

1 A Open Infrastructure

2 To ensure reproducibility, extensibility, and accessibility, we release all components of the benchmark
3 openly on GitHub and Hugging Face. This includes the dataset, instance generator, evaluation engine,
4 and baseline implementations. Evaluation instances can be used out of the box, while the modular
5 codebase allows users to integrate new solvers and adapt evaluation scripts.

6 A public leaderboard on huggingface¹ serves as the central hub for documentation, instance down-
7 loads, and leaderboard submissions. Submissions are validated automatically and ranked by total
8 cost, feasibility, and runtime. All data and code are versioned, containerized (Docker-supported), and
9 designed to support future extensions such as new routing scenarios or solver classes.

10 We welcome community contributions, including new solvers, datasets, and improvements to docu-
11 mentation or evaluation tools. By sharing the infrastructure broadly, we aim to foster collaboration
12 and accelerate progress in realistic stochastic routing research.

13 A.1 Reproducibility Requirements

14 To maintain transparency and enable fair comparison, submissions intended for leaderboard inclusion
15 or academic publication must satisfy several criteria. Solvers must be evaluated on the official
16 benchmark test set, with all hyperparameters, configuration details, and seed values fully documented.
17 Additionally, we encourage open-source releases or detailed methodological descriptions to ensure
18 algorithm reproducibility. Runtime should be measured using the official script or a clearly defined
19 procedure, consistent across all experiments.

20 These guidelines help uphold reproducibility standards advocated in combinatorial optimization liter-
21 ature [2, 1] and promote meaningful scientific comparisons under controlled, yet realistic, conditions.

22 B Baseline Models

23 **Ant Colony Optimization (ACO).** Routes are constructed by sampling next locations based on
24 pheromone intensity and heuristic proximity. The pheromone matrix is updated as:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^{(k)}, \quad \Delta\tau_{ij}^{(k)} = \begin{cases} \frac{Q}{L^{(k)}}, & \text{if } (i, j) \in \text{tour}^{(k)} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

25 where $\rho = 0.5$, $m = 50$ ants, $\alpha = 1$, and $\beta = 2$.

26 **Tabu Search.** Candidate solutions are evaluated using a penalized cost function:

$$f(S) = \text{Cost}(S) + \lambda \cdot \text{Penalty}(S), \quad (2)$$

27 where λ is adaptively tuned based on violation severity.

28 **Learning-Based Methods.** The Attention Model is trained to minimize the expected cost:

$$\mathcal{L}(\theta) = \mathbb{E}_{X \sim \mathcal{D}} [\mathbb{E}_{\pi_\theta(a|X)} [L(a|X)]] . \quad (3)$$

29 POMO uses multiple rollout agents initialized with distinct permutations. Its gradient signal is
30 computed as:

$$\nabla_\theta J(\theta) = \frac{1}{M} \sum_{m=1}^M \sum_t \nabla_\theta \log \pi_\theta(a_t^m | s_t^m) \cdot (R^m - b), \quad (4)$$

31 where M is the number of rollouts and b is a learned baseline for variance reduction.

32 C Detailed Solver Performance Breakdowns

33 Tables 1,2,3,4,5,6 present a comprehensive performance breakdown of various solvers across multi-
34 ple configurations for Capacitated VRP (CVRP) and Time Window VRP (TWVRP). Each solver,

¹<https://huggingface.co/spaces/ahmedheak1/SVRP-leaderboard>

35 NN+2opt, Tabu Search, ACO, OR-Tools, and RL-based methods (Attention, POMO), is evaluated
 36 under different settings including depot configurations (single depot, multi depot, depots equal to
 37 cities), problem sizes (ranging from 10 to 1000 customers), and feasibility constraints. Metrics
 38 include total cost, CVR (constraint violation rate), feasibility, runtime, and time window violations.
 39 Traditional heuristic solvers (NN+2opt, Tabu, ACO) generally yield competitive costs with increasing
 40 runtimes as problem size grows, while OR-Tools offers consistent feasibility but with significantly
 41 higher runtimes. Reinforcement learning solvers (Attention, POMO) demonstrate exceptionally fast
 42 runtimes (in milliseconds), achieving full feasibility across all tested instances, although their cost can
 43 vary notably, especially for large-scale problems where some cost inflation is observed (e.g. POMO
 44 on 1000-node CVRP). These results highlight trade-offs between solution quality, computational
 45 efficiency, and scalability across solver paradigms.

