Multi-Task Learning for Vehicle Routing Problems via Preference Learning and Local Refinement

Arthur Corrêa, Paulo Nascimento, and Samuel Moniz
Department of Mechanical Engineering
University of Coimbra, CEMMPRE, ARISE
Coimbra, Portugal
ajpcorrea@dem.uc.pt, pnascimento@dem.uc.pt, samuel.moniz@dem.uc.pt

***Abstract*—Managing modern logistics systems requires solving complex vehicle routing problems (VRPs) subject to diverse operational constraints, giving rise to multiple different variants. Traditional exact algorithms and meta-heuristics struggle to balance computational efficiency with the flexibility needed to handle this heterogeneity. Multi-task learning (MTL) is therefore gaining prominence, enabling a single model to address many VRP variants simultaneously. Although reinforcement learning remains the standard training paradigm, it suffers from vanishing advantages and task interference due to mismatched reward scales across problem structures. To deal with this, we propose POLAR (Preference Optimization with Local Augmented Refinement), a novel algorithm for MTL VRP models that replaces scalar rewards with preference optimization over relative pairwise solution comparisons, thereby eliminating scale discrepancies across heterogeneous tasks. To mitigate learning stagnation from local minima, POLAR further applies an *a posteriori* local search refinement that augments self-generated solutions and yields more informative training signals. Experimental results indicate that POLAR vastly outperforms the existing state-of-the-art across 16 VRP variants.**

***Index Terms*—vehicle routing, multi-task learning, preference optimization, reinforcement learning, local search**

I. Introduction

The Vehicle Routing Problem (VRP) is a central combinatorial optimization problem in modern logistics decision support. Exact methods guarantee optimality but scale poorly beyond roughly one hundred nodes, whereas meta-heuristics scale better but demand substantial expert tuning. In response, neural constructive solvers have recently emerged as fast alternatives [1]. For decision-support systems, however, one of the many practical challenges is not to solve a single VRP formulation in isolation, but to support *many* different variants within a single deployable solver [2].

Unified Multi-Task Learning (MTL) models [3], [5] address this heterogeneity by allowing a single model to learn shared representations across different tasks. However, despite recent progress, a notable limitation remains. Currently, training is still dominated by Reinforcement Learning (RL), more specifically, the REINFORCE with shared baselines algorithm [1], which has long been the standard in the literature. While effective, RL suffers from two drawbacks: (i) as policies improve, the advantage gap between sampled solutions and baselines shrinks, slowing convergence due to increasingly less informative advantages; (ii) different VRP variants present varying tour-cost scales, which can cause uneven gradient magnitudes and make one task improve at the expense of another [7]. Preference Optimization (PO) [8] offers an attractive compromise by learning from relative tour rankings, making supervision insensitive to heterogeneous cost scales. Yet, PO is only as informative as the tours being compared. When decoded solutions become increasingly similar, pairwise preferences carry small cost margins and learning can stagnate for reasons similar to the advantage collapse in RL.

These observations highlight a critical opportunity: improving the quality of candidate solutions used for preference learning. If policy-generated solutions could be efficiently refined in an *a posteriori* manner (without distorting the policy distribution), the resulting higher-quality pairwise comparisons could provide stronger learning signals. Such an approach has the potential to navigate toward more promising regions of the solution space by escaping local minima, narrowing the performance gap between neural methods and traditional optimization algorithms. We therefore propose **POLAR**, a novel training framework for cross-problem VRPs that combines PO with a selective Local Search (LS) algorithm.

II. Problem Definition

Following [5], we consider 16 VRP variants built from a classic Capacitated VRP (CVRP) formulation. Four additional constraints and their combinations—backhauls, open routes, distance limits, and time windows—define the variant set. We train and evaluate on instances with $n \in \{50, 100\}$ customers; for each variant and each instance size n, inference uses a fixed held-out test set of 1,000 instances. A feasible solution τ is a set of vehicle routes minimizing total travel cost subject to the active constraints. Solution quality is reported as the mean performance gap (%) relative to HGS-PyVRP [9], averaged over these test instances, together with the mean wall-clock time necessary to solve all 1,000 instances.

III. The POLAR Algorithm

POLAR combines PO with an LS procedure implemented through PyVRP's `LocalSearch` class [9]. Each epoch samples 100,000 instances in batches of size B. For each instance $\mathcal{G}_i$, the decoder generates N POMO trajectories [1] plus one greedy trajectory; the best tour undergoes a single LS iteration, is re-scored under policy p_θ, and enters the preference set.

TABLE I
MEAN GAP AND INFERENCE TIME (PER 1,000 INSTANCES), AVERAGED OVER 16 VARIANTS. BEST NEURAL GAP IS SHOWN IN **BOLD**.

	$n = 50$		$n = 100$	
Method	Gap (%)	Time	Gap (%)	Time
HGS-PyVRP	*	10.4m	*	20.8m
MTPOMO	2.448	2.2s	3.966	8.6s
MVMoE	2.289	3.2s	3.682	11.6s
RouteFinder	1.974	2.0s	3.131	8.4s
CaDA	1.714	2.0s	2.814	8.4s
CaDA†	1.665	3.6s	2.622	15.5s
POLAR	**1.130**	3.5s	**2.090**	15.0s

Let $\{\tau_i^j\}_{j=0}^N$ denote the decoded tours for instance $\mathcal{G}_i$, with rewards $\{r_i^j\}_{j=0}^N$ and log-likelihoods $\{\log p_\theta(\tau_i^j)\}_{j=0}^N$. A Boolean label $y_{j,k}^i$ indicates whether τ_i^j is preferred over τ_i^k. The PO objective for a batch with B instances is:

$$\mathcal{L}_{\text{PO}} = -\frac{1}{B(N+1)^2}\sum_{i=1}^{B}\sum_{j=0}^{N}\sum_{k=0}^{N} y_{j,k}^i \log\sigma\big(\alpha(\log p_\theta(\tau_i^j) - \log p_\theta(\tau_i^k))\big), \tag{1}$$

where α is a temperature parameter and σ the sigmoid. The policy is updated via gradient ascent on $\mathcal{L}_{\text{PO}}$.

