MULTI-AGENT ROUTE CONSTRUCTION FOR ROUTING FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Routing foundation models pretrain one policy across vehicle routing variants and transfer it when constraints, fleets or objectives change. We develop a multi-agent interface organized around vehicle and partial-route states and route–customer extensions. Allocation-driven rollouts diversify pretraining of separate source backbones on 16 CVRP variants, and configuration comparisons link multi-depot transfer to route-slot tokens and a geometric prior. We evaluate weight-frozen transfer under supplied constraint features and feasibility masks, and fine-tuning with new vehicle attributes and objectives. Frozen reuse gives in-distribution costs comparable to the strongest baseline, a 17.19% mean gap on 32 unseen variants at N=100 against 34.89% for that baseline with the lowest gap on 28 of them, and the lowest neural gap on 37/65 CVRPLib instances. At 1M fine-tuning samples, our adaptation pipeline has lower costs than both adapted multi-task baselines in all min-max cells and in three of four min-sum cells. Within our architecture and a fixed recipe, pretrained fine-tuning at 200K target exposures reaches lower mean cost than scratch training at 1M in each of three fleet settings over three downstream seeds.

1 INTRODUCTION

Routing foundation models pretrain across vehicle routing problem (VRP) variants to reuse weights when constraints change (Liu et al., 2024b; Zhou et al., 2024; Berto et al., 2025a). We argue that the decisions learned in common should be route–customer extensions over explicit vehicle and partial-route states, which also give new vehicle, constraint and objective information a place to enter. We study an interface organized around evolving vehicle and partial-route states, with each action choosing both the route to extend and its next target.

Multi-variant models already transfer to unseen constraint combinations (Gui et al., 2026); RouteFinder adds input adapters (Berto et al., 2025a), and GOAL fine-tunes across combinatorial problem types (Drakulic et al., 2025). Vehicle-aware models provide related construction primitives: JAMPR and 2D-Ptr jointly select a route and customer (Falkner & Schmidt-Thieme, 2020; Liu et al., 2024c), PARCO communicates between agents (Berto et al., 2025b), and ECHO conditions on the previously selected vehicle (Wu et al., 2026b). We use these precedents to study which route-construction interface to pretrain and how its learned decisions transfer; our experiments also adapt RouteFinder and CCL to heterogeneous fleets.

Our interface maintains a token per route slot, customer tokens and dynamic partial-route queries. Constraint and vehicle attributes enter these representations, while the shared action extends one route by a feasible customer or depot. The design hypothesis is that persistent route state provides a place for changing vehicle information while preserving the operations that score and execute extensions.

Within a rollout, route queries exchange current states, attend to feasible targets and score route–customer pairs with a non-learned distance prior; one softmax selects one pair (Figure 1). We pretrain separate $N = 50$ and $N = 100$ backbones on 16 CVRP variants. Rollouts vary their active-slot counts and sample initial pairs, and REINFORCE compares them through a shared baseline. Reuse then either freezes weights under supplied target features and feasibility masks or fine-tunes with vehicle inputs and an objective-specific reward.

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.

Frozen reuse yields a 17.19% mean gap on 32 unseen variants at $N = 100$ against CCL's 34.89%, and the lowest gap among the compared systems on 28 of them (Table 1). With 1M fine-tuning samples and 1,280 candidates, adaptation gives lower costs than RF-TE in all four Li min-sum cells and CCL in three, with CCL lower at N40/M3; min-max costs are lower than both in all three fleets (Table 3). Separately, fixing our architecture and recipe shows lower three-seed mean costs after 200K pretrained fine-tuning exposures than 1M scratch exposures in each fleet (Figure 2).

Our contributions are:

- A multi-agent route-construction interface for multi-variant pretraining that scores all route–customer pairs with one softmax, rather than choosing a vehicle before a node or several pairs at once, extends one pair per step over persistent route-slot tokens with route communication, and trains with allocation-driven rollouts; configuration comparisons link multi-depot transfer to the slot tokens and geometric prior, and ID quality to allocation and communication (Table 4).
- Evidence of weight-frozen reuse under unseen constraints and distributions, and fleet- and objective-specific adaptation quality against RF-TE and CCL at a stated budget.
- A within-architecture test of pretraining value, comparing pretrained and scratch initialization under a common target-training recipe.

2 Related Work

Multi-variant pretraining and adaptation. MTPOMO, MVMoE and CaDA share policies across VRP constraint combinations through attribute inputs, experts or constraint prompts (Liu et al., 2024b; Zhou et al., 2024; Li et al., 2025). CoEKS assigns experts to constraints and shares knowledge across them for unseen combinations (Yu et al., 2026), mixed-curvature pretraining fuses features in mixed-curvature spaces (Liu et al., 2025b), and bandit-based task sampling makes multi-task training more efficient (Wang et al., 2025). CCL uses stepwise constraint context and evaluates the 16-to-32 variant split used here (Gui et al., 2026). RouteFinder pretrains on 16 variants and introduces Efficient Adapter Layers, zero-initialized input extensions that we use for vehicle attributes (Berto et al., 2025a). Across problem types, Cross-Problem Learning studies full and adapter-only transfer (Lin et al., 2024), and GOAL fine-tunes across combinatorial problems (Drakulic et al., 2025). Within routing, learned prompts adapt a pretrained solver to new distributions without retraining (Liu et al., 2024a), and TuneNSearch pretrains on multi-depot VRP, fine-tunes to other variants and refines solutions by local search (Corrêa et al., 2026c). RF-TE and CCL use our HCVRP adapters (Section B.6); their results compare adaptation pipelines, rather than their authors' HCVRP implementations.

Explicit vehicle and route construction. Multi-agent reinforcement learning has been applied to cooperative pickup and delivery (Zong et al., 2022), multi-vehicle routing with soft time windows (Zhang et al., 2020) and solution rewriting by agents that observe only their own costs (Paul et al., 2022). MARDAM uses sequential multi-agent decisions for dynamic routing (Bono et al., 2021); for HCVRP, Li et al. (2022) select a vehicle then a node under min-sum and min-max objectives, token-based and rank-augmented solvers followed (Wang et al., 2024; Liu et al., 2025a), and Wan et al. (2025) learn the fleet size and mix together with the routes. JAMPR and 2D-Ptr jointly select route–customer pairs (Falkner & Schmidt-Thieme, 2020; Liu et al., 2024c); 2D-Ptr includes inter-vehicle attention, ECHO conditions node embeddings on the previous vehicle (Wu et al., 2026b), and the Multi-Agent Pointer Transformer routes several vehicles jointly in dynamic pickup and delivery (Zou et al., 2025). PARCO combines communication with parallel actions and conflict handling (Berto et al., 2025b), and GeoPAR adds geometry-guided attention for large instances (Wu et al., 2026a); CAMP and USPR condition on vehicle profiles (Hua et al., 2025; 2026), and MACSIM selects multiple pairs per step (Luttmann & Xie, 2026). Other min-max and multi-agent scheduling models encode agents but fix the acting agent through a sequential or event-driven rule (Cao et al., 2024; Son et al., 2024; Zheng et al., 2024; Park et al., 2023; 2021). Multi-depot solvers use multi-type attention (Li et al., 2024) or match customers to depots before routing (Nasehibasharzad et al., 2025) and are trained on multi-depot instances, whereas our multi-depot results use frozen weights. Vehicle-as-Prompt tests unseen combinations of heterogeneous-fleet constraints (Huang et al., 2026), TOP-Former applies a multi-agent transformer to team orienteering (Fuertes et al., 2025), and MAEnvs4VRP and RL4CO provide environments (Gama et al., 2026; Berto et al., 2025c).

Sequential route-by-route decoding can represent every solution when vehicles are identical and the fleet is unbounded, but the choice of route becomes a decision once vehicles differ. We therefore keep one softmax over route–customer pairs, as JAMPR and 2D-Ptr do, rather than selecting a vehicle before a node or several pairs per step, add route communication and allocation-driven multi-starts, and study multi-variant initialization and reuse.

Generalization and search. Pointer Networks and the attention model underlie construction policies (Vinyals et al., 2015; Kool et al., 2019). POMO varies initial customers under a shared baseline (Kwon et al., 2020), Sym-NCO adds symmetry regularization (Kim et al., 2022), and PolyNet learns diverse strategies with one decoder instead of handcrafted diversity rules (Hottung et al., 2025); our rollouts sample initial pairs and vary active-slot counts. INViT uses invariant local views (Fang et al., 2024) and TTPL test-time input projections (Chen et al., 2025); ELG ensembles a construction policy with a transferable local policy (Gao et al., 2024), ICAM conditions attention on instance scale (Zhou et al., 2026a), ReLD strengthens light decoders (Huang et al., 2025), and UniteFormer unifies node and edge inputs (Meng et al., 2025). Our CVRPLib evaluation freezes weights on generated X instances (Uchoa et al., 2017). We run HGS through PyVRP (Vidal, 2022; Wouda et al., 2024); LKH-3 and OR-Tools are other classical solver references (Helsgaun, 2017; Perron & Furnon, 2025). Section G gives the detailed literature comparison.

3 Problem Setup

An MTVRP instance has depots, N customers and a shared vehicle capacity Q. Pretraining uses RouteFinder's MTVRP family (Berto et al., 2025a): CVRP combined with subsets of open routes (O), time windows (TW), route-length limits (L) and backhauls (B), giving 16 variants with a total-distance objective. Evaluation adds multi-depot (MD) and mixed-backhaul (MB) constraints (Section A). Heterogeneous CVRP (HCVRP) has K_{fleet} vehicles with capacities Q_k, speeds v_k and unlimited depot refills (Li et al., 2022). For total distance L_k over vehicle k's trips, its objectives are min-sum travel time $\sum_k L_k/v_k$ and min-max travel time $\max_k L_k/v_k$.

A route slot is a route under construction, not a vehicle: the policy builds up to K_{slot} routes concurrently, with active set $\mathcal{A}_t$ at step t. A slot starts another route after a depot return, so the realized route count R is an output of the rollout and can exceed the slot cap. In HCVRP, $K_{\text{slot}} = K_{\text{fleet}}$ and all slots are active; refills reset capacity but preserve accumulated travel cost. S counts candidate rollouts.

Weight-frozen, task-specified transfer updates no parameters and uses target features and feasibility rules supplied by the environment. *Full fine-tuning* initializes from source weights and updates all parameters with a fresh optimizer. *Projection-only fine-tuning* updates input projections while keeping the encoder, communication block and pointer fixed. HCVRP uses the latter two regimes with separate models for each objective; objective switching without updates is not tested.

4 Multi-Agent Route Construction

The shared state s_t contains static instance tokens, the visited set and a route state per slot: current node, load, elapsed route time and route length. An action selects one active slot $k \in \mathcal{A}_t$ and a feasible target v, either a customer or its own depot. Executing this pair (k, v) advances that route and recomputes feasibility. New constraints and vehicle information enter through token attributes, route-state features and masks (Figure 1).

4.1 Encoder and route queries

Each slot is a token at its depot, following Equity-Transformer (Son et al., 2024) and PARCO (Berto et al., 2025b), and carries depot attributes, capacity, slot index and an active flag; customer tokens carry coordinates, demands and service constraints. The four constraint flags are projected to a prompt, as in CaDA (Li et al., 2025); we concatenate this prompt to every token and fuse it linearly. A Transformer encoder produces token embeddings $\mathbf{H}$ once per instance; pointer keys and values are cached throughout construction. Slot tokens above the reset active count are excluded as encoder attention keys (Section B.1).

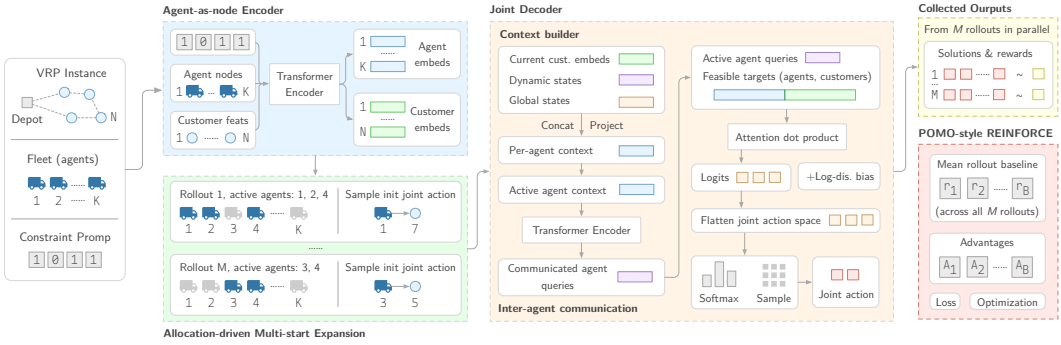


Figure 1: Multi-agent route construction during pretraining at one source size. Static tokens are encoded once; each step coordinates current route states and extends one route–target pair. Rollouts share a REINFORCE baseline.

For slot k at node pos_k, the dynamic route query combines its current-node embedding with its persistent slot token a_k:

$$c_k = W_c\Big[h^{\mathrm{n}}_{\mathrm{pos}_k} \,\|\, h^{\mathrm{n}}_{a_k} \,\|\, W_u u_k \,\|\, W_g \rho_t\Big], \tag{1}$$

where h^{n}_v is row v of $\mathbf{H}$, u_k contains available load, elapsed route time and remaining route distance, and ρ_t is the fraction of customers served. Vehicle attributes can be appended to the slot features and u_k without changing the construction operations.

4.2 Joint route–customer decoding

Route queries pass through one PARCO communication block (Berto et al., 2025b) and RMS normalization to give q_k. This attention exchanges current route states within a rollout; its mask separates rollouts but does not remove inactive slot rows. Each query then attends to cached keys and values of its feasible targets, yielding a glimpse g_k as in the attention model (Kool et al., 2019).

For target v with logit key h^{ℓ}_v and coordinates x_v, the score is

$$s_{k,v} = \frac{\langle g_k, h^{\ell}_v \rangle}{\sqrt{d}} - \log \|x_{\mathrm{pos}_k} - x_v\|. \tag{2}$$

The distance term is a non-learned prior with no gradient; non-finite logarithms are replaced by zero. Scores are flattened over pairs, clipped with $C\tanh(s_{k,v})$, $C = 10$ (Bello et al., 2016), masked to the feasible set $\mathcal{F}_t$ and normalized once:

$$\pi_\theta\big((k,v) \mid s_t\big) = \frac{\exp\big(C\tanh(s_{k,v})\big)}{\sum_{(k',v')\in\mathcal{F}_t} \exp\big(C\tanh(s_{k',v'})\big)}, \qquad (k,v) \in \mathcal{F}_t. \tag{3}$$

The prior enters before clipping and does not determine feasibility.

One pair is sampled or selected by argmax, and only its slot moves. An own-depot return resets trip resources while preserving accumulated distance; state and feasibility are then updated. The final mask encodes variant rules, own-depot returns and the rollout's active set, so inactive rows that pass through communication are never selected. The encoder mask is fixed at reset, whereas glimpse and final pair masks restrict the current feasible targets; their timing is specified in Section B.1. With one active slot, the distribution reduces to a single-route pointer (Section A).

4.3 Training with allocation-driven multi-start

We train separate $N = 50$ and $N = 100$ source checkpoints on equally sampled MTVRP variants. For total linehaul demand D^{L} and capacity Q, a rollout activates $\lceil D^{\mathrm{L}}/(cQ)\rceil$ slots, clamped between the number of depots and $K_{\mathrm{slot}} = 8$. The multiplier c changes the construction allocation, while feasibility and cost use the original Q. Each instance has $S = \min(50, N)$ rollouts sharing encoder outputs and parameters. They independently draw $c \sim \mathcal{U}[1.2, 2.0]$ in training, with evenly spaced values at evaluation. Their initial pairs are sampled from Equation (3), rather than forced to designated

customers as in POMO (Kwon et al., 2020). These mechanisms vary customer-to-slot assignments but do not guarantee distinct allocations.

Training uses REINFORCE with a shared per-instance rollout baseline. Each negative-cost reward is divided by the absolute exponential moving mean of its variant's reward; subtracting the mean normalized reward over the instance's S rollouts gives the advantage. Communication operates within each rollout, while this loss compares rollouts. Section B.4 gives mask timing, optimizer and checkpoint details.

4.4 Transfer and fine-tuning

Frozen transfer keeps all weights. MD inputs assign slots to depots in round-robin order and restrict returns to the assigned depot; MB switches strict backhaul precedence to the mixed rule, and size changes alter the number of customer tokens (Section A).

HCVRP keeps the encoder, communication, feasible glimpse, pair scoring and single-pair execution, with physical vehicles as active slots. Per-vehicle capacity and speed enter feasibility and time updates. Following RouteFinder's Efficient Adapter Layers (Berto et al., 2025a), zero-initialized projection columns add v_k to the slot token and $(v_k, L_k/v_k, \|x_{\text{pos}_k} - x_{\text{depot}}\|/v_k)$ to u_k. Cumulative travel time survives refills, unlike elapsed trip time; time to depot is recomputed at each step and included in final completion time. Zero initialization preserves the source projection mapping for the existing features; changed fleets and masks can still produce different rollouts.

Adaptation loads policy weights with a fresh optimizer and uses separate min-sum and min-max rewards. With fleet size supplied by the instance, the allocation multiplier is disabled and diversity comes from $S = 50$ sampled rollouts. Full fine-tuning updates all parameters; projection-only fine-tuning updates existing and new input projections while keeping encoder blocks, communication and pointer parameters fixed (Section B.5).

5 Experiments

We evaluate frozen reuse and adaptation quality, then separate the value of pretrained initialization from the effects of construction choices.

