponds to 20 rollouts per instance, while RADAR aug20 and MatNet aug40 correspond to 20 and 40 rollouts, respectively, as they are single-view methods. The counts are not arbitrary: they are set so that the total inference time per instance is approximately equal across methods. Since CARD's dual-view rollout doubles the per-rollout cost, it uses half the augmentations of RADAR. MatNet, with a lighter per-rollout cost, can afford a larger count within the same budget. This ensures that all reported gaps reflect solution quality under a matched computational budget.

A.7 Hyperparameter Settings

Table 8 lists the hyperparameters used across all experiments. All models are trained with the same set of hyperparameters for fair comparison.

A.8 Training Efficiency

Figure 2 in the main paper compares the training efficiency of CARD and RADAR on AMPDTSP100. CARD converges to a significantly lower final objective (1.95 vs. 2.23) and surpasses RADAR's 1000-epoch performance by epoch 106, reducing wall-clock time from 9.5 hours to 1.7 hours. This acceleration comes with a trade-off: CARD doubles the per-epoch cost due to dual-view rollout, but this is more than offset by the drastically fewer epochs required.

To isolate the contribution of dual-view training from that of dual-view inference, we take a RADAR model trained with single-view training, and apply the same dual-view inference as CARD. Table 9 compares three configurations on AMPDTSP100 under matched inference budgets (20 rollouts per instance).

Dual-view inference alone improves RADAR by only 3.1%, whereas CARD outperforms RADAR with dual-view inference by 16.0%. Notably, the benefit of dual-view inference is 14.7% for CARD

Table 8: Hyperparameters used in all experiments.

Hyperparameter	Value
Training epochs	1000
Instances per epoch	10000
Embedding dimension	256
Number of attention heads	8
QKV dimension per head	16
Encoder layers	5
MLP hidden dimension (FeedForward)	512
MLP hidden dimension (MixedScore)	16
SVD rank k	10
Sinkhorn iterations	10
Logit clipping	10
Learning rate	4×10^{-4}
Weight decay	1×10^{-6}
Optimizer	Adam

Table 9: Isolating dual-view training from inference on AMPDTSP100. All configurations use 20 rollouts per instance.

	Train	Infer	FULL SCORE
RADAR	single	single	2.4855
RADAR + dual-view inference	single	dual	2.4102
CARD	dual	dual	2.0772

but only 3.1% for RADAR, confirming that dual-view training teaches the policy to exploit the role-reversed view. Notably, CARD reaches RADAR's final convergence level (2.23) by around epoch 100. After a brief fluctuation at that level, CARD continues to improve and converges to 1.95. RADAR stagnates at 2.23 even after 1000 epochs. This suggests that the role-aware training helps CARD escape the local optimum where RADAR gets stuck. The curves are smoothed for readability.

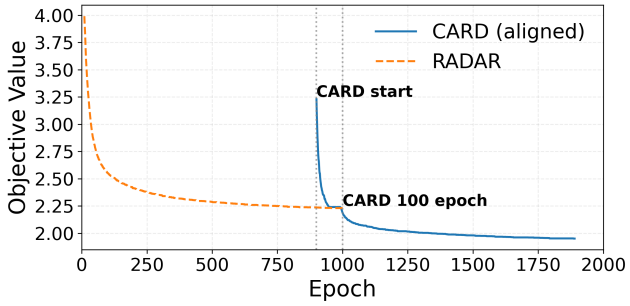


Figure 5: Aligned convergence curves on AMPDTSP100.

A.9 Multi-Seed Stability

To assess training stability, we train CARD with three random seeds on AMPDTSP (n=20). As shown in Table 10, the variation across seeds is below 0.5%, confirming that the reported results are not artifacts of favorable initialization. Multi-seed analysis at larger scales is computationally prohibitive (approximately one day per run on a single RTX 5090); we leave it to future work.

A.10 Complete In-Distribution Results

This section provides the full numerical results corresponding to Table 1 in the main paper, including absolute objective values (average total travel cost) for all methods, all problem variants, and all

Table 10: Multi-seed stability on AMPDTSP (n=20).

	Seed 0	Seed 1	Seed 2	Mean $\pm$ Std
CARD obj.	1.8066	1.8189	1.8028	1.8094 ± 0.0082

scales. All gaps reported in the main paper are computed relative to CARD using these absolute values.