Conceptually, POLAR is built around five design principles. First, it replaces scalar REINFORCE updates with denser $O(N^2)$ pairwise supervision that is inherently scale-invariant across heterogeneous VRP variants. Second, the LS is activated only in the final 50 of 300 epochs, when decoded tours are already competitive and refinement yields large marginal benefits. Third, each LS pass is restricted to a single iteration to limit distribution shifts between refined tours and the current policy. Fourth, refinement is offloaded to the CPU and parallelized across the batch, keeping the additional wall-clock overhead modest. Fifth, POLAR is architecture-agnostic, requiring only an autoregressive decoder and tour re-scoring. Additionally, POLAR's gains are largest once decoded tours are already competitive, but pairwise margins shrink.

IV. EXPERIMENTS AND RESULTS

Settings. We adopt the CaDA† architecture (CaDA encoder [6] with ReLD decoder [10]) and train it with POLAR for 300 epochs. We compare against HGS-PyVRP [9], MTPOMO [3], MVMoE [4], RouteFinder [5], CaDA [6], and CaDA†. All baselines were trained with RL, following the same settings as in [5]. Inference was performed using the POMO multiple-starting-node strategy and $\times 8$ augmentation [1]. HGS-PyVRP is run with a 10 s (n=50) and 20 s (n=100) limit per instance, parallelized over 16 CPU cores [5]. Table I shows the average results over all 16 variants.

Overall, HGS-PyVRP displays the best performance, since it is a meta-heuristic relying on advanced exploration mechanisms and substantially higher computational cost. Among neural methods, POLAR achieves the lowest mean gap across all 16 variants on both instance sizes, clearly improving over the remaining MTL baselines.

We also isolated the contribution of the training algorithm by retraining MTPOMO, MVMoE and RouteFinder with RL, PO at n=50 (Table II). Overall, POLAR yields the lowest mean gap on every backbone, confirming that gains stem from the training algorithm rather than a specific architecture.

TABLE II
ALGORITHM ABLATION ON n=50. MEAN PERFORMANCE GAP (%) RELATIVE TO HGS-PYVRP OVER 16 VARIANTS. BEST PER ROW IN **BOLD**.

Backbone	RL	PO	POLAR
MTPOMO	2.340	2.250	**2.120**
MVMoE	2.190	2.070	**1.940**
RouteFinder	2.043	1.914	**1.858**

Overhead and scalability. On an NVIDIA L40S, PO-only epochs averaged around **3m 15s** for n=50 and **5m 45s** for n=100. In turn, LS-active epochs averaged **5m 45s** and **10m 3s**, respectively. Since refinement is deferred to the last 50 epochs, full POLAR wall-clock training time is ∼18 h and ∼32 h compared to ∼16 h and ∼29 h for PO alone, a negligible increase of ∼**13%** for the reported generalization gains.

V. CONCLUSION

POLAR integrates preference learning and training-time local refinement into a scalable MTL recipe. By refining the best tour per instance during training, it provides denser, scale-invariant supervision without variant-specific reward engineering. Experiments on 16 variants show state-of-the-art performance, with ablations confirming consistent gains across architectures. Future work will focus on mitigating gradient interference and negative transfer.

REFERENCES

[1] Y.-D. Kwon, J. Choo, B. Kim, I. Yoon, Y. Gwon, and S. Min, "POMO: Policy optimization with multiple optima for reinforcement learning," in *Advances in Neural Information Processing Systems*, 2020.

[2] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, vol. 99, pp. 300–313, 2016.

[3] F. Liu, X. Lin, Z. Wang, Q. Zhang, T. Xialiang, and M. Yuan, "Multi-task learning for routing problem with cross-problem zero-shot generalization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2024.

[4] J. Zhou, Z. Cao, Y. Wu, W. Song, Y. Ma, J. Zhang, and C. Xu, "MVMoE: Multi-task vehicle routing solver with mixture-of-experts," *Proceedings of Machine Learning Research*, pp. 61 804–61 824, 2024.

[5] F. Berto, C. Hua, N. G. Zepeda, A. Hottung, N. Wouda, L. Lan, J. Park, K. Tierney, and J. Park, "RouteFinder: Towards foundation models for vehicle routing problems," *Transactions on Machine Learning Research*, 2025.

[6] H. Li, F. Liu, Z. Zheng, Y. Zhang, and Z. Wang, "CaDA: Cross-problem routing solver with constraint-aware dual-attention," in *Forty-second International Conference on Machine Learning*, 2025.

[7] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt, "Multi-task deep reinforcement learning with PopArt," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[8] M. Pan, G. Lin, Y.-W. Luo, B. Zhu, Z. Dai, L. Sun, and C. Yuan, "Preference optimization for combinatorial optimization problems," in *Forty-second International Conference on Machine Learning*, 2025.

[9] N. A. Wouda, L. Lan, and W. Kool, "PyVRP: a high-performance VRP solver package," *INFORMS Journal on Computing*, vol. 36, no. 4, pp. 943–955, 2024.

[10] Z. Huang, J. Zhou, Z. Cao, and Y. XU, "Rethinking light decoder-based solvers for vehicle routing problems," in *The Thirteenth International Conference on Learning Representations*, 2025.