5.1 Setup

The $N{=}50/100$ sources are evaluated on 16 ID and 32 unseen variants, with 1,000 instances per cell, and on 65 CVRPLib X instances (Uchoa et al., 2017) ranging from 100 to 1,000 customers. Frozen comparisons use MTPOMO (Liu et al., 2024b), MVMoE (Zhou et al., 2024), RF-TE (Berto et al., 2025a), CaDA (Li et al., 2025) and CCL (Gui et al., 2026) on ID, and the latter three on unseen variants and CVRPLib. All use the same test files, and CCL's ID values are those reported by its authors on these files; identical baseline MD/MB features and masks were not verified, so these are system-level comparisons under supplied task rules. HCVRP evaluates the Li fixed-fleet min-sum cells (Li et al., 2022) and the $N{=}60$ min-max instances shared by 2D-Ptr, PARCO and ECHO (Liu et al., 2024c; Berto et al., 2025b; Wu et al., 2026b), with 1,280 instances per cell. For HCVRP, RF-TE and CCL start from their official $N{=}100$ checkpoints, restored strictly and verified by hash, and keep their sequential vehicle-token construction. Each receives the same per-vehicle capacity, speed, cumulative time and time to depot as ours through zero-initialized input columns, and all parameters are trained (Section B.6). An independent checker replays every candidate of the fresh runs on the raw instance (1,280/1,280 complete, zero invalid).

ID results use 8 augmentations $\times$ 30 (N50) or 50 (N100) starts for our model (240/400 candidates) and 400/800 candidates for the baselines we evaluated; unseen, CVRPLib and design comparisons use 8 $\times$ 50 (400 candidates), and HCVRP uses 1,280 candidates. Gaps are mean percentage gaps to PyVRP/HGS (Wouda et al., 2024; Vidal, 2022), or to best-known solutions for CVRPLib; HCVRP reports mean travel-time cost. Table 3 and the cross-system budget curves use one downstream seed, whereas Figure 2 uses three, and each design configuration is one run; none replicates pretraining seeds.

Table 1: Frozen-weight reuse at N=100: gap (%) to HGS; lowest per row in bold. CCL's in-distribution values are as reported by its authors. N=50 and further baselines are in Section C.

Variant	RF-TE	CaDA	CCL	*Ours*	Variant	RF-TE	CaDA	CCL	*Ours*
In-distribution (16)					*Unseen, multi-depot (24)*				
CVRP	1.52	1.38	1.25	**1.23**	MDCVRP	67.11	75.39	41.68	**14.16**
VRPTW	3.15	2.94	1.93	**1.43**	MDOVRP	34.37	33.39	26.72	**15.88**
OVRP	4.00	3.81	3.53	**3.48**	MDVRPB	61.76	68.60	56.98	**17.55**
OVRPTW	2.35	2.11	1.16	**0.92**	MDVRPL	70.50	77.34	47.70	**14.34**
OVRPB	4.23	3.91	3.20	**3.19**	MDOVRPB	35.66	36.14	26.27	**16.36**
OVRPBTW	2.06	1.77	1.00	**0.86**	MDOVRPL	34.62	32.99	29.03	**15.73**
OVRPBL	4.27	3.93	3.35	**3.15**	MDVRPBL	68.90	70.88	58.52	**18.22**
OVRPBLTW	2.06	1.78	0.99	**0.88**	MDVRPMB	76.33	83.82	79.15	**28.66**
OVRPL	4.05	3.80	3.53	**3.48**	MDVRPTW	53.28	34.78	39.56	**14.80**
OVRPLTW	2.35	2.09	1.16	**0.92**	MDOVRPBL	35.77	36.11	26.55	**16.00**
VRPB	3.91	3.71	**3.51**	3.53	MDOVRPMB	53.65	53.01	42.04	**31.50**
VRPBTW	2.64	2.40	1.51	**1.26**	MDOVRPTW	41.36	32.80	31.20	**14.59**
VRPBL	5.01	4.81	**4.25**	4.46	MDVRPBTW	39.93	37.56	39.76	**18.29**
VRPBLTW	5.79	5.49	4.62	**4.30**	MDVRPLTW	53.98	35.86	45.06	**14.92**
VRPL	1.86	1.71	**1.54**	1.56	MDVRPMBL	80.89	83.44	73.65	**28.71**
VRPLTW	3.65	3.39	2.37	**1.80**	MDOVRPBTW	36.90	34.92	30.96	**17.44**
Mean	3.31	3.06	2.43	**2.28**	MDOVRPLTW	40.82	32.92	29.86	**14.24**
Unseen, single depot (8)					MDOVRPMBL	53.69	52.73	41.42	**31.23**
VRPMB	10.19	**8.82**	13.44	9.31	MDVRPBLTW	41.41	39.03	39.54	**18.14**
OVRPMB	18.90	**16.45**	19.64	16.73	MDVRPMBTW	54.40	53.00	47.66	**25.43**
VRPMBL	10.38	**9.01**	14.46	9.85	MDOVRPBLTW	36.63	35.15	31.60	**17.44**
OVRPMBL	18.88	**16.40**	20.75	16.63	MDOVRPMBTW	46.22	45.02	37.81	**22.69**
VRPMBTW	10.84	11.07	10.24	**6.40**	MDVRPMBLTW	55.13	53.99	50.70	**25.26**
OVRPMBTW	8.72	8.90	8.59	**5.27**	MDOVRPMBLTW	45.87	45.29	37.67	**22.71**
VRPMBLTW	10.81	11.00	9.91	**6.33**	*Mean*	50.80	49.34	42.13	**19.76**
OVRPMBLTW	8.74	8.89	8.44	**5.24**	*Mean, all 32 unseen*	41.14	39.83	34.89	**17.19**
Mean	12.18	11.32	13.18	**9.47**					

5.2 Frozen-weight transfer to unseen variants and CVRPLib

On the 16 ID variants at N=100 (Table 1), ours has the lowest mean gap (2.28% against 2.43% for CCL) and the lowest gap on 13 variants; CCL is lower on VRPB, VRPBL and VRPL.

With frozen weights and supplied MD/MB features and masks, the mean gap on the 32 unseen variants is 17.19% against 34.89% for CCL. Most of the margin lies in the 24 multi-depot variants (19.76% against 42.13%), on each of which ours has the lowest gap; on the eight single-depot MB variants the mean is 9.47% against 11.32% for CaDA, which is lower on VRPMB, OVRPMB, VRPMBL and OVRPMBL. Our gap is lower than CCL's on 32/32 variants at both sizes and the lowest of all compared systems on 28/32 (families in Table 10). Section 5.5 relates part of this margin to the geometric prior.

CVRPLib X tests scale and distribution shift together (Table 2). The frozen model has the lowest mean gap overall (8.29% against 8.60% for RF-TE) and is lowest on 37/65 instances. On the 32 instances with $N \geq 500$, at least five times the largest pretraining size, its mean gap is 11.76% against 12.23% for RF-TE and 24.11% for CCL, lowest on 22; per-instance results are in Section C.

Table 2: Frozen reuse on CVRPLib-X (Uchoa et al., 2017): mean gap (%) to best-known solutions by instance size, pretraining at $N \leq 100$; lowest in bold.

Method	$N \leq 250$ (33)	$N \geq 500$ (32)	All (65)
RF-TE	5.07	12.23	8.60
CaDA	5.26	13.62	9.37
CCL	**4.89**	24.11	14.36
Ours	4.91	**11.76**	**8.29**

5.3 FINE-TUNING TO HETEROGENEOUS FLEETS AND OBJECTIVES

Table 3: Objective-specific HCVRP adaptation after 1M main-run target exposures with 1,280 candidates per instance. Each cost is the mean over the test set; the lowest cost among the adapted models is in bold.

	Min-sum HCVRP (Li et al., 2022)				Min-max HCVRP (Berto et al., 2025b)		
Method	N40/M3	N60/M3	N80/M3	N80/M5	N60/M3	N60/M5	N60/M7
Exact†	55.43	78.47	102.42	102.42	–	–	–
SISR†	55.79	79.12	103.41	103.49	6.57	4.00	2.91
DRL-Li†	57.05	80.46	105.29	105.54	–	–	–
PARCO†	–	–	–	–	6.82	4.17	3.06
ECHO†	–	–	–	–	6.63	4.05	2.93
2D-Ptr†	–	–	–	–	6.82	4.20	3.09
RF-TE	57.02	80.38	105.42	105.54	6.79	4.16	3.05
CCL	**56.61**	80.05	105.40	105.49	6.78	4.15	3.06
Ours	56.63	**79.97**	**104.84**	**104.92**	**6.72**	**4.11**	**3.01**

Min-max values of *Ours* are Ours*, the seed-1234 full fine-tuning run of Section 5.4 (N=50 source), reused without further training; HPO and recipe differ (Section D.1). Other adapted models use N=100 sources. †Published values; – no published value on this benchmark.

HCVRP adaptation adds vehicle capacity, speed and travel-time state and updates all parameters with a fresh optimizer, separately for min-sum and min-max (Section 4.4). Table 3 aligns 1M main-run exposures and 1,280 candidates; source training and hyperparameter optimization (HPO) are separate. The Li systems adapt from N=100 sources on the N=60 random-fleet training distribution without Li-specific training. Each selects its learning rate on validation from three values, with a planned 100K training samples per value. Our single-seed mean cost is below RF-TE in 4/4 Li cells (0.51–0.69%) and CCL in 3/4 (0.09–0.54%); CCL is lower at N40/M3 by 0.03%.

For PARCO min-max, Ours* reuses the seed-1234 1M run from Section 5.4 without further training or HPO. It has lower mean cost than RF-TE and CCL in all three fleets, by 0.91–1.39% and 0.88–1.63%, respectively. Its N=50 source, fixed learning rate, precision and sampling differ from the baselines' (Section D.1). This is a budget-aligned system comparison; the differences are small. Published SISR and ECHO means are lower in all three cells, under unmatched training and unpaired evaluation protocols documented in Section D.1.

Rankings depend on the target budget (Figure 3 in Section D). For min-sum on random fleets, our cost is below RF-TE in all 12 budget–fleet cells and below CCL in 10/12, with CCL lower for three vehicles at 100K and 200K. For min-max, RF-TE is below reused Ours* in all three fleets at 100K (6.81 against 6.82 for three vehicles); from 200K on, Ours* is below both baselines in every fleet, and it is below CCL at every budget. To distinguish adaptation quality from the value of pretraining, we next fix our architecture and recipe and vary initialization and update scope.

5.4 PRETRAINED VERSUS SCRATCH INITIALIZATION

All three configurations train on N=60 random-fleet min-max instances with the same data stream and a fixed learning rate of 10^{-4}, so the 100K, 200K, 500K and 1M budgets are epochs 1, 2, 5 and 10 (Section B.5); scratch is not independently tuned. These budgets exclude source training; single 5M runs from an earlier study are reported separately (Sections D.4 and D.5). A has lower mean cost than B in all 12 budget–fleet cells. At 200K exposures, A reaches mean costs of 6.77, 4.14 and 3.04 for 3, 5 and 7 vehicles, below B's 1M costs of 6.93, 4.24 and 3.10 in each fleet (Figure 2). C lies between A and B at every evaluated budget and stays above A at 1M (6.79, 4.16 and 3.07 against 6.72, 4.11 and 3.01), showing a quality trade-off when updates are restricted to projections. This comparison supports pretrained initialization within our architecture and fixed recipe; it does not measure benefits across backbones, total training compute or wall-clock savings.

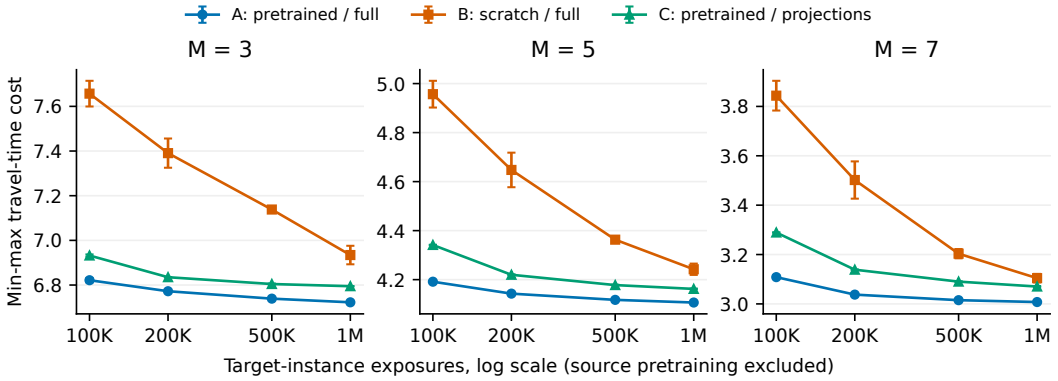


Figure 2: Adaptation from $N{=}50$ to $N{=}60$ min-max HCVRP (A: pretrained/full fine-tuning; B: scratch/full; C: pretrained/projections only); error bars are sample SD over three downstream seeds, 1,280 candidates.

Table 4: Construction configurations at $N{=}50$ with 400 candidates. Mean gap to PyVRP (%) over ID and unseen variants; each configuration is one training run. SDs and per-variant results are in Section E.1.

Configuration	ID16	Unseen32	MD24	SD-MB8
Full (ours)	1.327	11.740	13.100	7.662
2D-Ptr-style (our impl.)	1.636	20.232	24.367	7.825
POMO-style forced multi-start	1.555	12.025	13.612	7.266
− distance prior	1.437	17.987	21.454	7.585
− agent/virtual-depot token	1.430	13.578	15.526	7.736
− communication	1.534	11.564	12.569	8.552

5.5 Ablation Study

Table 4 compares construction configurations at $N{=}50$: Full is our $N{=}50$ source checkpoint, each other configuration is trained once with the same recipe, and all are evaluated on the 16 ID variants and on the 32 unseen variants, split into 24 multi-depot (MD) and 8 single-depot mixed-backhaul (SD-MB) variants. The 2D-Ptr-style pointer, implemented in our framework, keeps route states and the pair action but removes agent tokens, communication, the prior, feasibility-aware glimpse and allocation multi-start (Section B.2); it is not a reproduction of 2D-Ptr (Liu et al., 2024c). The POMO-style configuration replaces allocation-driven rollouts by distinct forced initial customers (Kwon et al., 2020), and the remaining configurations remove the prior, agent tokens or communication. All configurations keep the pair action, so the comparison is not between single-agent and multi-agent construction.

Distance prior and agent tokens. The pointer alone retains most ID quality (1.64% against 1.33% for Full, worse on 16/16 variants) but not transfer: its unseen gap is 20.2% against 11.7% (worse on 28/32), and the difference lies in the MD variants (24.4% against 13.1%, 24/24), while SD-MB gaps are close (7.82% against 7.66%). Removing the prior or the agent tokens raises the MD gap by 8.35 and 2.43 pp (24/24 and 16/24 variants worse) and changes SD-MB by less than 0.1 pp. In an earlier evaluation, removing the prior raised the unseen gap by 5.10 pp, about 37% of the 13.6 pp margin over CCL in that evaluation (Section E.3).

Communication. Communication lowers the gap on every single-depot variant: removing it raises the ID gap by 0.21 pp (16/16) and the SD-MB gap by 0.89 pp (8/8). On MD variants the sign reverses, and removing it lowers the gap by 0.53 pp (worse on 6/24), so its average over all unseen variants is near zero. These marginal removals are not additive.

Allocation-driven multi-start. Replacing allocation-driven rollouts by POMO-style forced starts raises the ID gap from 1.33% to 1.55% (worse on 16/16 variants) and the unseen gap from 11.7% to 12.0% (22/32), mainly on MD variants (18/24), while SD-MB gaps are lower (7.27% against 7.66%). In an earlier evaluation on the ID variants, a single greedy rollout of the allocation-trained model

reached a 4.100% gap against 4.426% for the forced-start model, and its rollouts differed more in customer-to-route partition (0.683 against 0.623, higher on 16/16 variants; Section E.5).

The design also has a cost: configurations without communication took 176–179 s versus Full's 335 s over the 48 variants, and separate $N{=}100$ measurements give 3.63 GiB peak memory versus 0.94 GiB for the lightest baseline; no time- or memory-matched configuration was evaluated (Section F).

6 Discussion and Limitations

The clearest effect of the pretrained construction is under unseen constraints: with frozen weights and supplied features and masks, its gap is the lowest among the compared systems on 28/32 variants at both sizes, whereas ID quality is comparable to the strongest baseline and single-seed adaptation differences are small and change rank with the target budget. Frozen transfer relies on supplied feasibility and the geometric prior (Table 4) as well as on learned representations.

Isolating the construction itself would require a sequential-construction configuration under the same recipe and the same construction inside other foundation backbones; neither was run, so the cross-system comparisons remain system-level. Adaptation comparisons align main-run exposures and candidates while differing in source training, HPO and, for min-max, recipe. Their single downstream seed and the three-seed ABC study have distinct statistical scopes, and the pretraining seed was replicated only for the frozen-transfer means.

The evidence covers constraint families within the six-attribute alphabet, with frozen transfer beyond $N{=}100$ tested only on CVRPLib up to $N{=}1{,}000$; pickup-and-delivery, split delivery, stochastic demand and non-Euclidean costs were not tested. Fleet adaptation covers 3 to 7 vehicles at $N \leq 100$, inside the training fleet range, and each objective uses a separately adapted model. Because the two objectives also change the fleet, no objective-only effect is identified, and crossed combinations of new constraints, fleets and objectives were not tested. Larger slot caps degrade quality beyond 20 in the separate $N{=}50$ sweep, and equal candidate counts do not equalize compute (Section F).