Table 1: NN+2opt - Detailed Performance Breakdown.

Configuration	Size	Cost	CVR	Feas	Runtime	TW Violations
single depot single vehicule sumDemands	10	2290.7	0.0	1.000	0.0	0.00
multi depot	10	2371.8	0.0	1.000	2.0	0.00
single depot single vehicule sumDemands	20	3736.5	0.0	1.000	0.3	0.00
multi depot	20	3662.9	0.0	1.000	3.2	0.00
single depot single vehicule sumDemands	50	4840.4	0.0	1.000	10.5	0.00
multi depot	50	5626.1	0.0	1.000	14.1	0.00
single depot single vehicule sumDemands	100	6841.4	0.0	1.000	31.8	0.00
multi depot	100	7868.2	0.0	1.000	31.3	0.00
single depot single vehicule sumDemands	200	11268.2	0.0	1.000	125.2	0.00
multi depot	200	11479.2	0.0	1.000	135.5	0.00
single depot single vehicule sumDemands	500	16390.0	0.0	1.000	829.5	0.00
multi depot	500	17551.0	0.0	1.000	826.3	0.00
single depot single vehicule sumDemands	1000	25844.3	0.0	1.000	3545.9	0.00
multi depot	1000	25817.4	0.0	1.000	3493.3	0.00
depots equal city	10	4564.6	3.3	0.967	2.0	0.00
single depot	10	4359.0	3.3	0.967	2.0	0.00
depots equal city	20	8192.2	0.0	1.000	5.7	0.00
single depot	20	8347.0	0.0	1.000	5.0	0.00
depots equal city	50	13666.8	0.0	1.000	14.9	0.00
single depot	50	13882.4	0.7	0.993	11.7	0.00
depots equal city	100	38704.2	6.0	0.940	52.2	0.00
single depot	100	30389.4	1.0	0.990	37.8	0.00
depots equal city	200	89937.2	10.1	0.899	145.2	0.00
single depot	200	55400.9	1.0	0.990	167.8	0.00
depots equal city	500	175711.7	7.7	0.923	1318.5	0.00
single depot	500	118279.0	2.2	0.978	929.7	0.00
depots equal city	1000	244956.8	6.1	0.939	3865.2	0.00
single depot	1000	187829.7	2.7	0.973	3911.5	0.00

Table 2: Tabu Search - Detailed Performance Breakdown.

Configuration	Size	Cost	CVR	Feas	Runtime	TW Violations
single depot single vehicle sumDemands	10	2297.2	0.0	1.000	19.4	0.00
multi depot	10	2373.8	0.0	1.000	13.3	0.00
single depot single vehicule sumDemands	20	3776.7	0.0	1.000	47.6	0.00
multi depot	20	3656.4	0.0	1.000	33.8	0.00
single depot single vehicule sumDemands	50	4897.0	0.0	1.000	79.8	0.00
multi depot	50	5749.3	0.0	1.000	102.9	0.00
single depot single vehicule sumDemands	100	6981.9	0.0	1.000	170.0	0.00
multi depot	100	8058.6	0.0	1.000	169.2	0.00
single depot single vehicule sumDemands	200	11417.8	0.0	1.000	373.9	0.00
multi depot	200	11602.8	0.0	1.000	314.2	0.00
single depot single vehicule sumDemands	500	16554.8	0.0	1.000	1270.4	0.00
multi depot	500	17676.2	0.0	1.000	1445.1	0.00
single depot single vehicule sumDemands	1000	25995.4	0.0	1.000	4647.9	0.00
multi depot	1000	25879.7	0.0	1.000	4544.5	0.00
depots equal city	10	3966.1	3.3	0.667	185.9	0.00
single depot	10	4067.6	3.3	0.667	193.6	0.00
depots equal city	20	8156.1	0.0	1.000	479.8	0.00
single depot	20	7661.3	0.0	1.000	489.9	0.00
depots equal city	50	13918.7	0.0	1.000	719.3	0.00
single depot	50	14269.3	0.7	0.667	654.4	0.00
depots equal city	100	39031.2	6.0	0.000	2013.6	0.00
single depot	100	30820.4	1.0	0.333	1998.3	0.00
depots equal city	200	90028.5	10.1	0.000	2662.6	0.00
single depot	200	55596.2	1.0	0.000	3014.1	0.00
depots equal city	500	176001.3	8.1	0.000	13851.1	0.00
single depot	500	118726.0	2.2	0.000	11822.7	0.00
depots equal city	1000	244953.3	6.2	0.000	50402.1	0.00
single depot	1000	187945.6	2.7	0.000	42673.2	0.00