Table 11: Complete in-distribution results with absolute objective values.

	n=20	n=50	n=100	n=20	n=50	n=100
	AMPDTSP			APDTSP		
MatNet (aug40)	1.8727	2.1630	2.3133	1.8200	2.0549	2.2173
RRNCO (aug20)	1.8768	2.0628	2.2641	1.8342	1.8519	1.9524
RADAR (aug20)	1.8962	2.0170	2.1825	1.7997	1.8434	1.8908
CARD (aug10)	1.8066	1.8496	1.9086	1.7747	1.8214	1.8681
	AVRPMB			AVRPSDP		
MatNet (aug40)	1.7514	1.8229	2.1994	2.4297	2.5210	2.5891
RRNCO (aug20)	1.7266	1.7341	1.8367	2.4118	2.4393	2.4876
RADAR (aug20)	1.7024	1.6810	1.7188	2.3756	2.4071	2.4627
CARD (aug10)	1.6364	1.6280	1.6323	2.3396	2.3646	2.3673

A.11 Complete Cross-Scale Generalization Results

This section provides the full numerical results corresponding to Table 2 in the main paper, including absolute objective values for all methods, all cross-scale settings, and all problem variants. All gaps reported in the main paper are computed relative to CARD using these absolute values.

Table 12: Complete cross-scale generalization results with absolute objective values.

	Method	20-200	50-500	100-200	100-500	100-1000	Average
AMPDTSP	MatNet	4.7725	–	3.5639	–	–	4.1682
	RRNCO	3.7156	3.8946	2.2931	3.2483	4.6254	3.5554
	RADAR	3.5542	3.7771	2.2184	3.1952	4.2173	3.3924
	CARD	2.7887	2.8298	2.0587	2.3947	3.6219	2.7388
AVRPMB	MatNet	4.8877	–	2.7882	–	–	3.8380
	RRNCO	2.5942	2.4376	1.8148	1.9125	2.0976	2.1713
	RADAR	2.5690	2.3988	1.7301	1.8589	2.0559	2.1225
	CARD	2.4640	2.1474	1.6854	1.8059	2.0154	2.0236
APDTSP	MatNet	4.9837	–	3.2637	–	–	4.1237
	RRNCO	2.6715	2.5643	2.1295	2.3241	2.8126	2.5004
	RADAR	2.6063	2.5316	2.0180	2.2877	2.7729	2.4433
	CARD	2.5417	2.4899	1.9923	2.2473	2.5770	2.3696
AVRPSDP	MatNet	4.6269	–	3.6567	–	–	4.1418
	RRNCO	2.8628	2.8916	2.2835	2.5694	3.3827	2.7980
	RADAR	2.8167	2.8314	2.2567	2.5189	3.1673	2.7182
	CARD	2.6343	2.6264	2.1425	2.2248	2.7086	2.4673

A.12 Detailed Out-of-Distribution Results

This section provides the complete breakdown of OOD performance by shift type. Table 3 in the main paper reports averaged gaps over all shifts. The following tables disaggregate these results per problem. The shift types are defined as follows: LA (low cost asymmetry) and HA (high cost asymmetry) control the severity of directional cost bias; for AVRPMB, HL and HB denote biased linehaul and backhaul demand shifts, respectively, while HAHL and HAHB combine high asymmetry with each demand shift; for AVRPSDP, HD and HP denote biased delivery and pickup demand

shifts, respectively, and HAHD and HAHP combine high asymmetry with each demand shift. All OOD models are evaluated without fine-tuning.

Table 13: Detailed OOD results on AMPDTSP and APDTSP.

	LA			HA		
	n=20	n=50	n=100	n=20	n=50	n=100
AMPDTSP						
MatNet gap	2.95%	18.11%	46.62%	2.73%	21.50%	47.03%
RRNCO gap	10.88%	25.92%	24.39%	8.42%	27.57%	24.92%
RADAR gap	9.52%	24.95%	14.04%	10.94%	29.58%	13.37%
CARD obj.	1.9406	1.9755	2.0401	2.1069	2.0913	2.2114
APDTSP						
MatNet gap	4.57%	23.32%	39.97%	5.22%	19.59%	45.82%
RRNCO gap	3.32%	3.99%	2.09%	3.21%	2.33%	3.88%
RADAR gap	1.68%	1.74%	2.03%	1.27%	1.96%	1.64%
CARD obj.	1.9769	2.0918	1.9873	2.1261	1.9704	2.0933

Table 13 reports detailed results on AMPDTSP and APDTSP under low (LA) and high (HA) cost asymmetry. For each problem, the CARD row shows absolute objective values; all other rows show gap (%) relative to CARD.