7 Conclusion

Multi-agent route construction provides a shared interface for pretraining vehicle and partial-route decisions across VRP variants. The resulting backbones support frozen reuse under supplied feasibility and objective-specific adaptation: at 1M main-run exposures, costs are lower than both adapted multi-task baselines in all min-max cells and in three of four min-sum cells. Within our architecture and recipe, pretrained fine-tuning at 200K exposures has lower three-seed mean costs than scratch training at 1M. These results establish transfer and fine-tuning gains of this construction in the tested settings.

AI use statement

Large language models were used to aid and polish the writing of this paper. The authors reviewed all text, claims, numbers, and citations and take full responsibility for the content.

REFERENCES

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

Federico Berto, Chuanbo Hua, Nayeli Gast Zepeda, André Hottung, Niels Wouda, Leon Lan, Junyoung Park, Kevin Tierney, and Jinkyoo Park. RouteFinder: Towards foundation models for vehicle routing problems. *Transactions on Machine Learning Research*, 2025a. URL https://arxiv.org/abs/2406.15007.

Federico Berto, Chuanbo Hua, Laurin Luttmann, Jiwoo Son, Junyoung Park, Kyuree Ahn, Changhyun Kwon, Lin Xie, and Jinkyoo Park. PARCO: Parallel autoregressive models for multi-agent combinatorial optimization. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2409.03811.

Federico Berto, Chuanbo Hua, Junyoung Park, Laurin Luttmann, Yining Ma, Fanchen Bu, Jiarui Wang, Haoran Ye, Minsu Kim, Sanghyeok Choi, Nayeli Gast Zepeda, André Hottung, Jianan Zhou, Jieyi Bi, Yu Hu, Fei Liu, Hyeonah Kim, Jiwoo Son, Haeyeon Kim, Davide Angioni, Wouter Kool, Zhiguang Cao, Qingfu Zhang, Joungho Kim, Jie Zhang, Kijung Shin, Cathy Wu, Sungsoo Ahn, Guojie Song, Changhyun Kwon, Kevin Tierney, Lin Xie, and Jinkyoo Park. RL4CO: An extensive reinforcement learning for combinatorial optimization benchmark. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5278–5289, 2025c. doi: 10.1145/3711896.3737433.

Guillaume Bono, Jilles S. Dibangoye, Olivier Simonin, Laëtitia Matignon, and Florian Pereyron. Solving multi-agent routing problems using deep attention mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7804–7813, 2021. doi: 10.1109/TITS.2020.3009289.

Yuhong Cao, Zhanhong Sun, and Guillaume Sartoretti. DAN: Decentralized attention-based neural network for the MinMax multiple traveling salesman problem. In *Distributed Autonomous Robotic Systems (DARS 2022)*, Springer Proceedings in Advanced Robotics, pp. 202–215. Springer, 2024. doi: 10.1007/978-3-031-51497-5_15.

Yuanyao Chen, Rongsheng Chen, Fu Luo, and Zhenkun Wang. Improving generalization of neural combinatorial optimization for vehicle routing problems via test-time projection learning. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.02392.

Jan Christiaens and Greet Vanden Berghe. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433, 2020. doi: 10.1287/trsc.2019.0914.

Arthur Corrêa, Paulo Nascimento, and Samuel Moniz. FiLMMeD: Feature-wise linear modulation for cross-problem multi-depot vehicle routing. *arXiv preprint arXiv:2604.28102*, 2026a. URL https://arxiv.org/abs/2604.28102.

Arthur Corrêa, Paulo Nascimento, and Samuel Moniz. Improving cross-problem vehicle routing with locally augmented preferences and representation disentanglement. *arXiv preprint arXiv:2608.24859*, 2026b. URL https://arxiv.org/abs/2608.24859.

Arthur Corrêa, Cristóvão Silva, Liming Xu, Alexandra Brintrup, and Samuel Moniz. TuneNSearch: A hybrid transfer learning and local search approach for solving vehicle routing problems. *Computers & Operations Research*, 190:107433, 2026c. doi: 10.1016/j.cor.2026.107433.

Darko Drakulic, Sofia Michel, and Jean-Marc Andreoli. GOAL: A generalist combinatorial optimization agent learner. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.15079.

Jonas K. Falkner and Lars Schmidt-Thieme. Learning to solve vehicle routing problems with time windows through joint attention. *arXiv preprint arXiv:2006.09100*, 2020. URL https://arxiv.org/abs/2006.09100.

Han Fang, Zhihao Song, Paul Weng, and Yutong Ban. INViT: A generalizable routing problem solver with invariant nested view transformer. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12973–12992, 2024. URL https://proceedings.mlr.press/v235/fang24c.html.

Daniel Fuertes, Carlos R. del Blanco, Fernando Jaureguizar, and Narciso García. TOP-Former: A multi-agent transformer approach for the team orienteering problem. *IEEE Transactions on Intelligent Transportation Systems*, 26(9):13799–13810, 2025. doi: 10.1109/TITS.2025.3566157.

Ricardo Gama, Ricardo Cunha, Daniel Fuertes, Carlos R. del Blanco, and Hugo L. Fernandes. Multiagent environments for vehicle routing problems. *INFORMS Journal on Computing*, 2026. doi: 10.1287/ijoc.2025.1211.

Chengrui Gao, Haopu Shang, Ke Xue, Dong Li, and Chao Qian. Towards generalizable neural solvers for vehicle routing problems via ensemble with transferrable local policy. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6914–6922, 2024. doi: 10.24963/ijcai.2024/764.

Yong Liang Goh, Zhiguang Cao, Yining Ma, Jianan Zhou, Mohammed Haroon Dupty, and Wee Sun Lee. SHIELD: Multi-task multi-distribution vehicle routing solver with sparsity and hierarchy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19679–19711, 2025. URL https://proceedings.mlr.press/v267/goh25a.html.

Shuangchun Gui, Suyu Liu, Xuehe Wang, and Zhiguang Cao. Chain-of-context learning: Dynamic constraint understanding for multi-task VRPs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.01667.

Keld Helsgaun. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. Technical report, Roskilde University, 2017. URL https://forskning.ruc.dk/en/publications/an-extension-of-the-lin-kernighan-helsgaun-tsp-solver-for-constra.

André Hottung, Mridul Mahajan, and Kevin Tierney. PolyNet: Learning diverse solution strategies for neural combinatorial optimization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c7b9810b2e27bccdc8d605a3a54c8cd6-Abstract-Conference.html.

Chuanbo Hua, Federico Berto, Jiwoo Son, Seunghyun Kang, Changhyun Kwon, and Jinkyoo Park. CAMP: Collaborative attention model with profiles for vehicle routing problems. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1015–1024. IFAAMAS, 2025. URL https://arxiv.org/abs/2501.02977.

Chuanbo Hua, Federico Berto, Zhikai Zhao, Jiwoo Son, Changhyun Kwon, and Jinkyoo Park. USPR: Learning a unified solver for profiled routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 36973–36981, 2026. doi: 10.1609/aaai.v40i43.41025.

Shihong Huang, Shengjie Wang, Lei Gao, Hong Ma, Zhanluo Zhang, Feng Zhang, and Weihua Zhou. Vehicle-as-prompt: A unified deep reinforcement learning framework for heterogeneous fleet vehicle routing problem. *arXiv preprint arXiv:2604.05195*, 2026. URL https://arxiv.org/abs/2604.05195.

Ziwei Huang, Jianan Zhou, Zhiguang Cao, and Yixin Xu. Rethinking light decoder-based solvers for vehicle routing problems. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/447ac93bf22099aa346a45577376492d-Abstract-Conference.html.

Amine Jari, Sohaib Afifi, Rym Nesrine Guibadj, and Eric Lefèvre. G-UniRouting: A graph-based unified neural model for solving multi-attribute vehicle routing problems. In *2025 IEEE 37th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 491–498, 2025. doi: 10.1109/ICTAI66417.2025.00073.

Minsu Kim, Junyoung Park, and Jinkyoo Park. Sym-NCO: Leveraging symmetricity for neural combinatorial optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1936–1949, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0cddb777d3441326544e21b67f41bdc8-Abstract-Conference.html.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1803.08475.

Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21188–21198, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f231f2107df69eab0a3862d50018a9b2-Abstract.html.

Han Li, Fei Liu, Zhi Zheng, Yu Zhang, and Zhenkun Wang. CaDA: Cross-problem routing solver with constraint-aware dual-attention. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35438–35456, 2025. URL https://proceedings.mlr.press/v267/li25bi.html.

Jingwen Li, Yining Ma, Ruize Gao, Zhiguang Cao, Andrew Lim, Wen Song, and Jie Zhang. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 52(12):13572–13585, 2022. doi: 10.1109/TCYB.2021.3111082.

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, 2024. doi: 10.1109/TITS.2024.3413077.

Zhuoyi Lin, Yaoxin Wu, Bangjian Zhou, Zhiguang Cao, Wen Song, Yingqian Zhang, and Senthilnath Jayavelu. Cross-problem learning for solving vehicle routing problems. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6958–6966, 2024. doi: 10.24963/ijcai.2024/769.

Fei Liu, Xi Lin, Weiduo Liao, Zhenkun Wang, Qingfu Zhang, Xialiang Tong, and Mingxuan Yuan. Prompt learning for generalized vehicle routing. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6976–6984, 2024a. doi: 10.24963/ijcai.2024/771.

Fei Liu, Xi Lin, Zhenkun Wang, Qingfu Zhang, Xialiang Tong, and Mingxuan Yuan. Multi-task learning for routing problem with cross-problem zero-shot generalization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1898–1908, 2024b. doi: 10.1145/3637528.3672040.

Qidong Liu, Chaoyue Liu, Shaoyao Niu, Cheng Long, Jie Zhang, and Mingliang Xu. 2D-Ptr: 2D array pointer network for solving the heterogeneous capacitated vehicle routing problem. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1238–1246. IFAAMAS, 2024c. URL https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p1238.pdf.

Qidong Liu, Jiurui Lian, Chaoyue Liu, and Zhiguang Cao. Enhancing generalization in large-scale HCVRP: A rank-augmented neural solver. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1845–1856, 2025a. doi: 10.1145/3711896.3736935.

Suyu Liu, Zhiguang Cao, Shanshan Feng, and Yew-Soon Ong. A mixed-curvature based pre-training paradigm for multi-task vehicle routing solver. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38066–38101, 2025b. URL https://proceedings.mlr.press/v267/liu25b.html.

Laurin Luttmann and Lin Xie. Multi-action self-improvement for neural combinatorial optimization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.12273.

Dian Meng, Zhiguang Cao, Jie Gao, Yaoxin Wu, and Yaqing Hou. UniteFormer: Unifying node and edge modalities in transformers for vehicle routing problems. In *Advances in Neural Information Processing Systems*, 2025. URL https://neurips.cc/virtual/2025/poster/119374.

Saeed Nasehibasharzad, Farhana Choudhury, Egemen Tanin, and Majid Sarvi. DeepMDV: Global spatial matching for multi-depot vehicle routing problems. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pp. 77–89, 2025. doi: 10.1145/3748636.3762711.

Wenzheng Pan, Hao Xiong, Jiale Ma, Wentao Zhao, Yang Li, and Junchi Yan. UniCO: On unified combinatorial optimization via problem reduction to matrix-encoded general TSP. In *International Conference on Learning Representations*, 2025a. URL https://iclr.cc/virtual/2025/poster/27742.

Yuxin Pan, Zhiguang Cao, Chengyang Gu, Liu Liu, Peilin Zhao, Yize Chen, and Fangzhen Lin. Multi-task vehicle routing solver via mixture of specialized experts under state-decomposable MDP. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2510.21453.

Junyoung Park, Sanjar Bakhtiyar, and Jinkyoo Park. ScheduleNet: Learn to solve multi-agent scheduling problems with reinforcement learning. *arXiv preprint arXiv:2106.03051*, 2021. URL https://arxiv.org/abs/2106.03051.

Junyoung Park, Changhyun Kwon, and Jinkyoo Park. Learn to solve the min-max multiple traveling salesmen problem with reinforcement learning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, pp. 878–886. IFAAMAS, 2023. URL https://www.ifaamas.org/Proceedings/aamas2023/pdfs/p878.pdf.

Nathalie Paul, Tim Wirtz, Stefan Wrobel, and Alexander Kister. Multi-agent neural rewriter for vehicle routing with limited disclosure of costs. *arXiv preprint arXiv:2206.05990*, 2022. URL https://arxiv.org/abs/2206.05990.

Laurent Perron and Vincent Furnon. OR-Tools, 2025. URL https://developers.google.com/optimization/.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1–2):483–523, 2020. doi: 10.1007/s10107-020-01523-z.

Jiwoo Son, Minsu Kim, Sanghyeok Choi, Hyeonah Kim, and Jinkyoo Park. Equity-transformer: Solving NP-hard min-max routing problems as sequential generation with equity context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 20265–20273, 2024. doi: 10.1609/aaai.v38i18.30007.

Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017. doi: 10.1016/j.ejor.2016.08.012.

Thibaut Vidal. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, 140:105643, 2022. doi: 10.1016/j.cor.2021.105643.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/29921001f2f04bd3baee84a12e98098f-Abstract.html.

Pengfu Wan, Jiawei Chen, and Gangyan Xu. Deep reinforcement learning for solving the fleet size and mix vehicle routing problem. *arXiv preprint arXiv:2512.24251*, 2025. URL https://arxiv.org/abs/2512.24251.

Chenguang Wang, Zhang-Hua Fu, Pinyan Lu, and Tianshu Yu. Efficient training of multi-task neural solver for combinatorial optimization. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2305.06361.

Yujun Wang, Xiaopeng Hong, Yabin Wang, Junzhou Zhao, Guanghui Sun, and Baoxing Qin. Token-based deep reinforcement learning for heterogeneous VRP with service time constraints. *Knowledge-Based Systems*, 300:112173, 2024. doi: 10.1016/j.knosys.2024.112173.

Niels A. Wouda, Leon Lan, and Wouter Kool. PyVRP: A high-performance VRP solver package. *INFORMS Journal on Computing*, 36(4):943–955, 2024. doi: 10.1287/ijoc.2023.0055.

Wenjian Wu, Zesheng Jia, Jiaying Tang, Benyuan Yang, and Jin Wang. GeoPAR: Large-scale multi-agent combinatorial optimization with geometry-guided parallel autoregressive learning. *arXiv preprint arXiv:2609.00577*, 2026a. URL https://arxiv.org/abs/2609.00577.

Xuan Wu, Di Wang, Chunguo Wu, Kaifang Qi, Chunyan Miao, Yubin Xiao, Jian Zhang, and You Zhou. Efficient neural combinatorial optimization solver for the min-max heterogeneous capacitated vehicle routing problem. *Expert Systems with Applications*, 331:133234, 2026b. doi: 10.1016/j.eswa.2026.133234. Reported results are taken from arXiv:2507.21386v3.

Zikang Yu, Jinbiao Chen, and Jiahai Wang. Combination-of-experts with knowledge sharing for cross-task vehicle routing problems. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007700.

Ke Zhang, Fang He, Zhengchao Zhang, Xi Lin, and Meng Li. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 121:102861, 2020. doi: 10.1016/j.trc.2020.102861.

Yuepeng Zheng, Fu Luo, Zhenkun Wang, Yaoxin Wu, and Yu Zhou. MTL-KD: Multi-task learning via knowledge distillation for generalizable neural vehicle routing solver. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/899c6a43b9976e1077522fe5a39cafa3-Abstract-Conference.html.

Zhi Zheng, Shunyu Yao, Zhenkun Wang, Xialiang Tong, Mingxuan Yuan, and Ke Tang. DPN: Decoupling partition and navigation for neural solvers of min-max vehicle routing problems. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61559–61592, 2024. URL https://proceedings.mlr.press/v235/zheng24m.html.

Changliang Zhou, Xi Lin, Zhenkun Wang, Xialiang Tong, Mingxuan Yuan, and Qingfu Zhang. Instance-conditioned adaptation for large-scale generalization of neural routing solver. *IEEE Transactions on Intelligent Transportation Systems*, 27:8270–8284, 2026a. doi: 10.1109/TITS.2026.3674538.

Changliang Zhou, Canhong Yu, Shunyu Yao, Xi Lin, Zhenkun Wang, Yu Zhou, and Qingfu Zhang. URS: A unified neural routing solver for cross-problem zero-shot generalization. *arXiv preprint arXiv:2509.23413*, 2026b. URL https://arxiv.org/abs/2509.23413. Version 3; accepted to ICML 2026 according to the arXiv record.

Jianan Zhou, Zhiguang Cao, Yaoxin Wu, Wen Song, Yining Ma, Jie Zhang, and Xu Chi. MVMoE: Multi-task vehicle routing solver with mixture-of-experts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61804–61824, 2024. URL https://proceedings.mlr.press/v235/zhou24c.html.

Zefang Zong, Meng Zheng, Yong Li, and Depeng Jin. MAPDP: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9980–9988, 2022. doi: 10.1609/aaai.v36i9.21236.

Zengyu Zou, Jingyuan Wang, Yixuan Huang, and Junjie Wu. Multi-agent pointer transformer: Seq-to-seq reinforcement learning for multi-vehicle dynamic pickup-delivery problems. *arXiv preprint arXiv:2511.17435*, 2025. URL https://arxiv.org/abs/2511.17435.

A Problem Details and Supplied Feasibility

This appendix records the variant family, the slot accounting, and the feasibility rules that the environment supplies to the policy in every regime of Section 3. All rules below are implemented in the environment, not learned; the policy only scores the extensions the environment marks feasible.