Table 3: ACO - Detailed Performance Breakdown.

Configuration	Size	Cost	CVR	Feas	Runtime	TW Violations
single depot single vehicule sumDemands	10	2183.6	0.0	1.000	14.3	0.00
multi depot	10	2325.4	0.0	1.000	11.9	0.00
single depot single vehicule sumDemands	20	3725.9	0.0	1.000	34.6	0.00
multi depot	20	3644.2	0.0	1.000	31.4	0.00
single depot single vehicule sumDemands	50	4840.5	0.0	1.000	165.2	0.00
multi depot	50	5626.2	0.0	1.000	179.5	0.00
single depot single vehicule sumDemands	100	6840.4	0.0	1.000	698.1	0.00
multi depot	100	7868.4	0.0	1.000	678.2	0.00
single depot single vehicule sumDemands	200	11264.3	0.0	1.000	2295.7	0.00
multi depot	200	11473.0	0.0	1.000	2380.3	0.00
single depot single vehicule sumDemands	500	16389.2	0.0	1.000	15573.5	0.00
multi depot	500	17551.6	0.0	1.000	16468.6	0.00
single depot single vehicule sumDemands	1000	25840.7	0.0	1.000	58364.4	0.00
multi depot	1000	25815.8	0.0	1.000	59341.2	0.00
depots equal city	10	3931.6	3.3	0.667	9.4	0.00
single depot	10	3819.2	3.3	0.667	9.6	0.00
depots equal city	20	7714.2	0.0	1.000	34.2	0.00
single depot	20	7749.4	0.0	1.000	34.1	0.00
depots equal city	50	13535.4	0.0	1.000	166.9	0.00
single depot	50	13872.4	0.7	0.667	143.6	0.00
depots equal city	100	37800.2	6.0	0.000	629.4	0.00
single depot	100	30389.5	1.0	0.333	679.0	0.00
depots equal city	200	89937.2	10.1	0.000	2556.8	0.00
single depot	200	55401.8	1.0	0.000	2327.0	0.00
depots equal city	500	175711.1	7.7	0.000	15299.3	0.00
single depot	500	118280.2	2.2	0.000	14781.5	0.00
depots equal city	1000	244999.0	6.1	0.000	70932.6	0.00
single depot	1000	187332.2	2.8	0.000	54846.8	0.00

Table 4: OR-Tools - Detailed Performance Breakdown.