Table 14: Detailed OOD results on AVRPMB100 and AVRPSDP50 by shift type.

AVRPMB100						
	LA	HA	HL	HB	HAHL	HAHB
MatNet gap	40.65%	44.23%	23.98%	30.85%	25.09%	32.71%
RRNCO gap	12.25%	10.52%	7.93%	20.85%	1.06%	25.35%
RADAR gap	6.98%	9.12%	0.19%	16.25%	0.30%	16.40%
CARD obj.	1.7172	1.7881	2.1878	2.1951	2.4682	2.4762
AVRPSDP50						
	LA	HA	HD	HP	HAHD	HAHP
MatNet gap	7.71%	12.37%	4.69%	40.86%	7.08%	46.84%
RRNCO gap	2.71%	4.23%	2.40%	42.04%	3.54%	47.19%
RADAR gap	1.35%	2.56%	1.19%	43.25%	2.38%	49.92%
CARD obj.	2.5861	2.7100	2.3117	2.3216	2.6312	2.6107

Table 14 reports detailed results on AVRPMB100 and AVRPSDP50 under six shift types: low/high cost asymmetry (LA/HA), biased demand (HL/HB for AVRPMB, HD/HP for AVRPSDP), and their combinations. We report the per-shift breakdown at the most representative scale for each variant (n=100 for AVRPMB, n=50 for AVRPSDP, where the super-additive effect is most pronounced); averaged gaps across all scales are reported in Table 3.

A.13 Super-Additive Effect Analysis

Figure 3 visualizes the super-additive effect reported in Table 3 of the main paper. Table 15 lists the absolute costs and degradations of all methods on AVRPSDP50 and AVRPSDP100 under individual and combined shifts.

On AVRPSDP50, all single-asymmetry baselines exhibit super-additive degradation: MatNet, RRNCO, and RADAR have HAHP degradations of 1.31, 1.40, and 1.51, each exceeding the additive sum of their HA and HP degradations (1.27, 1.24, and 1.29). CARD shows the opposite pattern: its degradation under HAHP (0.25) remains below the sum of its individual degradations (0.35 under HA and -0.04 under HP). Notably, the demand shift reduces the non-dominant role's demand range, which objectively simplifies the problem; CARD benefits from this simplification (its cost under HP falls below its in-distribution cost), whereas the baselines' costs rise substantially, indicating that they cannot exploit the simplified demand structure without modeling roles.

The same pattern holds at $n = 100$: MatNet, RRNCO, and RADAR all exhibit super-additive degradation (1.92, 2.00, and 2.00 vs. additive sums of 1.63, 1.81, and 1.77), while CARD remains sub-additive (0.25 vs. 0.30).

Table 15: Absolute costs and degradations relative to in-distribution performance on AVRPSDP50 and AVRPSDP100. The Excess column shows $\Delta_{\text{HAHP}} - (\Delta_{\text{HA}} + \Delta_{\text{HP}})$; positive values indicate super-additive degradation, negative values indicate sub-additive.

		ID	HA	HP	HAHP	Excess
AVRPSDP50	MatNet obj.	2.5210	3.0451	3.2703	3.8335	
	MatNet Δ	–	0.5241	0.7493	1.3125	$+0.0391$
	RRNCO obj.	2.4393	2.8246	3.2975	3.8426	
	RRNCO Δ	–	0.3853	0.8582	1.4033	$+0.1598$
	RADAR obj.	2.4071	2.7793	3.3257	3.9140	
	RADAR Δ	–	0.3722	0.9186	1.5069	$+0.2161$
	CARD obj.	2.3646	2.7100	2.3216	2.6107	
	CARD Δ	–	0.3454	-0.0430	0.2461	-0.0563
AVRPSDP100	MatNet obj.	2.5891	2.9151	3.8942	4.5128	
	MatNet Δ	–	0.3260	1.3051	1.9237	$+0.2926$
	RRNCO obj.	2.4876	2.8637	3.9258	4.4837	
	RRNCO Δ	–	0.3761	1.4382	1.9961	$+0.1818$
	RADAR obj.	2.4627	2.8275	3.8676	4.4625	
	RADAR Δ	–	0.3648	1.4049	1.9998	$+0.2301$
	CARD obj.	2.3673	2.7233	2.3087	2.6147	
	CARD Δ	–	0.3560	-0.0586	0.2474	-0.0500