A.1 Variant family

The 16 pretraining variants are the subsets of {O, TW, L, B} applied to CVRP, generated with the RouteFinder MTVRP generator (Berto et al., 2025a) with customers in the unit square. The 32 unseen variants of the design analysis (Table 4) are 24 multi-depot (MD) variants, namely the 16 pretraining variants and the 8 mixed-backhaul (MB) variants each with several depots, plus the 8 single-depot MB variants; the 16 in-distribution variants complete the set of 48. The attribute flags that the encoder receives are computed from the instance itself: open route from the instance flag, TW from a finite depot deadline, L from a finite distance limit, and B from any nonzero backhaul demand. MD and MB have no flag of their own; they enter only through the depot assignment and the backhaul rule below.

A.2 Slots, routes, and termination

Every slot k is materialized as a token at a physical depot; with D depots, slot k is assigned to depot $k \bmod D$, so every depot owns at least one active slot. A slot may return only to its own depot. On a depot return, the slot's load, elapsed route time, and current route length are reset, while its accumulated distance is kept; the slot can then start a new route. A slot that has already served customers and stands at its depot may not select its depot again while feasible customers remain, so an idle slot re-enters construction only through a customer. For slots outside $\mathcal{A}_t$, the environment permits only a zero-cost stay at their own depot; the final policy mask excludes these slots from pair selection. A rollout terminates when all customers are served or after $2N$ steps; in the second case it is truncated. During training a truncated rollout receives a penalty of two per unserved customer on top of its cost; at evaluation a truncated candidate is a failure and is never assigned a finite cost. The reported cost of a completed rollout is the sum over slots of the accumulated distance plus the return distance to the own depot, which is zero for open routes.

A.3 Supplied feasibility rules

For slot k at node pos_k with elapsed time t_k, route length ℓ_k, loads $(u_k^{\mathrm{L}}, u_k^{\mathrm{B}})$, speed v_k, and a candidate customer j with demands $(d_j^{\mathrm{L}}, d_j^{\mathrm{B}})$, window $[e_j, l_j]$, and service time σ_j, the customer is feasible when all active conditions hold:

- *Time windows:* $t_k + \mathrm{dist}(\mathrm{pos}_k, j)/v_k < l_j$, and, unless the route is open, $\max(t_k + \mathrm{dist}(\mathrm{pos}_k, j)/v_k,\, e_j) + \sigma_j + \mathrm{dist}(j, \mathrm{depot}_k)/v_k$ is below the depot deadline.
- *Distance limit:* $\ell_k + \mathrm{dist}(\mathrm{pos}_k, j) + \mathrm{dist}(j, \mathrm{depot}_k) \le L_{\max}$, with the return term dropped for open routes.
- *Capacity:* $u_k^{\mathrm{L}} + d_j^{\mathrm{L}} \le Q_k$ and $u_k^{\mathrm{B}} + d_j^{\mathrm{B}} \le Q_k$, with Q_k per vehicle in HCVRP and shared otherwise.
- *Backhaul with precedence (B):* a linehaul customer is feasible only while unserved linehaul customers remain, the linehaul capacity permits, and the slot's current node is not a backhaul customer; a backhaul customer is feasible when the backhaul capacity permits.
- *Mixed backhaul (MB):* both capacity conditions hold and $d_j^{\mathrm{L}} \le Q_k - u_k^{\mathrm{B}}$; linehaul and backhaul customers may be interleaved.
- *Unvisited:* j has not been served.

The own depot of slot k is feasible when the slot has served at least one customer, when all customers are served, or when no customer is feasible for it; it is masked for a slot that stands at the depot after serving customers while feasible customers remain. The adaptation configurations also set an initial-dispatch flag that masks the depot for slots that have not yet served a customer while feasible customers exist. Open routes set every return distance and return-time term to zero.

A.4 HCVRP DETAILS AND UNITS

In HCVRP the number of slots equals the fleet size given by the instance, all slots are active, and Q_k and v_k are per-vehicle columns that the feasibility kernel and the time update broadcast in place of the shared values. Demands and capacities are divided by 41, the exclusive upper bound of the capacity range, so that they lie in the range the source policy was trained on; the time objectives depend only on distances and speeds and are unaffected. The random-fleet generator follows the 2D-Ptr and PARCO setting (Liu et al., 2024c; Berto et al., 2025b): customer and depot positions uniform in the unit square, demands uniform in $\{1, \dots, 9\}$, capacities uniform in $\{20, \dots, 40\}$, speeds uniform in $[0.5, 1.0]$. The Li fixed fleets (Li et al., 2022) use capacities $(20, 25, 30)$ for three vehicles and $(20, 25, 30, 35, 40)$ for five, with min-sum speeds $1/4, 1/5, 1/6$ for three vehicles and $1/4, 1/5, 1/6, 1/7, 1/8$ for five; the converted instances scale these speeds by four so that they fall into the range $[0.5, 1.0]$ of the random-fleet adaptation data, and the reported Li costs are multiplied by four exactly once to restore the original unit. The MTVRP source generator uses a constant speed of one, so source pretraining saw no heterogeneous speeds; the range above is that of HCVRP adaptation, not of pretraining. A depot return refills the vehicle and starts a new trip; the objectives of Section 3 accumulate over all trips of a vehicle. With common speed v, min-sum travel time is total distance divided by v and has the same optimal solutions; min-max still aggregates by a maximum instead of a sum. With heterogeneous speeds neither time objective is generally equivalent to total distance.

A.5 SINGLE-SLOT REDUCTION

Remark 1 (Single active slot)**.** *Fix a step t with $\mathcal{A}_t = \{k\}$. Then $\mathcal{F}_t = \{k\} \times \mathcal{T}_t$ for the feasible target set $\mathcal{T}_t$ of slot k, and Equation (3) becomes*

$$\pi_\theta(v \mid s_t) = \frac{\exp(C \tanh(s_{k,v}))}{\sum_{v' \in \mathcal{T}_t} \exp(C \tanh(s_{k,v'}))},$$

a single-route pointer over targets with the same glimpse and the same distance prior.

The remark describes one step's distribution and says nothing about the policies obtained by training with one or several active slots.

B ARCHITECTURE, TRAINING, AND ADAPTATION SPECIFICATION

B.1 ARCHITECTURE

All models use $d = 128$ and 8 attention heads. The encoder has six pre-norm Transformer blocks with RMS normalization and a parallel gated MLP with SiLU activation, followed by a final RMS normalization. The constraint prompt is a linear map from the four attribute flags to d, concatenated to every token, layer-normalized, and projected back to d. The route query of Equation (1) concatenates four d-dimensional parts and projects them to d; the route-state projection maps the three state features (six after HCVRP adaptation) to d and the progress feature to d. The communication block is one Transformer block of the same form as the encoder blocks, followed by RMS normalization. The pointer is the attention-model pointer (Kool et al., 2019): an 8-head glimpse over cached keys and values with an output projection, a dot product with cached logit keys scaled by $\sqrt{d}$, and tanh clipping with $C = 10$. No graph context is added to the query. The glimpse keys, glimpse values, and logit keys are one linear map of the encoder output computed once per instance. The adapted HCVRP policy has 2,025,088 parameters; the source policy has 512 fewer, which are the zero-initialized columns added at adaptation.

Slot-token features comprise depot coordinates, open-route flag, distance limit, depot deadline, vehicle capacity, normalized slot index and active flag. Customer features comprise coordinates, linehaul and backhaul demands, time windows and service time. The encoder mask is fixed at reset using the active count computed with $c = 1.2$, before per-rollout multipliers are drawn; each rollout's active set is a prefix of this reset set. Route queries are computed for every slot up to the largest active count in the batch of rollouts. The communication mask separates rollouts but passes inactive rows within a rollout; the final action mask removes every pair of those rows. The glimpse mask restricts

each slot's cached attention keys to its currently feasible targets. The non-learned log-distance term is added to the glimpse-based logits, with non-finite logarithms replaced by zero, before tanh clipping, final feasibility masking and the single pair softmax.

B.2 Design-analysis configurations

Table 5 lists the switches that define the configurations of Table 4. The "feasible" glimpse restricts keys to feasible targets; "unvisited" restricts them to unvisited nodes and physical depots. "Allocation" denotes the per-rollout multiplier c; "on" means $c \sim \mathcal{U}[1.2, 2.0]$ in training and evenly spaced values in $[1.2, 2.0]$ at inference. Every configuration keeps the joint (k, v) action, the environment, the reward, and the training driver; the 2D-Ptr-style configuration keeps the vehicle–customer pointer action of 2D-Ptr and removes the listed components together, and it does not contain 2D-Ptr's inter-vehicle attention. The POMO-style configuration replaces allocation-driven rollouts by distinct forced initial customers, as in POMO (Kwon et al., 2020). Each configuration is one training run with the pretraining recipe of Section B.4. Best checkpoints are selected on validation and lie at different epochs (295, 299, 299, 297, 295, 299 for Full, no prior, no agent token, no communication, 2D-Ptr-style and POMO-style).

Table 5: Switches of the construction configurations. Mask and allocation settings are defined in the text.

Configuration	Encoder and scores	Glimpse and allocation
Full (ours)	Tokens: yes. Comm.: 1 block; prior: yes.	Glimpse: feasible. Allocation: on (train); on (inference).
2D-Ptr-style (our impl.)	Tokens: no; node-only physical-depot encoder. Comm.: none; prior: no.	Glimpse: unvisited. Allocation: off (train); off (inference).
− distance prior	Tokens: yes. Comm.: 1 block; prior: no.	Glimpse: feasible. Allocation: on (train); on (inference).
− agent/virtual-depot token	Tokens: no; slot rows skip attention. Comm.: 1 block; prior: yes.	Glimpse: feasible. Allocation: on (train); on (inference).
− communication	Tokens: yes. Comm.: none; prior: yes.	Glimpse: feasible. Allocation: on (train); on (inference).
POMO-style forced multi-start	Tokens: yes. Comm.: 1 block; prior: yes.	Glimpse: feasible. Allocation: off; forced first customers (train); forced starts, greedy (inference).

In the "− agent/virtual-depot token" configuration the slot rows keep their prompt-fused initial embedding and are excluded from encoder self-attention, so customers never attend to slot tokens; the rows remain valid pointer keys and the action space is unchanged. The 2D-Ptr-style configuration instead encodes the D physical depots and the N customers only and copies each depot's encoding to the slots assigned to it; it requires a shared vehicle capacity.

B.3 Source checkpoints

Two source checkpoints are used, both trained with the recipe below and differing only in problem size and validation sets: the $N = 50$ source (best validation epoch 295) and the $N = 100$ source (best validation epoch 297). Table 6 maps each experiment to its source. No experiment uses a single checkpoint across all settings.

B.4 Pretraining recipe

The encoder runs before allocation multipliers are sampled; only the final feasibility mask is recomputed for each rollout's active count, as detailed in Section B.1. The slot-count calculation uses linehaul demand only and leaves physical capacity unchanged.

Table 6: Source checkpoint of each experiment. Fine-tuned models load policy weights only and start a new optimizer.

Experiment	Source	Update
Frozen reuse: ID, unseen constraints, CVRPLib (Table 1)	$N = 50$ and $N = 100$ sources by evaluation size	none
Design analysis, Full configuration (Table 4)	$N = 50$ source	none (evaluation only)
Pretraining benefit A/C (Figure 2)	$N = 50$ source	full / projection-only, min-max
Pretraining benefit B	random initialization, same architecture	full, min-max
Ours* min-max at 1M (Table 3)	A above, downstream seed 1234	reused without further training
Ours min-sum at 100K–1M (Table 3, Section D)	$N = 100$ source	full, min-sum
RF-TE and CCL, both objectives	official $N = 100$ RF-TE and CCL checkpoints	full, our adapters (Section B.6)

- *Data:* 100,000 freshly generated instances per epoch for 300 epochs, one problem size per source, variants drawn with equal probability from the 16-variant preset, one depot, $K_{\text{slot}} = 8$, seed 1234.
- *Rollouts:* $S = \min(50, N)$ sampled rollouts per instance, no training-time augmentation, per-rollout multiplier $c \sim \mathcal{U}[1.2, 2.0]$, initial pair sampled from the policy, temperature 1.
- *Loss:* REINFORCE with the mean over the S rollouts as baseline; rewards (negative costs) divided by the absolute value of an exponential moving mean of the reward of their variant with rate 0.25 before the baseline is subtracted.
- *Optimizer:* Adam, learning rate 3×10^{-4}, weight decay 10^{-6}, learning rate multiplied by 0.1 after epochs 270 and 295, effective batch 256 instances (391 optimizer steps per epoch).
- *Checkpoint selection:* best mean validation max-reward over the 16 variant validation sets (64 instances each), evaluated with sampling and 8 symmetric augmentations.

B.5 Adaptation recipes and update scopes

Common adaptation changes. Table 7 summarizes what is kept and what changes in each reuse regime. The environment provides per-vehicle capacity and speed, the reward is the min-sum or min-max travel time, and three new input features (speed v_k, cumulative travel time L_k/v_k, and time to depot) enter through four zero-initialized columns: speed on the slot token (a $1 \times d$ map) and all three features in the route state (three columns of the route-state projection). A depot visit refills the vehicle's capacity but does not reset its cumulative travel time. The same added inputs are used by the pretraining-benefit runs and by the Ours min-sum runs of the budget-aligned comparison below. The initial-dispatch flag is set. The allocation multiplier and any forced start are off; rollouts are $S = 50$ samples per instance and the fleet size of a training batch is drawn from $\{3, \ldots, 7\}$ by trimming a seven-vehicle draw. Weights are loaded from the source policy only and the optimizer is new.

Pretraining-benefit study (A/B/C). Target $N = 60$; learning rate 10^{-4} fixed for all configurations; effective batch 256 (64 instances with four accumulation steps on one GPU, fp16 precision); 100,000 instances per epoch, so the budgets 100K, 200K, 500K, and 1M are epochs 1, 2, 5, and 10; three downstream seeds per arm with the training RNG reset after checkpoint loading. A loads the $N = 50$ source and updates all parameters; B trains the same architecture from random initialization; C loads the same source and updates the input projections only. Evaluation uses 1,280 candidates per instance (8 augmentations, 40 starts, 4 repeats).

Budget-aligned comparison with RF-TE and CCL. All conditions share one runner, one replayable data stream (target $N = 60$, fleet drawn per 64-instance block from $\{3, \ldots, 7\}$), bfloat16 precision, two GPUs, effective batch 256, 50 training rollouts per instance, and the budgets 100K, 200K, 500K, and 1M main-run exposures at downstream seed 1234. The learning rate of each

Table 7: What is kept and what changes when the pretrained construction is reused. Weights are updated only in the two fine-tuning regimes.

	Frozen transfer (MD, MB, size)	HCVRP adaptation (min-sum or min-max)
Kept	all operations and weights	encoder, communication, glimpse and score, single-pair step, training driver
Inputs	attribute flags and features of the variant	per-vehicle Q_k, v_k; zero-initialized columns for v_k, L_k/v_k, and time to depot
Masks and state	depot assignment, own-depot return, MB rule	per-vehicle capacity and speed in feasibility and time; depot return refills
Reward	total distance (unchanged)	$\sum_k L_k/v_k$ or $\max_k L_k/v_k$, one model per objective
Rollouts	allocation multiplier on	allocation multiplier off; $S = 50$ sampled rollouts
Updated weights	none	all (full) or input projections only (projection-only)

condition is chosen among $\{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}\}$ by a 100K-exposure pilot per rate on 256 validation instances, an allowance of 300K exposures per condition that is separate from the main run; the selected rates are 3×10^{-4} for Ours min-sum and RF-TE min-sum, 10^{-4} for RF-TE min-max and CCL min-sum, and 3×10^{-4} for CCL min-max. Ours min-max is not part of this runner: Table 3 reuses the seed-1234 A run of the study above, which has a different source size, precision, learning-rate selection, and rollout grouping (Table 15). Evaluation uses 1,280 candidates per instance (8 augmentations, 160 starts). Debugging runs made before a correction of the capacity mask and an unfinished exploratory Ours min-max run were discarded; no weights, validation scores, or learning-rate decisions from them are reused, and their exposures are not part of the allowance above. The Li cells evaluate the same 1M random-fleet checkpoints on the Li test instances without Li-specific training. Details of the learning-rate pilots and the adapted-baseline results are reported by the experiment appendices (Section D).

Update scopes. Full fine-tuning updates every parameter. Projection-only fine-tuning updates the token input projections (depot, customer, capacity, slot index, active flag, speed) and the route-state projections (state features, progress, query fusion), and keeps the constraint prompt fusion, the encoder blocks, the communication block and its normalization, and the pointer projections fixed; this is 68,480 of the 2,025,088 policy parameters, about 3.4%. The projection-only regime updates the pre-existing projection weights as well as the new columns; it is not a new-columns-only experiment.

B.6 Baseline adapters for RF-TE and CCL

The RF-TE and CCL rows of Table 3 are pretrained baselines evaluated with our HCVRP adaptation implementation, not the original authors' HCVRP results. Both baselines construct routes sequentially: the action set holds one token per physical vehicle followed by the customers; selecting a vehicle token closes the active trip, returns that vehicle to the depot, and makes the selected vehicle active, and the last customer closes its trip automatically. Each baseline receives the same per-vehicle information as ours: capacity, speed, cumulative vehicle time, and time to the depot.

- *RF-TE:* the official $N = 100$ RouteFinder Transformer checkpoint (Berto et al., 2025a), restored strictly and verified by hash. Capacity and speed are appended as two zero-initialized columns of the vehicle-token projection; capacity, speed, the active vehicle's cumulative time, and time to depot are appended as four zero-initialized columns of the context projection; a zero-initialized dynamic embedding adds the per-vehicle cumulative time to the glimpse keys, glimpse values, and logit keys. All source tensors are checked to be unchanged after adaptation, and all parameters are trainable.
- *CCL:* the pinned upstream implementation (Gui et al., 2026) with its $N = 100$ checkpoint, restored strictly and verified by hash. Capacity and speed enter through a zero-initialized vehicle-feature projection on the vehicle tokens; remaining capacity, capacity, speed, cumulative time, and time to depot enter through a zero-initialized fleet-context projection added to the query; a zero-initialized workload projection adds the per-vehicle cumulative time to the pointer keys and logits. The trajectory refinement of CCL is applied with probability 0.25 during training and 0.5 at evaluation, as in the upstream code; the initial vehicle of each

rollout is assigned in a balanced order over feasible vehicles with zero log-probability. All parameters are trainable.