Configuration	Size	Cost	CVR	Feas	Runtime	TW Violations
single depot single vehicule sumDemands	10	2049.2	0.0	1.000	1037.9	0.00
multi depot	10	2167.6	0.0	1.000	1003.3	0.00
single depot single vehicule sumDemands	20	3238.9	0.0	1.000	999.5	0.00
multi depot	20	3142.2	0.0	1.000	1002.6	0.00
single depot single vehicule sumDemands	50	3773.4	0.0	1.000	1015.9	0.00
multi depot	50	4714.2	0.0	1.000	1015.9	0.00
single depot single vehicule sumDemands	100	6283.5	0.0	1.000	1046.5	0.00
multi depot	100	6250.4	0.0	1.000	1048.8	0.00
single depot single vehicule sumDemands	200	9198.8	0.0	1.000	1174.7	0.00
multi depot	200	8956.2	0.0	1.000	1185.4	0.00
single depot single vehicule sumDemands	500	15677.5	0.0	1.000	2129.5	0.00
multi depot	500	15883.2	0.0	1.000	2085.2	0.00
single depot single vehicule sumDemands	1000	25844.3	0.0	1.000	8412.4	0.00
multi depot	1000	25816.3	0.0	1.000	9434.5	0.00
depots equal city	10	4564.7	3.3	0.967	12.6	0.00
single depot	10	4359.0	3.3	0.967	3.6	0.00
depots equal city	20	8192.3	0.0	1.000	8.2	0.00
single depot	20	8346.9	0.0	1.000	7.2	0.00
depots equal city	50	13666.7	0.0	1.000	30.3	0.00
single depot	50	13882.3	0.7	0.993	27.7	0.00
depots equal city	100	38704.1	6.0	0.940	108.6	0.00
single depot	100	30389.3	1.0	0.990	87.8	0.00
depots equal city	200	89937.5	10.1	0.899	345.3	0.00
single depot	200	55401.8	1.0	0.990	329.8	0.00
depots equal city	500	175711.4	7.7	0.923	2010.0	0.00
single depot	500	118279.4	2.2	0.978	2020.5	0.00
depots equal city	1000	244998.0	6.1	0.939	8273.1	0.00
single depot	1000	187830.1	2.7	0.973	8464.4	0.00

Table 5: RL Algorithms – Detailed Performance on CVRP (runtimes in ms).

Solver	Configuration	Size	Cost	CVR	Feas	Runtime (ms)	TW Violations
Attention	single depot single vehicule sumDemands	10	2364.12	0.00	1.000	0.365	0.00
POMO	single depot single vehicule sumDemands	10	2312.68	0.00	1.000	0.282	0.00
Attention	single depot single vehicule sumDemands	20	3222.68	0.00	1.000	0.269	0.00
POMO	single depot single vehicule sumDemands	20	3341.56	0.00	1.000	0.279	0.00
Attention	single depot single vehicule sumDemands	50	5803.63	0.00	1.000	0.304	0.00
POMO	single depot single vehicule sumDemands	50	5920.19	0.00	1.000	0.287	0.00
Attention	single depot single vehicule sumDemands	100	8553.26	0.00	1.000	0.319	0.00
POMO	single depot single vehicule sumDemands	100	16983.50	0.00	1.000	0.319	0.00
Attention	single depot single vehicule sumDemands	200	13228.84	0.00	1.000	0.353	0.00
POMO	single depot single vehicule sumDemands	200	12726.96	0.00	1.000	0.360	0.00
Attention	single depot single vehicule sumDemands	500	22496.94	0.00	1.000	0.463	0.00
POMO	single depot single vehicule sumDemands	500	88789.44	0.00	1.000	0.506	0.00
Attention	single depot single vehicule sumDemands	1000	37430.47	0.00	1.000	0.649	0.00
POMO	single depot single vehicule sumDemands	1000	184656.10	0.00	1.000	0.689	0.00

Table 6: RL Algorithms – Detailed Performance on TWVRP (runtimes in ms).

Solver	Configuration	Size	Cost	CVR	Feas	Runtime (ms)	TW Violations
Attention	single depot	10	3 940.38	0.00	1.000	0.916	0.00
POMO	single depot	10	3 854.6	0.00	1.000	0.707	0.00
Attention	single depot	20	6 504.73	0.00	1.000	1.780	0.00
POMO	single depot	20	6 744.7	0.00	1.000	1.841	0.00
Attention	single depot	50	29 132.94	0.00	1.000	0.731	0.00
POMO	single depot	50	29 718.0	0.00	1.000	0.689	0.00
Attention	single depot	100	57 778.84	0.00	1.000	0.864	0.00
POMO	single depot	100	114 726.7	0.00	1.000	0.864	0.00
Attention	single depot	200	113 742.27	0.00	1.000	0.868	0.00
POMO	single depot	200	109 427.1	0.00	1.000	0.886	0.00
Attention	single depot	500	271 201.60	0.00	1.000	1.412	0.00
POMO	single depot	500	438 502.6	0.00	1.000	1.412	0.00
Attention	single depot	1000	531 470.88	0.00	1.000	1.638	0.00
POMO	single depot	1000	611 307.8	0.00	1.000	1.672	0.00