Sampling for both baselines uses tanh clipping with $C = 10$ and temperature 1. In the budget-aligned runner an independent checker replays every candidate of the models it evaluates (Ours min-sum, RF-TE, and CCL) on the raw instance, verifies that each customer is served once and that each trip respects the capacity of its vehicle, and assigns an infinite cost to invalid candidates rather than replacing them. The reused Ours* min-max cells (Table 6) are read from the stored per-instance costs of the earlier study; that run recorded neither dataset hashes nor an independent replay, and no retrospective verification of it is claimed.

C In-distribution, unseen-constraint and CVRPLib results

This appendix expands Section 5.2. All experiments here reuse the pretrained weights without updates.

C.1 Datasets, candidates and denominators

The 16 ID variants are the subsets of {O, TW, L, B} on top of the capacitated base problem; the 32 unseen variants activate MD and/or MB in addition, 24 of them with MD and 8 with MB only. Each (variant, size) cell has 1,000 test instances at $N \in \{50, 100\}$. For the ID variants our model decodes 8 augmentations $\times$ 30 starts at N=50 and $\times$ 50 starts at N=100 (240 and 400 candidates), and the baselines we evaluated use 400 and 800 candidates; CCL's ID values are those reported by its authors on the same test files. Unseen-variant and CVRPLib results use 8 augmentations $\times$ 50 starts (400 candidates) for every method. Gaps are relative to PyVRP/HGS (Wouda et al., 2024; Vidal, 2022) and are averaged per cell; family and size summaries are macro-averages over variants, and gaps from different benchmarks are never pooled into one grand average. Best counts compare the displayed neural methods and count ties for each; no aggregate across source sizes is reported.

C.2 In-distribution comparison

Table 9 reports the per-variant ID results of MTPOMO, MVMoE, RF-TE, CaDA, CCL and our model; mean gaps and best counts at both sizes for all six methods are in Table 8, and Table 1 shows the N=100 gaps of four of them. At N=50 CCL has the lowest mean gap (1.355%) and ours the second lowest (1.493%); ours is best on the eight time-window variants, one of them tied with CCL at the displayed precision, and CCL on the eight without. At N=100 ours has the lowest mean gap (2.277% against 2.431% for CCL) and is best on 13 of 16 variants; CCL is best on VRPB, VRPBL and VRPL.

Table 9: In-distribution results on the 16 training variants. * marks the HGS-PyVRP gap reference; the best neural entry per cell is in bold.

Variant	Method	N50			N100		
		Obj.	Gap	Time	Obj.	Gap	Time
CVRP	HGS-PyVRP	10.372	*	10.4m	15.628	*	20.8m
	MTPOMO	10.520	1.423%	2s	15.941	2.030%	8s
	MVMoE	10.499	1.229%	3s	15.888	1.693%	11s
	RF-TE	10.502	1.257%	2s	15.860	1.524%	8s
	CaDA	10.505	1.287%	2s	15.843	1.376%	8s
	CCL	**10.473**	**0.977%**	5s	15.823	1.248%	19s
	Ours	10.492	1.152%	4s	**15.821**	**1.232%**	16s

Continued on next page

Table 9 (continued)

Variant	Method	N50			N100		
		Obj.	Gap	Time	Obj.	Gap	Time
OVRP	HGS-PyVRP	6.507	*	10.4m	9.725	*	20.8m
	MTPOMO	6.717	3.194%	2s	10.216	5.028%	8s
	MVMoE	6.705	3.003%	3s	10.177	4.617%	11s
	RF-TE	6.682	2.658%	2s	10.115	3.996%	8s
	CaDA	6.677	2.585%	1s	10.095	3.805%	8s
	CCL	**6.636**	**1.957%**	5s	10.068	3.527%	20s
	Ours	6.667	2.460%	4s	**10.063**	**3.476%**	17s
OVRPB	HGS-PyVRP	6.898	*	10.4m	10.335	*	20.8m
	MTPOMO	7.105	2.973%	2s	10.882	5.264%	8s
	MVMoE	7.089	2.744%	3s	10.841	4.869%	11s
	RF-TE	7.065	2.385%	2s	10.774	4.233%	8s
	CaDA	7.064	2.377%	1s	10.739	3.909%	8s
	CCL	**7.008**	**1.568%**	5s	10.666	3.203%	19s
	Ours	7.033	1.952%	4s	**10.664**	**3.186%**	19s
OVRPBL	HGS-PyVRP	6.899	*	10.4m	10.335	*	20.8m
	MTPOMO	7.112	3.053%	2s	10.888	5.318%	8s
	MVMoE	7.094	2.799%	3s	10.847	4.929%	11s
	RF-TE	7.068	2.417%	2s	10.778	4.266%	8s
	CaDA	7.062	2.339%	1s	10.741	3.928%	8s
	CCL	**7.009**	**1.569%**	5s	10.681	3.348%	20s
	Ours	7.032	1.930%	4s	**10.661**	**3.153%**	17s
OVRPL	HGS-PyVRP	6.507	*	10.4m	9.724	*	20.8m
	MTPOMO	6.720	3.248%	2s	10.224	5.112%	8s
	MVMoE	6.706	3.028%	3s	10.184	4.693%	11s
	RF-TE	6.683	2.680%	2s	10.121	4.054%	8s
	CaDA	6.680	2.623%	1s	10.093	3.795%	8s
	CCL	**6.637**	**1.968%**	5s	10.067	3.528%	20s
	Ours	6.667	2.464%	4s	**10.062**	**3.477%**	17s
VRPB	HGS-PyVRP	9.687	*	10.4m	14.377	*	20.8m
	MTPOMO	10.036	3.596%	2s	15.102	5.052%	8s
	MVMoE	10.007	3.292%	3s	15.023	4.505%	10s
	RF-TE	9.979	3.000%	2s	14.935	3.906%	8s
	CaDA	9.979	3.010%	1s	14.910	3.707%	8s
	CCL	**9.916**	**2.352%**	5s	**14.882**	**3.513%**	19s
	Ours	9.945	2.659%	4s	14.885	3.533%	16s
VRPBL	HGS-PyVRP	10.186	*	10.4m	14.779	*	20.8m
	MTPOMO	10.679	4.760%	2s	15.718	6.294%	8s
	MVMoE	10.639	4.384%	3s	15.642	5.771%	11s
	RF-TE	10.569	3.713%	2s	15.523	5.008%	8s
	CaDA	10.576	3.776%	1s	15.490	4.811%	8s
	CCL	**10.484**	**2.883%**	5s	**15.407**	**4.249%**	19s
	Ours	10.522	3.299%	4s	15.437	4.455%	16s
VRPL	HGS-PyVRP	10.587	*	10.4m	15.766	*	20.8m
	MTPOMO	10.775	1.733%	2s	16.157	2.483%	8s
	MVMoE	10.753	1.525%	3s	16.099	2.113%	11s
	RF-TE	10.747	1.485%	2s	16.057	1.858%	8s
	CaDA	10.749	1.505%	1s	16.036	1.713%	8s
	CCL	**10.710**	**1.145%**	5s	**16.009**	**1.541%**	19s
	Ours	10.731	1.355%	4s	16.012	1.561%	16s

Continued on next page

Table 9 (continued)

Variant	Method	N50			N100		
		Obj.	Gap	Time	Obj.	Gap	Time
VRPTW	HGS-PyVRP	16.031	*	10.4m	25.423	*	20.8m
	MTPOMO	16.419	2.423%	2s	26.433	3.962%	9s
	MVMoE	16.400	2.298%	3s	26.390	3.789%	11s
	RF-TE	16.341	1.933%	2s	26.228	3.154%	8s
	CaDA	16.312	1.745%	1s	26.169	2.935%	9s
	CCL	16.190	0.979%	5s	25.913	1.928%	21s
	Ours	**16.159**	**0.799%**	4s	**25.787**	**1.431%**	18s
OVRPTW	HGS-PyVRP	10.510	*	10.4m	16.926	*	20.8m
	MTPOMO	10.676	1.558%	2s	17.442	3.022%	9s
	MVMoE	10.674	1.541%	3s	17.416	2.870%	12s
	RF-TE	10.645	1.264%	2s	17.328	2.352%	9s
	CaDA	10.630	1.122%	1s	17.283	2.109%	9s
	CCL	10.569	0.543%	6s	17.123	1.164%	21s
	Ours	**10.557**	**0.447%**	5s	**17.081**	**0.921%**	18s
OVRPBTW	HGS-PyVRP	11.669	*	10.4m	19.156	*	20.8m
	MTPOMO	11.823	1.307%	3s	19.656	2.592%	9s
	MVMoE	11.816	1.245%	4s	19.637	2.499%	13s
	RF-TE	11.790	1.027%	2s	19.555	2.062%	9s
	CaDA	11.775	0.898%	2s	19.495	1.770%	9s
	CCL	11.721	0.436%	6s	19.348	1.002%	21s
	Ours	**11.720**	**0.436%**	4s	**19.321**	**0.862%**	19s
OVRPBLTW	HGS-PyVRP	11.668	*	10.4m	19.156	*	20.8m
	MTPOMO	11.823	1.315%	3s	19.658	2.602%	9s
	MVMoE	11.816	1.249%	4s	19.640	2.514%	12s
	RF-TE	11.789	1.017%	2s	19.554	2.061%	9s
	CaDA	11.777	0.914%	2s	19.497	1.780%	9s
	CCL	11.721	0.442%	6s	19.346	0.992%	22s
	Ours	**11.719**	**0.441%**	4s	**19.323**	**0.875%**	19s
OVRPLTW	HGS-PyVRP	10.510	*	10.4m	16.926	*	20.8m
	MTPOMO	10.677	1.572%	2s	17.442	3.020%	9s
	MVMoE	10.677	1.564%	3s	17.418	2.880%	12s
	RF-TE	10.646	1.267%	2s	17.328	2.352%	9s
	CaDA	10.631	1.133%	1s	17.280	2.092%	9s
	CCL	10.569	0.546%	6s	17.123	1.164%	22s
	Ours	**10.557**	**0.447%**	5s	**17.081**	**0.918%**	18s
VRPBTW	HGS-PyVRP	18.292	*	10.4m	29.467	*	20.8m
	MTPOMO	18.649	1.938%	2s	30.478	3.426%	9s
	MVMoE	18.632	1.841%	3s	30.437	3.284%	12s
	RF-TE	18.573	1.517%	2s	30.249	2.641%	9s
	CaDA	18.543	1.361%	1s	30.174	2.398%	9s
	CCL	18.430	0.738%	6s	29.911	1.506%	21s
	Ours	**18.427**	**0.737%**	4s	**29.839**	**1.262%**	18s
VRPBLTW	HGS-PyVRP	18.361	*	10.4m	29.026	*	20.8m
	MTPOMO	19.001	3.486%	3s	30.948	6.622%	9s
	MVMoE	18.983	3.388%	3s	30.892	6.429%	12s
	RF-TE	18.910	2.990%	2s	30.705	5.785%	9s
	CaDA	18.894	2.902%	1s	30.620	5.491%	9s
	CCL	18.773	2.244%	6s	30.366	4.616%	21s
	Ours	**18.760**	**2.171%**	4s	**30.273**	**4.296%**	18s

Continued on next page

Table 9 (continued)

Variant	Method	N50			N100		
		Obj.	Gap	Time	Obj.	Gap	Time
VRPLTW	HGS-PyVRP	16.356	*	10.4m	25.757	*	20.8m
	MTPOMO	16.832	2.877%	2s	26.913	4.455%	9s
	MVMoE	16.817	2.783%	3s	26.866	4.272%	12s
	RF-TE	16.728	2.248%	2s	26.706	3.645%	9s
	CaDA	16.709	2.130%	1s	26.631	3.394%	9s
	CCL	16.579	1.333%	6s	26.366	2.365%	20s
	Ours	**16.542**	**1.140%**	4s	**26.221**	**1.801%**	18s

Three pretraining seeds give an SD of 0.029 pp on the ID mean gap and 0.310 pp on the unseen mean gap (ranges 0.06 pp and 0.6 pp); the selected checkpoint of each pretraining run is at about 116K steps.

C.3 Unseen constraints

For MD variants the agent/virtual-depot tokens carry the coordinates and time-window features of their depot, depots are assigned to slots round-robin, and the return mask allows each route to close only at its own depot; for MB variants the strict-backhaul precedence mask is relaxed. The four-bit constraint prompt keeps its training vocabulary, and no parameter is updated. Relative to CCL the gap is lower on 32 of 32 variants at both sizes. Relative to the three compared systems (RF-TE, CaDA, CCL) our model has the lowest gap on 28/32 variants at both sizes; the four exceptions are the single-depot MB variants VRPMB, OVRPMB, VRPMBL and OVRPMBL, on which CaDA is lower. The unseen mean gaps and best counts in Tables 1 and 8 are computed from the per-variant tables below, and Table 10 groups the $N{=}100$ variants by attribute family.

The effect of the geometric prior on these variants is reported in Section E.

C.4 CVRPLib X

The 65 X instances (Uchoa et al., 2017) have N between 100 and 1,000, with random, clustered and random-clustered customer placements and central, eccentric and random depots; they are generated benchmarks, and we do not describe them as operational data. Gaps are relative to the best-known solutions distributed with the benchmark. Mean gaps are 8.286% (ours), 8.595% (RF-TE), 9.373% (CaDA) and 14.356% (CCL), and the lowest-gap counts are 37, 9, 8 and 11 of 65. Split by size, our model has the lowest gap on 15 of the 33 instances with $N \leq 250$ and on 22 of the 32 instances with $N \geq 500$ (Table 13); per-instance gaps are in Table 14.

The slot-cap sweep of Section F is a separate $N{=}50$ experiment and is not a measurement on CVRPLib.

Table 8: Frozen-weight reuse at both source sizes. Mean gap (%) to HGS, or to best-known solutions for CVRPLib-X, and the number of lowest-gap variants or instances among the displayed methods (ties count for each).

In-distribution (16 training variants)

Method	N50 gap	# Best	N100 gap	# Best
MTPOMO	2.529%	0/16	4.143%	0/16
MVMoE	2.370%	0/16	3.858%	0/16
RF-TE	2.054%	0/16	3.306%	0/16
CaDA	1.982%	0/16	3.063%	0/16
CCL	1.355%	9/16	2.431%	3/16
Ours	1.493%	8/16	2.277%	13/16

Unseen constraints (32 variants)

Method	N50 gap	# Best	N100 gap	# Best
RF-TE	36.529%	0/32	41.144%	0/32
CaDA	34.866%	4/32	39.834%	4/32
CCL	27.588%	0/32	34.892%	0/32
Ours	13.965%	28/32	17.189%	28/32

CVRPLib-X (65 instances, N from 100 to 1,000)

Method	$N \leq 250$ gap	# Best	$N \geq 500$ gap	# Best
RF-TE	5.068%	6/33	12.232%	3/32
CaDA	5.257%	3/33	13.618%	5/32
CCL	4.892%	9/33	24.115%	2/32
Ours	4.914%	15/33	11.763%	22/32

Table 10: Unseen variants at N=100 grouped by attribute family (8 variants each). Each cell is the mean gap to HGS over the family, computed from Table 12.

Family	RF-TE	CaDA	CCL	Ours
MB only (no MD)	12.18%	11.32%	13.18%	9.47%
MD only (no MB, no TW)	51.09%	53.85%	39.18%	16.03%
MD + TW (no MB)	43.04%	35.38%	35.94%	16.23%
MD + MB	58.27%	58.79%	51.26%	27.02%

Table 11: Per-task results on 32 unseen tasks (N=50). Lowest gap per task in bold; summary rows are computed from the per-task gaps (ties count for each method).