46 C.1 Qualitative Results

47 As shown in figures 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10, we qualitatively observe that for CVRP instances
 48 with a small number of customers, both Attention and POMO models, as well as classical methods
 49 (ACO, NN2OPT, and OR-Tools), generate highly structured and near-optimal routes. As the number
 50 of customers increases, route complexity grows, making it harder for models to preserve efficiency
 51 and structure. For TWVRP, the models' priority shifts toward satisfying delivery time windows, often
 52 at the expense of distance optimization. This results in routes that appear less spatially coherent but

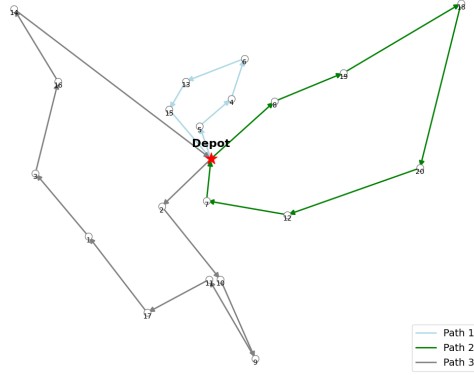


Figure 1: CVRP 20 customers – Attention Model

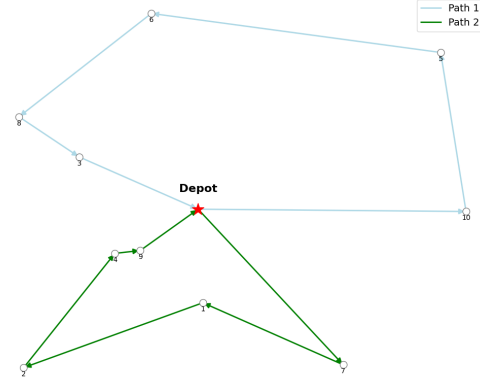


Figure 2: CVRP 10 customers – POMO

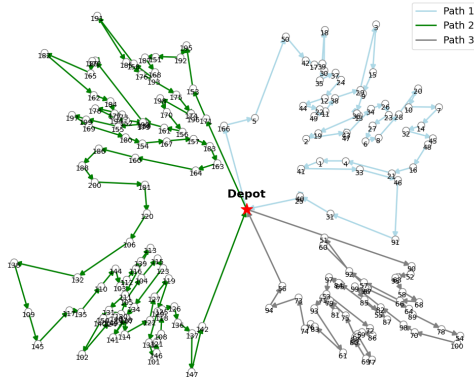


Figure 3: CVRP 200 customers – Attention Model

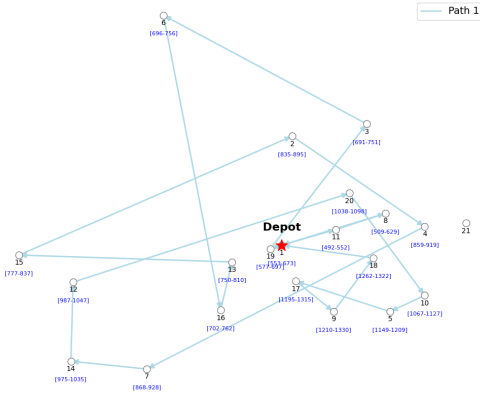


Figure 4: TWVRP 20 customers – Attention Model

53

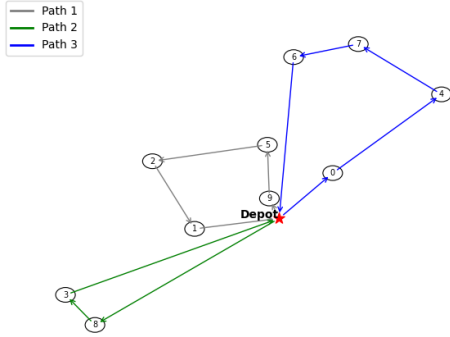


Figure 5: CVRP 10 customers – ACO

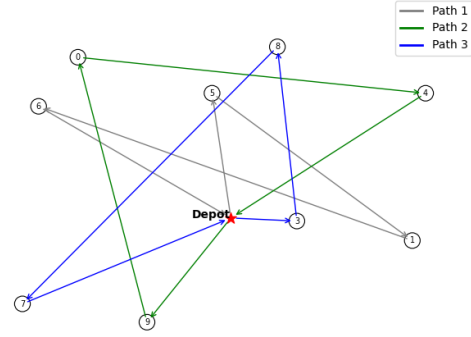


Figure 6: TWVRP 10 customers – ACO

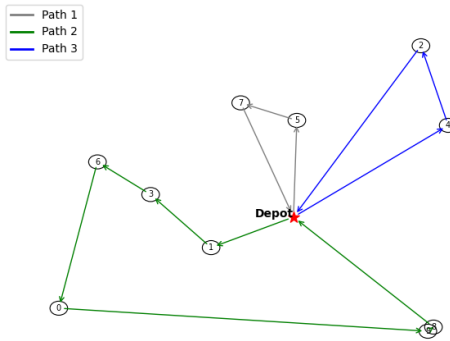


Figure 7: CVRP 10 customers – NN2OPT

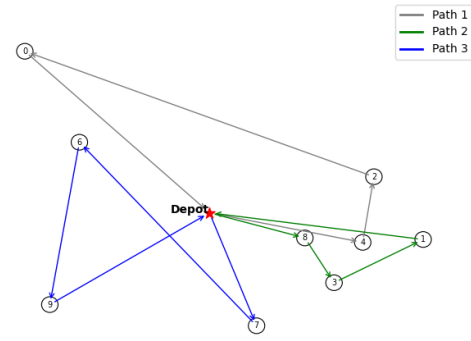


Figure 8: TWVRP 10 customers – NN2OPT

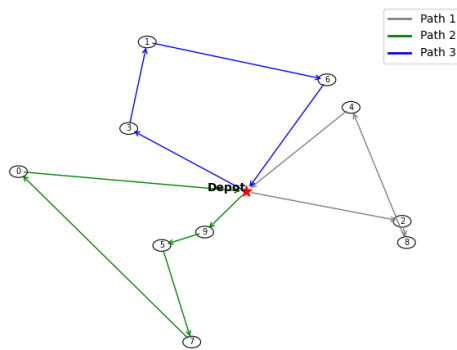


Figure 9: CVRP 10 customers – OR-Tools

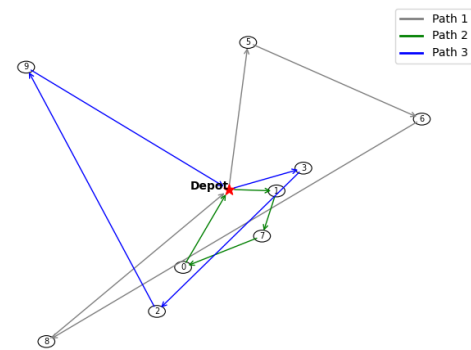


Figure 10: TWVRP 10 customers – OR-Tools

54 D Reinforcement Learning

55 D.1 Problem Formulation

56 We model both the Capacitated Vehicle Routing Problem (CVRP) and Vehicle Routing Problem
57 with Time Windows (VRPTW) as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where
58 each state $s_t \in \mathcal{S}$ encodes the vehicle’s current position, remaining capacity, visited set (and only
59 for VRPTW the current time and per-customer time windows $[e_i, \ell_i]$). Actions $a_t \in \mathcal{A}(s_t)$ select the
60 next customer, and transitions $P(s_{t+1} | s_t, a_t)$ deterministically update the tour while, in VRPTW,
61 adding stochastic delays.