Tasks	RF-TE		CaDA		CCL		Ours	
	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓
VRPMB	9.879	8.861%	**9.781**	**7.749%**	9.943	9.538%	9.832	8.320%
MDCVRP	12.559	56.957%	12.335	54.083%	10.829	34.846%	**9.028**	**12.425%**
MDOVRP	6.876	29.051%	6.825	28.069%	6.413	20.155%	**5.982**	**12.076%**
MDVRPB	12.725	60.956%	12.654	60.054%	11.679	47.450%	**8.940**	**12.868%**
MDVRPL	12.618	57.370%	12.426	54.988%	11.278	40.248%	**9.079**	**12.908%**
OVRPMB	6.949	13.690%	**6.872**	**12.440%**	6.917	13.149%	6.887	12.651%
VRPMBL	10.239	8.050%	**10.163**	**7.215%**	10.376	9.456%	10.216	7.765%
MDOVRPB	7.556	31.861%	7.622	33.009%	6.876	19.787%	**6.393**	**11.375%**
MDOVRPL	6.871	28.946%	6.807	27.746%	6.442	20.716%	**5.984**	**12.126%**
MDVRPBL	12.831	61.175%	12.585	58.011%	11.483	43.846%	**9.055**	**13.434%**
MDVRPMB	12.856	76.544%	12.493	71.550%	11.540	58.118%	**8.913**	**22.124%**
MDVRPTW	17.818	48.941%	16.971	41.739%	16.107	34.403%	**13.733**	**14.595%**
OVRPMBL	6.949	13.686%	**6.871**	**12.423%**	6.932	13.404%	6.882	12.582%
VRPMBTW	17.298	8.074%	17.198	7.434%	16.988	6.099%	**16.838**	**5.161%**
MDOVRPBL	7.550	31.772%	7.606	32.713%	6.858	19.463%	**6.405**	**11.571%**
MDOVRPMB	7.617	47.411%	7.541	45.953%	6.720	29.826%	**6.362**	**22.917%**
MDOVRPTW	10.618	34.976%	10.204	29.610%	9.783	24.150%	**8.940**	**13.457%**
MDVRPBTW	18.591	37.364%	19.025	40.629%	18.425	36.029%	**15.666**	**15.662%**
MDVRPLTW	18.127	51.276%	17.125	42.765%	16.769	39.728%	**13.778**	**14.801%**
MDVRPMBL	12.744	74.112%	12.434	69.825%	11.474	56.409%	**8.949**	**21.992%**
OVRPMBTW	11.132	6.265%	11.087	5.849%	10.966	4.658%	**10.964**	**4.639%**
VRPMBLTW	17.597	7.982%	17.495	7.337%	17.514	7.433%	**17.122**	**5.027%**
MDOVRPBTW	11.399	32.332%	11.190	29.807%	10.774	24.830%	**9.878**	**14.451%**
MDOVRPLTW	10.599	34.731%	10.196	29.501%	9.721	23.331%	**8.947**	**13.506%**
MDOVRPMBL	7.602	47.108%	7.540	45.940%	6.733	30.088%	**6.362**	**22.927%**
MDVRPBLTW	19.048	40.583%	19.243	42.015%	18.544	36.702%	**15.708**	**15.796%**
MDVRPMBTW	17.830	48.485%	18.394	53.345%	17.103	42.314%	**14.562**	**21.170%**
OVRPMBLTW	11.138	6.317%	11.090	5.875%	10.993	4.922%	**10.955**	**4.555%**
MDOVRPBLTW	11.389	32.207%	11.185	29.743%	10.767	24.749%	**9.880**	**14.470%**
MDOVRPMBTW	11.055	40.153%	10.833	37.232%	10.171	28.688%	**9.412**	**19.091%**
MDVRPMBLTW	18.222	51.554%	18.511	54.026%	17.940	49.174%	**14.588**	**21.305%**
MDOVRPMBLTW	11.054	40.136%	10.818	37.044%	10.203	29.114%	**9.413**	**19.118%**
Avg. Gap	36.529%		34.866%		27.588%		13.965%	
# Best (Best/Total)	0/32		4/32		0/32		28/32	

Table 12: Per-task results on 32 unseen tasks (N=100). Lowest gap per task in bold; summary rows are computed from the per-task gaps (ties count for each method).

Tasks	RF-TE		CaDA		CCL		Ours	
	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓
VRPMB	14.888	10.189%	**14.710**	**8.822%**	15.322	13.438%	14.764	9.306%
MDCVRP	19.684	67.107%	20.628	75.392%	16.769	41.675%	**13.511**	**14.156%**
MDOVRP	10.683	34.368%	10.605	33.387%	10.095	26.715%	**9.233**	**15.880%**
MDVRPB	18.721	61.761%	19.494	68.604%	18.185	56.977%	**13.620**	**17.554%**
MDVRPL	20.100	70.498%	20.867	77.336%	17.494	47.699%	**13.544**	**14.340%**
OVRPMB	10.711	18.899%	**10.490**	**16.449%**	10.780	19.642%	10.517	16.725%
VRPMBL	15.198	10.375%	**15.016**	**9.011%**	15.761	14.464%	15.125	9.849%
MDOVRPB	11.752	35.659%	11.790	36.137%	10.959	26.274%	**10.082**	**16.359%**
MDOVRPL	10.703	34.620%	10.574	32.985%	10.272	29.033%	**9.206**	**15.730%**
MDVRPBL	19.606	68.898%	19.827	70.875%	18.446	58.520%	**13.759**	**18.224%**
MDVRPMB	18.698	76.325%	19.458	83.817%	18.997	79.149%	**13.644**	**28.657%**
MDVRPTW	29.468	53.283%	25.955	34.783%	26.901	39.557%	**22.134**	**14.798%**
OVRPMBL	10.709	18.877%	**10.486**	**16.399%**	10.883	20.748%	10.511	16.625%
VRPMBTW	28.256	10.840%	28.317	11.074%	28.100	10.239%	**27.122**	**6.401%**
MDOVRPBL	11.761	35.771%	11.788	36.106%	10.983	26.545%	**10.065**	**16.004%**
MDOVRPMB	11.748	53.650%	11.692	53.010%	10.876	42.044%	**10.069**	**31.500%**
MDOVRPTW	18.299	41.356%	17.205	32.804%	17.009	31.200%	**14.833**	**14.590%**
MDVRPBTW	30.681	39.926%	30.176	37.559%	30.677	39.759%	**25.947**	**18.293%**
MDVRPLTW	29.640	53.976%	26.200	35.862%	27.977	45.055%	**22.154**	**14.924%**
MDVRPMBL	19.216	80.892%	19.460	83.444%	18.479	73.647%	**13.696**	**28.711%**
OVRPMBTW	18.449	8.724%	18.478	8.901%	18.427	8.590%	**17.863**	**5.266%**
VRPMBLTW	28.604	10.805%	28.658	11.002%	28.374	9.907%	**27.451**	**6.331%**
MDOVRPBTW	19.590	36.897%	19.305	34.918%	18.764	30.960%	**16.828**	**17.441%**
MDOVRPLTW	18.232	40.817%	17.221	32.923%	16.838	29.857%	**14.816**	**14.241%**
MDOVRPMBL	11.751	53.691%	11.670	52.726%	10.830	41.416%	**10.050**	**31.229%**
MDVRPBLTW	31.044	41.408%	30.537	39.033%	30.667	39.535%	**25.970**	**18.139%**
MDVRPMBTW	29.650	54.395%	29.383	53.001%	28.388	47.661%	**24.112**	**25.434%**
OVRPMBLTW	18.452	8.739%	18.476	8.887%	18.404	8.443%	**17.861**	**5.244%**
MDOVRPBLTW	19.553	36.632%	19.341	35.154%	18.853	31.599%	**16.828**	**17.440%**
MDOVRPMBTW	18.888	46.216%	18.735	45.019%	17.823	37.814%	**15.869**	**22.690%**
MDVRPMBLTW	29.825	55.132%	29.611	53.989%	29.000	50.703%	**24.102**	**25.256%**
MDOVRPMBLTW	18.846	45.873%	18.770	45.286%	17.803	37.665%	**15.870**	**22.710%**
Avg. Gap	41.144%		39.834%		34.892%		17.189%	
# Best (Best/Total)	0/32		4/32		0/32		28/32	

Table 13: CVRPLib-X best counts computed from the per-instance gaps.

Subset	Best / total	Scope
All	37/65	Reported instances
$N \leq 250$	15/33	Small
$N \geq 500$	22/32	Large

The benchmark is synthetic; the instances follow Uchoa et al. (2017).

Table 14: CVRPLib-X (65 instances) (Uchoa et al., 2017). Gap to the best-known solution; best row entry in bold.

Instance	N	Optimal	RF-TE		CaDA		CCL		Ours	
			Obj. ↓	Gap ↓	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓	Obj. ↓	Gap ↓
X-n101-k25	100	27591	29087	5.422%	29185	5.777%	**28900**	**4.744%**	29066	5.350%
X-n106-k14	105	26362	27159	3.023%	26952	2.238%	27173	3.076%	**26873**	**1.938%**
X-n110-k13	109	14971	15314	2.291%	15262	1.944%	15334	2.425%	**15212**	**1.610%**
X-n115-k10	114	12747	13281	4.189%	**13169**	**3.311%**	13282	4.197%	13400	5.131%
X-n120-k6	119	13332	13765	3.248%	13735	3.023%	13821	3.668%	**13673**	**2.565%**
X-n125-k30	124	55539	58570	5.457%	**57405**	**3.360%**	57732	3.949%	58697	5.686%
X-n129-k18	128	28940	29598	2.274%	**29397**	**1.579%**	29548	2.101%	29583	2.225%
X-n134-k13	133	10916	11624	6.486%	11512	5.460%	11476	5.130%	**11376**	**4.214%**
X-n139-k10	138	13590	**13756**	**1.221%**	13877	2.112%	13938	2.561%	13854	1.943%
X-n143-k7	142	15700	16257	3.548%	16195	3.153%	16213	3.268%	**16037**	**2.153%**
X-n148-k46	147	43448	**45036**	**3.655%**	45761	5.324%	45522	4.774%	46012	5.901%
X-n153-k22	152	21220	23478	10.641%	23154	9.114%	**23105**	**8.883%**	23300	9.802%
X-n157-k13	156	16876	17339	2.744%	17344	2.773%	**17123**	**1.464%**	17204	1.944%
X-n162-k11	161	14138	14664	3.720%	14814	4.781%	**14638**	**3.537%**	14641	3.565%
X-n167-k10	166	20557	21432	4.256%	21437	4.281%	**20940**	**1.863%**	21107	2.680%
X-n172-k51	171	45607	48118	5.506%	48181	5.644%	48142	5.558%	**47939**	**5.113%**
X-n176-k26	175	47812	**51400**	**7.504%**	52698	10.219%	52533	9.874%	52974	10.796%
X-n181-k23	180	25569	26106	2.100%	26099	2.073%	26218	2.538%	**26060**	**1.924%**
X-n186-k15	185	24145	25148	4.154%	25461	5.450%	25096	3.939%	**24971**	**3.425%**
X-n190-k8	189	16980	17868	5.230%	18470	8.775%	17572	3.486%	**17504**	**3.092%**
X-n195-k51	194	44225	47390	7.157%	46726	5.655%	**46568**	**5.298%**	47869	8.240%
X-n200-k36	199	58578	61199	4.474%	61198	4.473%	61433	4.874%	**60959**	**4.065%**
X-n204-k19	203	19565	20594	5.259%	20497	4.764%	20557	5.070%	**20387**	**4.206%**
X-n209-k16	208	30656	31876	3.980%	32092	4.684%	32453	5.862%	**31487**	**2.714%**
X-n214-k11	213	10856	11670	7.498%	11812	8.806%	11560	6.485%	**11489**	**5.831%**
X-n219-k73	218	117595	120349	2.342%	120464	2.440%	124930	6.238%	**118943**	**1.146%**
X-n223-k34	222	40437	42251	4.486%	42359	4.753%	**42103**	**4.120%**	42191	4.338%
X-n228-k23	227	25742	28798	11.872%	27988	8.725%	**27731**	**7.727%**	29688	15.333%
X-n233-k16	232	19230	20713	7.712%	20638	7.322%	**20296**	**5.543%**	20435	6.266%
X-n237-k14	236	27042	29595	9.441%	30451	12.606%	29943	10.728%	**29227**	**8.080%**
X-n242-k48	241	82751	**85704**	**3.569%**	85780	3.660%	86015	3.944%	86884	4.995%
X-n247-k50	246	37274	**40642**	**9.036%**	41037	10.096%	40954	9.873%	41748	12.006%
X-n251-k28	250	38684	**40134**	**3.748%**	40663	5.116%	40484	4.653%	40185	3.883%
X-n502-k39	501	69226	71742	3.634%	72427	4.624%	72438	4.640%	**71393**	**3.130%**
X-n513-k21	512	24201	28542	17.937%	30037	24.115%	27700	14.458%	**26968**	**11.438%**
X-n524-k153	523	154593	173568	12.274%	**171656**	**11.037%**	178458	15.437%	174991	13.195%
X-n536-k96	535	94846	103499	9.123%	102768	8.352%	106288	12.064%	**102143**	**7.694%**
X-n548-k50	547	86700	**100681**	**16.126%**	102813	18.585%	106464	22.796%	100838	16.308%
X-n561-k42	560	42717	49406	15.659%	50410	18.009%	48094	12.587%	**47607**	**11.447%**
X-n573-k30	572	50673	55876	10.268%	56622	11.740%	**55593**	**9.709%**	56126	10.761%
X-n586-k159	585	190316	205224	7.833%	205385	7.918%	232361	22.092%	**203415**	**6.883%**
X-n599-k92	598	108451	**116732**	**7.636%**	117727	8.553%	129174	19.108%	117147	8.019%
X-n613-k62	612	59535	67530	13.429%	68696	15.388%	**67489**	**13.360%**	67551	13.464%
X-n627-k43	626	62164	67415	8.447%	68838	10.736%	71602	15.182%	**66858**	**7.551%**
X-n641-k35	640	63684	70670	10.970%	73329	15.145%	73053	14.712%	**69361**	**8.916%**
X-n655-k131	654	106780	112134	5.014%	110761	3.728%	122640	14.853%	**109485**	**2.534%**
X-n670-k130	669	146332	168867	15.400%	**165711**	**13.243%**	183348	25.296%	175307	19.801%
X-n685-k75	684	68205	77727	13.961%	78145	14.574%	77958	14.300%	**77429**	**13.525%**
X-n701-k44	700	81923	91008	11.090%	92254	12.611%	94292	15.098%	**89776**	**9.586%**
X-n716-k35	715	43373	49503	14.133%	51313	18.306%	49893	15.032%	**48365**	**11.509%**
X-n733-k159	732	136187	148369	8.945%	**148357**	**8.936%**	163255	19.876%	153700	12.860%
X-n749-k98	748	77269	**84985**	**9.986%**	85634	10.826%	88803	14.927%	85874	11.138%
X-n766-k71	765	114417	130109	13.715%	**128140**	**11.994%**	138265	20.843%	140420	22.727%
X-n783-k48	782	72386	82894	14.517%	84805	17.157%	87558	20.960%	**82117**	**13.443%**
X-n801-k40	800	73311	85941	17.228%	89329	21.849%	92808	26.595%	**85218**	**16.242%**
X-n819-k171	818	158121	174037	10.066%	173461	9.701%	231883	46.649%	**169868**	**7.429%**
X-n837-k142	836	193737	208396	7.566%	208225	7.478%	274880	41.883%	**208063**	**7.395%**
X-n856-k95	855	88965	98455	10.667%	100057	12.468%	123333	38.631%	**96828**	**8.838%**
X-n876-k59	875	99299	107300	8.057%	110332	11.111%	115958	16.777%	**107205**	**7.962%**
X-n895-k37	894	53860	64631	19.998%	67511	25.345%	65179	21.016%	**61772**	**14.690%**
X-n916-k207	915	329179	353584	7.414%	353009	7.239%	509703	54.841%	**350865**	**6.588%**
X-n936-k151	935	132715	162220	22.232%	**154841**	**16.672%**	195452	47.272%	172308	29.833%
X-n957-k87	956	85465	102788	20.269%	105200	23.091%	135339	58.356%	**101754**	**19.060%**
X-n979-k58	978	118976	129946	9.220%	133144	11.908%	148811	25.076%	**127163**	**6.882%**
X-n1001-k43	1000	72355	85828	18.621%	89231	23.324%	113779	57.251%	**83629**	**15.582%**
Avg. Gap			8.595%		9.373%		14.356%		**8.286%**	
# Best (Best/Total)			9/65		8/65		11/65		**37/65**	

D HCVRP ADAPTATION: FULL RESULTS

This appendix expands Section 5.3. Every number is a single-downstream-seed value (seed 1234) unless stated otherwise, and none of it is pooled with the three-seed study of Section 5.4.

D.1 PROTOCOL CONTRACTS

Data and evaluation. All fresh adaptation runs use the same $N{=}60$ random-fleet data contract with 3 to 7 vehicles; the test files are the PARCO $N{=}60$ instances with 3, 5 and 7 vehicles (1,280 each, hashes recorded per run) and the four Li min-sum cells N40/M3, N60/M3, N80/M3 and N80/M5 (1,280 each). Li costs are travel times with the $\times 4$ unit conversion applied once. Each cell reports the mean over all 1,280 instances at 1,280 candidates; every cell of the fresh runs reached 1,280/1,280 instances with zero invalid attempts.

Search allowance and main run. The planned HPO allowance, separate from the main run, is three learning rates $\times 100\text{K} = 300\text{K}$ target exposures for each Li system and RF-TE/CCL in the min-max part of Table 3. For the planned protocol, the grid and selected rates are listed in Section B.5, and the recorded per-condition settings in Table 15. Ours* reuses the ABC seed-1234 1M full-FT run without additional training or HPO, trained with a fixed rate of 10^{-4}, no search, fp16, one GPU and $8 \times 40 \times 4$ sampling; the fresh runs use bf16, two GPUs and 8×160 sampling.

Table 15: Recorded source sizes and adaptation/evaluation settings. LR is fixed for reused Ours* and validation-selected for the fresh runs. GPU counts refer to training; grouping describes evaluation candidates.

Model	Objective	Source N	LR	Precision	GPUs	Grouping
Ours	min-sum	100	$3e-04$	bfloat16	2	8×160
RF-TE	min-sum	100	$3e-04$	bfloat16	2	8×160
RF-TE	min-max	100	$1e-04$	bfloat16	2	8×160
CCL	min-sum	100	$1e-04$	bfloat16	2	8×160
CCL	min-max	100	$3e-04$	bfloat16	2	8×160
Ours*	min-max	50	$1e-4$	fp16	1	$8 \times 40 \times 4$

Source sizes differ as well: $N{=}100$ for the three Li rows and for RF-TE/CCL min-max, $N{=}50$ for Ours*. Equal source size does not mean equal source mixture or compute.