62 The reward is $r(s_t, a_t) = -d_{i,j} - \tau [t_{\text{arrive}} > \ell_i]$ when visiting customer j , with $d_{i,j}$ the Euclidean
63 distance and τ a large penalty for time-window violations, and zero upon return to the depot. We
64 follow a constructive, autoregressive decoding: at each step we append one customer until all are
65 visited.

66 D.2 Policy

67 We adopt the encoder–decoder with multi-head attention of Kool [3]. Given embedded node features
68 $\mathbf{x}_i \in \mathbb{R}^d$, each of the L encoder layers applies multi-head self-attention. At step t , with context
69 embedding \mathbf{h}_t , we score each remaining node j by $u_{t,j} = \mathbf{v}^\top \tanh(W_1 \mathbf{h}_t + W_2 \mathbf{x}_j)$ and define
70 $\pi_\theta(a_t = j | s_t) = \exp(u_{t,j}) / \sum_{k \notin \mathcal{V}_t} \exp(u_{t,k})$.

71 We optimize the policy by maximizing the expected return $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$ using two con-
72 structive, autoregressive policy-gradient methods. A constructive policy builds a complete solution
73 by sequentially selecting one customer at a time until the tour is finished, while an autoregressive
74 policy conditions each action on the history of previous choices, enabling the network to capture
75 dependencies across steps.

76 We first apply REINFORCE [5], which updates parameters via $\nabla_\theta J(\theta) = \mathbb{E}[\sum_t \nabla_\theta \log \pi_\theta(a_t |$
77 $s_t) (R(\tau) - b(s_t))]$, where $b(s_t)$ is a rollout baseline obtained by greedy decoding; then POMO [4]
78 samples K different start nodes per instance, computes returns R_k and a shared baseline $\bar{R} =$
79 $\frac{1}{K} \sum_k R_k$, and applies $\nabla_\theta J(\theta) = \frac{1}{K} \sum_{k=1}^K \nabla_\theta \log \pi_\theta(\tau_k) (R_k - \bar{R})$. REINFORCE offers simplicity
80 and unbiased gradients, while POMO’s shared baseline exploits VRP permutation symmetry for
81 variance reduction; together they provide a strong comparison between a classical Monte Carlo
82 approach and a state-of-the-art, variance-reduced VRP-specific algorithm.

83 D.3 Training Details

84 All models were implemented in the RL4CO framework and trained end-to-end with Adam at a
85 learning rate of 10^{-4} . For CVRP with REINFORCE we used a batch size of 512 and generated
86 100 000 synthetic instances on the fly; for VRPTW with POMO we used batch size 64 and 1 000 000
87 instances. Validation employed greedy decoding under nominal travel-time conditions. VRPTW
88 environments included log-normal delays calibrated to traffic data, Gaussian time-of-day kernels, and
89 Poisson accident events, with infeasible actions heavily penalized to enforce time windows.

90 D.4 Evaluation on SVRPBench

91 After training, we converted each of the 500+ SVRPBench instances into the RL4CO environment
92 format and ran the trained policies in greedy mode, selecting at each step $a_t = \arg \max_j \pi_\theta(a_t =$
93 $j | s_t)$. To assess robustness, we then simulated each resulting tour under multiple sampled delay
94 realizations and reported average tour length and feasibility rates. Despite domain shift, attention-
95 based RL policies maintained high feasibility and near-optimal costs across all problem sizes.

96 **References**

- 97 [1] Yossiri Adulyasak and Patrick Jaillet. Models and algorithms for stochastic and robust vehicle
98 routing with deadlines. *Transportation Science*, 50(2):608–626, 2016.
- 99 [2] K. Chepuri and T. Homem-de Mello. Solving the vehicle routing problem with stochastic
100 demands using the cross-entropy method. *Annals of Operations Research*, 134(1):153–181,
101 2005.
- 102 [3] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems!
103 *International Conference on Learning Representations (ICLR)*, 2019.
- 104 [4] Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai
105 Min. Pomo: Policy optimization with multiple optima for reinforcement learning. *Advances in*
106 *Neural Information Processing Systems*, 33:21188–21198, 2020.
- 107 [5] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforce-
108 ment learning. *Machine learning*, 8(3):229–256, 1992.