Published references. Published rows use separate training budgets and are not paired-instance comparisons. Li reference values follow Li et al. (2022), whose Exact row uses Pessoa et al. (2020). The min-max SISR, PARCO and 2D-Ptr values follow Berto et al. (2025b); 2D-Ptr's method is due to Liu et al. (2024c) and SISR to Christiaens & Vanden Berghe (2020). ECHO values follow Wu et al. (2026b) (arXiv v3, Tables 5–7): best-of-1,280 sampling and 10,000 test instances per setting are reported, whereas its released files contain 1,280 instances each. Its published mean has not been verified on that released subset. Equal candidate counts do not establish identical augmentation, vehicle permutation, sampling RNG or decoding.

D.2 ENDPOINT EFFECT SIZES AND CONDITIONAL STATISTICS

The endpoint means are those of Table 3. With $100 \times (\text{baseline} - \text{ours})/\text{baseline}$ per cell, Ours is 0.51–0.69% below RF-TE in the four Li cells, 0.033% above CCL at N40/M3 and 0.094–0.543% below CCL in the other three; Ours* is 0.91–1.39% below RF-TE and 0.88–1.63% below CCL in the three min-max cells. Win counts on the unrounded means are 4/4 and 3/4 (Li) and 3/3 and 3/3 (min-max), with no exact ties. A separate paired-instance aggregate (per-instance ratio, then equal-weight fleet mean) on the random-fleet min-sum test gives improvements of 0.435% [0.409, 0.462] over RF-TE and 0.296% [0.269, 0.322] over CCL at 1M, and 0.285% [0.260, 0.310] and 0.012% [−0.013, 0.036] at 200K (Table 16); the intervals resample instances conditional on the one trained seed and say nothing about training-seed uncertainty. This aggregate is not interchangeable with the cell-mean ratios above. Min-max is excluded from the paired analysis because Ours* is a documented reuse with a different source and recipe.

Table 16: Conditional paired-instance uncertainty for random-fleet min-sum. Improvement is the mean of paired instance ratios, then an equal fleet mean; it differs from a ratio of mean costs and from the Li transfer metric.

Budget	Baseline	Improvement (%)	95% CI
200K	RF-TE	0.285	[0.260, 0.310]
200K	CCL	0.012	[-0.013, 0.036]
1M	RF-TE	0.435	[0.409, 0.462]
1M	CCL	0.296	[0.269, 0.322]

The interval resamples test instances conditional on one trained seed; it does not measure training-seed uncertainty.

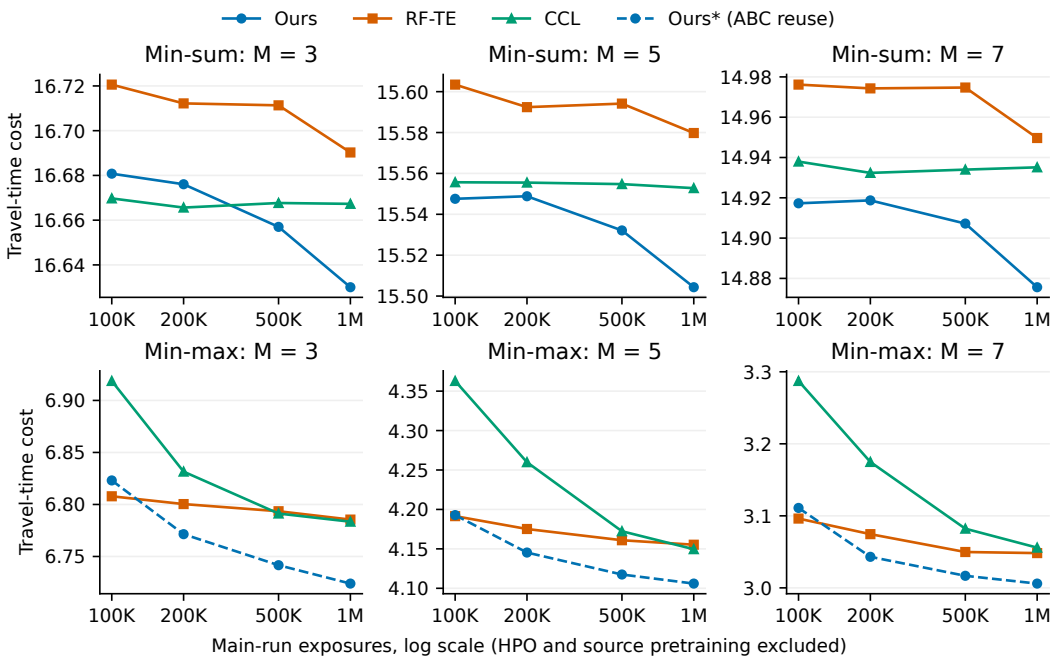


Figure 3: Random-fleet N=60 costs at four adaptation budgets (log axis), seed 1234 and 1,280 candidates. Top: min-sum for Ours/RF-TE/CCL. Bottom: min-max for RF-TE/CCL and reused Ours* (dashed). Main-run exposures exclude HPO and source pretraining; one seed gives no seed error bars.

D.3 Budget curves

The curves evaluate 100K, 200K, 500K and 1M main-run exposures. Min-max Ours* comes from the seed-1234 full fine-tuning runs of the ABC study at the same four budgets, the runs whose 1M endpoint Table 3 reuses; its source (N=50) and recipe differ from RF-TE/CCL as listed above.

Table 17 reports min-sum for Ours, RF-TE and CCL and min-max for Ours*, RF-TE and CCL. On min-sum, Ours has lower cost than RF-TE in all 12 (budget, fleet) cells. Against CCL, the M3 column is the exception at low budget: 16.6808 (Ours) against 16.6697 (CCL) at 100K and 16.6761 against 16.6656 at 200K; at 500K and 1M Ours is lower (16.6570 vs. 16.6677 and 16.6300 vs. 16.6673), and at M5 and M7 Ours is lower at all four budgets. On min-max, RF-TE is lower than Ours* in all three fleets at 100K (6.8079, 4.1915 and 3.0962 against 6.8231, 4.1928 and 3.1110); from 200K on, Ours* is lower than both RF-TE and CCL in all three fleets, and it is lower than CCL at every budget. These comparisons share the source and recipe differences listed above, and one seed of sparse checkpoints supports no statement about convergence speed or wall-clock.

D.4 Pretraining-benefit study by budget

Table 18 lists the twelve (budget, fleet) means and sample SDs over the three downstream seeds behind Figure 2; A is below B at every budget and fleet, and C lies between them at every budget and

Table 17: All-instance costs at the recorded main-run budgets, seed 1234; 1,280 candidates.

Min-sum

Budget	Fleet	Ours	RF-TE	CCL
100K	3	16.6808	16.7206	16.6697
100K	5	15.5476	15.6034	15.5557
100K	7	14.9173	14.9762	14.9380
200K	3	16.6761	16.7122	16.6656
200K	5	15.5488	15.5924	15.5555
200K	7	14.9187	14.9743	14.9324
500K	3	16.6570	16.7113	16.6677
500K	5	15.5321	15.5942	15.5548
500K	7	14.9072	14.9747	14.9340
1M	3	16.6300	16.6903	16.6673
1M	5	15.5043	15.5798	15.5528
1M	7	14.8756	14.9497	14.9351

Min-max

Budget	Fleet	Ours*	RF-TE	CCL
100K	3	6.8231	6.8079	6.9187
100K	5	4.1928	4.1915	4.3629
100K	7	3.1110	3.0962	3.2872
200K	3	6.7714	6.8004	6.8316
200K	5	4.1453	4.1752	4.2599
200K	7	3.0431	3.0746	3.1748
500K	3	6.7416	6.7935	6.7913
500K	5	4.1175	4.1609	4.1723
500K	7	3.0168	3.0498	3.0822
1M	3	6.7239	6.7855	6.7834
1M	5	4.1060	4.1553	4.1496
1M	7	3.0059	3.0483	3.0558

Ours* reuses the seed-1234 ABC full fine-tuning runs (N50 source, different recipe; Section D).

stays above A at 1M. The three configurations share one target recipe (Section B.5), so the scratch configuration is not independently tuned, and source pretraining cost is excluded.

Table 18: ABC mean ± sample SD over 3 downstream seeds (1234, 2345, 3456), best of 1,280. Fixed source initialization; source pretraining cost is excluded.

Exposures	Fleet	A: full FT	B: scratch	C: projections
100K	3	6.8216 ± 0.0094	7.6569 ± 0.0575	6.9322 ± 0.0049
100K	5	4.1916 ± 0.0039	4.9568 ± 0.0545	4.3414 ± 0.0010
100K	7	3.1084 ± 0.0072	3.8435 ± 0.0602	3.2892 ± 0.0009
200K	3	6.7723 ± 0.0008	7.3906 ± 0.0655	6.8350 ± 0.0009
200K	5	4.1429 ± 0.0026	4.6476 ± 0.0705	4.2198 ± 0.0019
200K	7	3.0376 ± 0.0048	3.5018 ± 0.0755	3.1386 ± 0.0002
500K	3	6.7391 ± 0.0025	7.1381 ± 0.0076	6.8045 ± 0.0021
500K	5	4.1175 ± 0.0002	4.3631 ± 0.0086	4.1778 ± 0.0015
500K	7	3.0151 ± 0.0016	3.2035 ± 0.0188	3.0903 ± 0.0012
1M	3	6.7227 ± 0.0020	6.9343 ± 0.0414	6.7948 ± 0.0007
1M	5	4.1066 ± 0.0010	4.2424 ± 0.0231	4.1622 ± 0.0015
1M	7	3.0076 ± 0.0030	3.1043 ± 0.0173	3.0703 ± 0.0021

All configurations use a common target recipe; scratch is not independently tuned. No legacy continuation is included.

D.5 Separate earlier results

The following results come from earlier single runs made before the 1M study; they are reported for completeness and are not rows, averages or curve points of the 1M study.

Li-specialized fine-tuning. An earlier single model was fine-tuned from the $N{=}100$ source directly on the Li fixed-fleet distribution (N from 40 to 80, 3 to 5 vehicles), for up to 3M exposures with validation-selected checkpoint, seed 1234. Its best-of-1,280 means were 56.52, 79.75, 104.35 and 104.36 on N40/M3, N60/M3, N80/M3 and N80/M5 (Table 19), i.e. 1.97%, 1.64%, 1.89% and 1.90% above the exact reference; these are below the DRL-Li sampling row in all four cells and above SISR in all four. It uses more target exposures and target-specific data than the 1M random-fleet rows and is not a substitute for them.

5M single-run reference. In the study of Section 5.4, one additional scratch run and the earlier single runs of A and C at 5M exposures give 6.6870/4.0700/2.9660 (A), 6.7264/4.0970/2.9898 (B) and 6.7625/4.1418/3.0445 (C) for 3/5/7 vehicles ($n{=}1$; Table 20). C at 5M updates 68,480 policy parameters, about 3.4% of the policy, with the backbone frozen. These points are not a continuation of the three-seed curves and are not combined with the 200K observation of Section 5.4.

Table 19: Earlier Li-specialized model, separate from random-fleet adaptation. All-instance mean in Li travel-time units, with the stored conversion already applied. N100 source, seed 1234, a maximum 3M-exposure Li fixed-fleet run and validation checkpoint selection; 1,280 candidates.

Model	N40/M3	N60/M3	N80/M3	N80/M5
Ours (Li-specialized)	56.5235	79.7548	104.3517	104.3648

Not part of the random-fleet results.

Table 20: Separate long-budget reference from the ABC summary. A/C are single runs from an earlier study; B is a separate scratch run. These rows are not a continuation of the controlled multi-seed curves.

Exposures	Fleet	A: full	B: scratch	C: projections
5M	3	6.6870 (n=1)	6.7264 (n=1)	6.7625 (n=1)
5M	5	4.0700 (n=1)	4.0970 (n=1)	4.1418 (n=1)
5M	7	2.9660 (n=1)	2.9898 (n=1)	3.0445 (n=1)

Source pretraining cost is excluded; the projection-only configuration updates input projections with the backbone fixed.

The partially completed $N{=}100$-target study is reported in Section F.

E DIAGNOSTICS

This appendix expands Section 5.5: the full configuration table with the MD and single-depot MB split, completion and inference measurements, the geometric prior, the earlier component ablation, and allocation diversity.

E.1 CONFIGURATIONS BY CONSTRAINT GROUP

Table 21: Construction configurations by constraint group. Gap to PyVRP (%, mean ± sample SD across variants).

Configuration	ID16	Unseen32	MD24	SD-MB8
Full (ours)	1.327 ± 0.843	11.740 ± 4.681	13.100 ± 4.285	7.662 ± 3.372
2D-Ptr-style (our impl.)	1.636 ± 0.990	20.232 ± 8.987	24.367 ± 5.637	7.825 ± 4.330
POMO-style forced multi-start	1.555 ± 0.860	12.025 ± 4.826	13.612 ± 3.988	7.266 ± 4.044
− distance prior	1.437 ± 0.904	17.987 ± 7.377	21.454 ± 4.378	7.585 ± 3.630
− agent/virtual-depot token	1.430 ± 0.854	13.578 ± 6.571	15.526 ± 6.239	7.736 ± 3.359
− communication	1.534 ± 0.902	11.564 ± 4.545	12.569 ± 4.479	8.552 ± 3.423

Table 21 reports all six configurations for ID16, Unseen32 and its MD24/SD-MB8 split, as mean gaps and sample SDs across variants. These SDs describe variant heterogeneity, not training-seed uncertainty. The corresponding per-variant differences are in Table 22. The group results are read separately; marginal removals do not identify additive contributions.

Each configuration is one training run (switches in Table 5); the selected checkpoints are epoch 295 (full, 2D-Ptr-style), 297 (no communication) and 299 (no prior, no agent tokens, POMO-style). The $N{=}50$ ID values of Table 8 use 240 candidates for our model, whereas the configurations here use 400, so the two are not directly comparable.

E.2 COMPLETION AND INFERENCE MEASUREMENTS

Three of the 48,000 instances were not completed by any configuration and are excluded; truncation shares and evaluation times are listed in Table 23.

E.3 THE GEOMETRIC PRIOR

The prior is a non-learned distance term added to the pair scores before clipping; it carries no parameters and passes no gradient (Section 4.2). In an earlier evaluation, removing it changed the ID gap by $+0.11$ pp, the CVRPLib X gap by $+1.65$ pp and the unseen gap by $+5.10$ pp, about 37% of the 13.6 pp margin over CCL in that evaluation; the change was $+6.97$ pp on the 24 MD variants and -0.08 pp on the 8 single-depot MB variants. The retrained no-prior configuration under the current evaluator is reported by group in Table 21. The two evaluations differ in evaluator and multi-depot semantics and are not summed. The prior and the self-depot return mask are distinct: the mask decides which pairs are feasible, the prior only re-weights feasible pairs.

E.4 EARLIER COMPONENT ABLATION

An earlier ablation retrained one element at a time at $N{=}50$ with 8×50 candidates under the evaluator used at that time, with the full model at a 1.386% ID gap: removing communication $+0.21$ pp, the prior $+0.11$ pp, the agent tokens $+0.11$ pp; replacing allocation multi-start by POMO-style forced initial-customer multi-start $+0.22$ pp; and replacing the single-pair softmax by a PARCO-style parallel decoder (Berto et al., 2025b) $+8.09$ pp. The last change alters the action factorization rather than removing an element, and the decoder swap is not a controlled comparison of the two published systems. These values are kept separate from Section E.1, whose configurations were re-evaluated with a different evaluator and multi-depot semantics.

Table 22: Per-variant gap of Full (%) and differences of each configuration from Full (pp). Ptr: 2D-Ptr-style (our impl.); POMO: forced multi-start.

Variant	Full	Ptr	POMO	−prior	−token	−comm.
CVRP	1.052	0.107	0.032	0.105	0.100	0.078
MDCVRP	8.663	9.419	0.274	7.862	0.618	-0.152
MDOVRP	9.474	6.373	0.907	10.945	-3.429	0.058
MDOVRPB	9.865	6.211	0.867	9.000	-3.850	-0.389
MDOVRPBL	9.880	6.627	0.759	8.807	-3.790	-0.432
MDOVRPBLTW	11.244	13.218	1.249	9.788	4.703	-0.961
MDOVRPBTW	11.124	13.131	1.345	10.076	4.926	-1.115
MDOVRPL	9.438	6.945	0.576	10.806	-3.316	-0.000
MDOVRPLTW	9.752	13.311	1.181	8.795	4.853	-0.720
MDOVRPMB	21.274	8.491	0.603	9.541	-5.082	0.154
MDOVRPMBL	21.343	9.118	0.431	9.306	-5.075	0.037
MDOVRPMBLTW	16.204	13.336	-0.279	9.203	4.788	-0.639
MDOVRPMBTW	16.027	13.586	-0.072	9.544	4.872	-0.680
MDOVRPTW	9.644	13.050	1.284	9.120	5.016	-0.718
MDVRPB	10.307	8.566	0.199	7.111	-0.080	-0.317
MDVRPBL	10.510	10.596	0.562	7.675	0.532	0.316
MDVRPBLTW	12.202	13.834	0.795	7.416	8.855	-1.627
MDVRPBTW	12.261	13.678	0.931	7.168	8.642	-1.857
MDVRPL	8.850	9.776	0.090	7.237	0.969	-0.185
MDVRPLTW	10.781	14.879	0.929	6.336	8.924	-1.218
MDVRPMB	19.570	11.101	-0.023	7.100	-0.533	0.144
MDVRPMBL	19.200	12.421	-0.058	7.705	0.069	0.376
MDVRPMBLTW	17.881	14.546	-0.658	6.941	8.667	-0.798
MDVRPMBTW	18.110	14.063	-0.492	6.603	8.112	-0.697
MDVRPTW	10.782	14.153	0.892	6.436	8.836	-1.320
OVRP	2.263	0.580	0.240	0.212	-0.083	0.390
OVRPB	1.786	0.344	0.162	0.137	-0.148	0.240
OVRPBL	1.760	0.373	0.204	0.167	-0.110	0.272
OVRPBLTW	0.457	0.150	0.180	0.028	0.104	0.126
OVRPBTW	0.455	0.156	0.177	0.030	0.094	0.122
OVRPL	2.264	0.580	0.236	0.207	-0.107	0.357
OVRPLTW	0.449	0.185	0.251	0.048	0.104	0.154
OVRPMB	12.609	1.493	0.384	0.274	-0.033	1.196
OVRPMBL	12.589	1.454	0.344	0.269	-0.002	1.174
OVRPMBLTW	4.674	-0.686	-1.227	-0.436	-0.108	0.813
OVRPMBTW	4.733	-0.717	-1.229	-0.488	-0.176	0.737
OVRPTW	0.444	0.189	0.247	0.062	0.116	0.159
VRPB	2.440	0.490	0.218	0.215	0.204	0.251
VRPBL	3.100	0.548	0.364	0.183	0.425	0.227
VRPBLTW	0.852	0.270	0.270	0.089	0.220	0.256
VRPBTW	0.671	0.220	0.285	0.078	0.131	0.184
VRPL	1.277	0.126	0.051	0.090	0.213	0.064
VRPLTW	1.170	0.373	0.341	0.067	0.250	0.250
VRPMB	8.321	0.643	0.192	0.017	0.221	0.272
VRPMBL	8.085	0.407	0.170	0.003	0.178	0.171
VRPMBLTW	5.079	-0.567	-0.871	-0.067	0.335	1.358
VRPMBTW	5.206	-0.725	-0.930	-0.187	0.177	1.398
VRPTW	0.788	0.266	0.389	0.058	0.150	0.187

Table 23: Recorded inference measurements of the construction configurations: mean share of truncated candidate rollouts and evaluation time over the 48 variants.

Configuration	Truncated (%)	Seconds
Full (ours)	1.325	335.03
2D-Ptr-style (our impl.)	0.977	179.02
POMO-style forced multi-start	1.727	363.00
− distance prior	1.662	305.16
− agent/virtual-depot token	2.285	331.12
− communication	1.565	175.78

Time includes warm-up.

E.5 Allocation diversity

Allocation diversity is one minus the adjusted Rand index between the customer-to-route partitions of two rollouts, averaged over 16 ID variants $\times$ 64 instances $\times$ 50 rollouts without augmentation. The allocation-driven rollouts give 0.683 against 0.623 for POMO-style forced initial-customer starts (higher on 16/16 variants, sign test $p = 1.5 \times 10^{-5}$); a second, order-based diversity index showed no statistically detectable difference ($p = 0.11$). With a single greedy rollout the model trained with allocation-driven multi-start reaches a 4.100% ID gap against 4.426% for the forced-start model, so the difference is produced at training time (Table 24). Rollouts are not guaranteed to differ in allocation.

Table 24: Allocation diagnostics. Partition diversity is one minus ARI over 16 variants, 64 instances per variant and 50 rollouts without augmentation. The greedy row is a separate single-rollout evaluation.

Measure	Allocation	POMO
Partition diversity D_α	0.683	0.623
Single-rollout greedy gap (%)	4.100	4.426

F Resources and scope

F.1 Time, memory and parameters

The selected $N{=}50$ source checkpoint is at about 116K steps, with a recorded training time of about 27 hours on one RTX 4090. At inference, the earlier resource measurements give a peak memory of 3.63 GiB for our model against 0.94 GiB for the lightest compared baseline, a factor of 3.9, and 16–19 s per 1,000-instance cell at $N{=}100$ against 8–9 s for that baseline; our policy has 2.02M parameters against 4.05M, so the memory cost is activation memory of the communication block and of the flattened pair scores rather than parameter count. At $N{=}50$ the same measurement gives 4–5 s per cell for our model against 5–6 s for CCL. No runtime- or memory-matched configuration was constructed. In the configuration comparison of Section E, wall-clock over the 48 variants with warm-up included was 335 s for Full against 176 s without communication and 179 s for the 2D-Ptr-style pointer (Table 23), so the communication block accounts for about half of the inference time at this size. The recorded evaluation time and peak memory of the 1M HCVRP runs are listed in Table 25; they are implementation-specific measurements of different code bases. Equal candidate counts do not imply equal time or memory.

F.2 Slot count

The communication block attends over the active slots and the pair scores span active slots $\times$ targets, so per-step cost grows quadratically in the number of active slots and the pair grid linearly. A sweep of the slot limit over 8, 10, 20, 50 and 100 at $N{=}50$ (batch 64) gives ID gaps of 1.377%, 1.376%, 1.386%, 1.631% and 2.135%, with measured time rising from 11.4 to 74.5 ms and peak memory from 1.88 to 15.47 GiB between the two endpoints (Table 26). The table separates quality at all recorded

Table 25: Recorded evaluation resources at 1M, 1,280 candidates, seed 1234; ranges over the 3 fleets. Time covers the evaluation run; peak memory is in MB (10^6 bytes). Ours min-max uses a different evaluator and is omitted.

Objective	Model	Seconds	Peak memory (MB)
min-sum	Ours	1366.80–1431.44	69.8–121.5
min-sum	RF-TE	403.27–542.81	165.5–171.3
min-sum	CCL	997.61–1003.38	68.3–71.0
min-max	RF-TE	425.84–438.75	164.2–170.1
min-max	CCL	981.70–1011.92	68.3–71.0

These are implementation-specific measurements; equal candidates do not establish equal compute.

caps from the two recorded resource endpoints; time and memory were recorded only at the two endpoints.

Table 26: Slot-cap sweep at N50, batch 64 (slots are construction slots, not vehicles).

Quality at every recorded slot cap

Slot cap	Gap (%)
8	1.377
10	1.376
20	1.386
50	1.631
100	2.135

Recorded resource endpoints

Slot cap	Time (ms)	Memory (GiB)
8	11.4	1.88
100	74.5	15.47

Source training took approximately 116K steps and 27 hours on RTX 4090; this cost is excluded from the target-exposure curves.

Quality is flat up to 20 slots and degrades from 50, beyond the training range of at most 8; the cap bounds the routes under construction at one time, not the routes in a solution, because a slot starts a new route after a depot return (Section A.2). This sweep is an N=50 experiment and does not measure the CVRPLib runs of Section C. The label symmetry of slots is not removed by the model and is one source of multi-start diversity; it is not treated as a cost.

F.3 Partially completed N100-target study

A separate min-max study adapted to N=100 random-fleet instances (3 to 7 vehicles, tested at 3/5/7) from two starting points: the same-seed N=60 full fine-tuning checkpoint at 1M (from N60) and the original N=50 source used in Section 5.4 (from base), each with a fresh optimizer, rate 10^{-4}, the same N=100 data stream per seed and up to 1M additional target instances; there is no scratch configuration. The study is incomplete: 11 of 15 checkpoints and 66 of 90 evaluation cells were evaluated, and both seed-3456 1M runs did not finish and are not included. Available values (best of 1,280, mean $\pm$ SD over the available downstream seeds) are given in Table 27; from N60 has already consumed 1M N=60 instances, so equal additional budget is not equal total target budget.

No three-seed curve is drawn from these n=2 rows, and the study is not a main-text claim.

F.4 Scope of the evidence

Weight-frozen transfer was evaluated up to N=1,000 on CVRPLib only, and on constraint families within the six-attribute alphabet; pickup-and-delivery, split delivery, stochastic demand and non-Euclidean metrics were not tested. Fleet adaptation covers 3 to 7 physical vehicles at $N \leq 100$, with vehicle counts inside the training range, so no extrapolation to unseen vehicle counts is claimed;

Table 27: Incomplete N100-target study: available downstream-seed means and sample SD only. The from-base configuration starts at the ABC source; from-N60 has already used the listed N60 exposures. Main-run budgets exclude source pretraining.

Source	N60 used	N100 added	n	M3	M5	M7
from base	0	200K	2	10.6139 ± 0.0012	6.3690 ± 0.0004	4.6058 ± 0.0031
from base	0	1M	2	10.5236 ± 0.0019	6.3155 ± 0.0010	4.5605 ± 0.0009
from n60	1M	0	3	10.7250 ± 0.0048	6.3806 ± 0.0046	4.6103 ± 0.0068
from n60	1M	200K	2	10.5216 ± 0.0032	6.3055 ± 0.0037	4.5489 ± 0.0014
from n60	1M	1M	2	10.4920 ± 0.0010	6.2824 ± 0.0009	4.5279 ± 0.0007

The study is incomplete; missing training replicates are not imputed, and equal additional N100 exposures do not imply equal total adaptation exposure.

the Li cells are a size and fixed-fleet transfer within that range. The min-sum and min-max models are separate adaptations, and the two objectives also change the fleet, so no factorial objective-only effect is identified. The cross-system comparisons of Section 5.3 are n=1 and the within-architecture study of Section 5.4 is n=3 downstream seeds; the pretraining seed was replicated only for the frozen-transfer means of Section C.

G Related-work comparison matrix

Tables 28 and 29 summarize the works closest to ours. "Not reported" marks an item we did not find in the paper. "Single route" means that one vehicle is extended at a time, with depot returns separating vehicles. "Pair" means one (vehicle, node) pair drawn from a distribution over active vehicles and targets. Fields that do not apply, such as policy-specific communication, training and adaptation for the environment library or an external comparison type for our own row, are left out. The comparison field states how each work enters this paper. *Direct* rows are models we evaluated ourselves in Section 5; *published reference* rows report numbers from the original paper under their own training budgets; *component source* marks a module or rule we reuse; *literature only* works are discussed but not compared numerically. The matrix describes differences in scope and mechanism and does not rank the works; our row uses the same fields without serving as a checklist.

Table 28: Multi-variant and cross-problem pretraining. See the text for comparison categories.

Work	Construction	Training, reuse and comparison
MTPOMO (Liu et al., 2024b)	node tokens with attribute composition; single route Action: node. Communication: none (single route).	Source: 5 CVRP variants. Reuse: zero-shot to 6 variants; fine-tuning on unseen variants. Comparison: direct (in-distribution).
MVMoE (Zhou et al., 2024)	node tokens; MoE in encoder and decoder; single route Action: node. Communication: none (single route).	Source: 6 CVRP variants. Reuse: zero-shot to 10 variants; few-shot fine-tuning. Comparison: direct (in-distribution).
RouteFinder (Berto et al., 2025a)	node tokens, global attributes on the depot, current-vehicle state; single route Action: node. Communication: none (single route).	Source: 16 variants, mixed batches. Reuse: EAL fine-tuning to 32 MB/MD variants; backbone freezing not stated. Comparison: direct (RF-TE: frozen transfer, HCVRP adaptation); component source (EAL, instance generator).
CaDA (Li et al., 2025)	constraint prompt, dual attention; single route Action: node. Communication: none (single route).	Source: 16 variants. Reuse: not reported (evaluated on the 16 training variants). Comparison: direct (frozen transfer); component source (constraint prompt).
CCL (Gui et al., 2026)	node tokens re-embedded each step from trajectory context; single route Action: node. Communication: none (single route).	Source: 16 variants. Reuse: zero-shot to 32 MB/MD variants. Comparison: direct (frozen transfer, HCVRP adaptation).

Continued on next page

Table 28 (continued)

Work	Construction	Training, reuse and comparison
SHIELD (Goh et al., 2025)	node tokens, mixture-of-depths decoder, clustering; single route Action: node. Communication: none (single route).	Source: 6 variants, 3 map distributions. Reuse: zero-shot to 10 variants and 6 distributions. Comparison: literature only.
MTL-KD (Zheng et al., 2025)	heavy-decoder student; single route Action: node. Communication: none (single route).	Source: 6 variants, distilled from single-task teachers. Reuse: zero-shot to 10 variants; up to 1,000 nodes. Comparison: literature only.
MoSES (Pan et al., 2025b)	RouteFinder or CaDA backbone with LoRA experts; single route Action: node. Communication: none (single route).	Source: 16 variants on a CVRP-pretrained backbone. Reuse: not reported (all 16 variants seen in training). Comparison: literature only.
URS (Zhou et al., 2026b)	unified data representation, mixed attention bias, generated decoder parameters Action: node. Communication: not reported.	Source: 11 problems. Reuse: zero-shot to 99 unseen variants. Comparison: literature only.
POLAR+PLE (Corrêa et al., 2026b)	shared and task-specific encoder experts Action: node. Communication: not reported.	Source: 16 variants. Reuse: zero-shot to 32 MB/MD variants. Comparison: literature only.
FiLMMeD (Corrêa et al., 2026a)	FiLM-modulated customer embeddings Action: node. Communication: not reported.	Source: 24 multi-depot variants. Reuse: fine-tuning to 16 single-depot variants. Comparison: literature only.
Cross-Problem Learning (Lin et al., 2024)	TSP-pretrained Transformer with problem-specific modules Action: node. Communication: none (single route).	Source: TSP. Reuse: full fine-tuning, or adapters with frozen backbone, to OP, PCTSP, CVRP. Comparison: literature only.
GOAL (Drakulic et al., 2025)	node, edge, and instance features with task adapters; for routing, remaining capacity (and current time) of the current route as instance features; single route Action: node, optionally with one of K options (routing: direct or via depot). Communication: none (single route).	Source: 8 CO problems, imitation learning. Reuse: fine-tuning of all parameters to 8 new problems. Comparison: literature only.
UniCO (Pan et al., 2025a)	matrix-encoded general TSP Action: not reported. Communication: not reported.	Source: several CO problem types reduced to general TSP. Reuse: not reported. Comparison: literature only.
G-UniRouting (Jari et al., 2025)	graph encoder, constraint-aware decoder Action: not reported. Communication: not reported.	Source: 24 VRP variants. Reuse: not reported. Comparison: literature only.

Table 29: Vehicle, route and agent construction. See the text for comparison categories.

Work	Construction	Training, reuse and comparison
MARDAM (Bono et al., 2021)	attention-based multi-agent policy Action: not reported. Communication: not reported.	Source: dynamic and stochastic VRP. Reuse: not reported. Comparison: literature only.
DRL for HCVRP (Li et al., 2022)	node encoder; vehicle features and max-pooled partial routes Action: vehicle, then node (two decoders). Communication: feature concatenation, no attention.	Source: HCVRP min-sum and min-max, fixed fleets, per size. Reuse: size generalization, CVRPLib. Comparison: published reference.

Continued on next page

Table 29 (continued)

Work	Construction	Training, reuse and comparison
JAMPR (Falkner & Schmidt-Thieme, 2020)	node, tour, and vehicle encoders Action: pair over a capped set of concurrent routes. Communication: averaged fleet embedding.	Source: CVRP and VRPTW variants, per problem. Reuse: not reported. Comparison: literature only.
2D-Ptr (Liu et al., 2024c)	static node encoder, dynamic vehicle encoder Action: pair. Communication: self-attention among vehicles.	Source: min-max HCVRP, per fleet size and customer count. Reuse: other fleet sizes and customer counts without fine-tuning. Comparison: published reference; our 2D-Ptr-style configuration keeps its pair action.
ECHO (Wu et al., 2026b)	node and edge features; 2D-Ptr vehicle encoder; conditioning on the previous vehicle Action: pair. Communication: self-attention among vehicles.	Source: min-max HCVRP, per fleet size and customer count. Reuse: cross-scale and cross-distribution tests. Comparison: published reference.
PARCO (Berto et al., 2025b)	agent and node embeddings; dynamic agent state in the query Action: parallel, one node per agent, priority-based conflict handling. Communication: communication layers.	Source: HCVRP (min-max, random fleets) and other problems, per problem. Reuse: size and agent-count generalization reported for OMDCPDP. Comparison: published reference; component source (communication layer, agent tokens).
CAMP (Hua et al., 2025)	vehicle-profile embeddings, bipartite message passing Action: parallel, as in PARCO. Communication: decoder communication layer.	Source: two profiled VRPs, per problem. Reuse: not reported. Comparison: literature only.
MACSIM (Luttmann & Xie, 2026)	agent–task bipartite graph Action: several pairs per step from one joint score matrix. Communication: agent and task attention.	Source: FJSP, FFSP, min-max HCVRP, per problem. Reuse: larger FJSP instances. Comparison: literature only.
Vehicle-as-Prompt (Huang et al., 2026)	node, vehicle-type, and vehicle–depot views with a constraint prompt Action: vehicle type, then customers, route by route, in one action space. Communication: self-attention among vehicle types (encoder).	Source: 5 heterogeneous-fleet variants, min-sum cost. Reuse: zero-shot to 10 unseen combinations and larger sizes. Comparison: literature only.
Equity-Transformer (Son et al., 2024)	one depot token per agent Action: node, agents in sequence. Communication: none.	Source: min-max mTSP and mPDP. Reuse: fine-tuning to larger sizes. Comparison: component source (agent tokens).
DAN (Cao et al., 2024)	agent and city encoders Action: node, per agent on arrival. Communication: self-attention among agents.	Source: min-max mTSP. Reuse: larger instances without fine-tuning. Comparison: literature only.
DPN (Zheng et al., 2024)	agent and customer embeddings with partition and navigation parts Action: node, agents in sequence. Communication: none.	Source: min-max mTSP, mPDP, MDVRP, FMDVRP, per problem. Reuse: fine-tuning to larger sizes. Comparison: literature only.
MAEnvs4VRP (Gama et al., 2026)	environment library Action: agents act in turn.	Comparison: literature only (infrastructure).
This paper	slot tokens and customer tokens with a constraint prompt; per-step route states Action: pair, one per step. Communication: communication layer over active slots within a rollout.	Source: 16 CVRP variants, two source sizes. Reuse: frozen transfer to 32 MB/MD variants and CVRPLib; per-objective fine-tuning to HCVRP.