A Foundation Model for Vehicle Routing Problems

Federico Berto[a,*], Chuanbo Hua[a,*], Nayeli Gast Zepeda[b,*], André Hottung[b], Niels Wouda[c], Leon Lan[d], Junyoung Park[a], Kevin Tierney[b], Jinkyoo Park[a,e]

[a] KAIST, South Korea, [b] Bielefeld University, Germany, [c] Rotterdam School of Management, Netherlands, [d] VU Amsterdam, Netherlands, [e] Omelet, South Korea,
[*] Equal contributions. Correspondence to: fberto@kaist.ac.kr

Extended abstract submitted for presentation at the 12[th] Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

March 12, 2025

Keywords: foundation models, vehicle routing problem, reinforcement learning, neural combinatorial optimization

1 INTRODUCTION

Vehicle Routing Problems (VRPs) are a fundamental class of combinatorial optimization problems with significant applications in logistics and transportation. Traditional methods from operations research (OR) have developed sophisticated heuristics and metaheuristics to tackle VRPs, but these often require expert knowledge and manual tuning to adapt to different VRP variants. Moreover, such methods usually suffer from scalability issues at larger scales for complex variants where good heuristics are unavailable.

Recently, Neural Combinatorial Optimization (NCO) approaches have emerged, leveraging advances in deep learning to learn solvers for VRPs without requiring costly labeled datasets by leveraging reinforcement learning (Kool *et al.*, 2019, Hottung *et al.*, 2024). NCO allows to automatically discover heuristics by leveraging neural networks, unlike hand-crafted rules of traditional approaches. However, most existing NCO methods focus on specific VRP variants and generally lack the ability to generalize across multiple variants.

In this work, we introduce *RouteFinder*, a comprehensive foundation model framework for solving a wide range of VRP variants. We treat different VRP variants as combinations of fundamental attributes within one unified environment. By leveraging a modern transformer-based architecture akin to advances in large language models and reinforcement learning techniques, RouteFinder effectively learns to solve multiple VRP variants within a single foundation model.

Our main contributions are:

- We propose a unified VRP environment capable of representing any combination of VRP attributes, enabling the modeling of multiple VRP variants within a single framework.
- We design a transformer-based model with global attribute embeddings to enhance the representation of VRP variants and improve the model's ability to generalize across tasks.
- We introduce two reinforcement learning techniques: *Mixed Batch Training* and *Multi-Variant Reward Normalization*, to stabilize training across multiple VRP variants.
- We present *Efficient Adapter Layers*, allowing for efficient fine-tuning of the pre-trained RouteFinder model to new VRP variants with unseen attributes.

We evaluate RouteFinder on 24 VRP variants and demonstrate that it achieves competitive results, outperforming existing multi-task learning models for VRPs.

2 METHODOLOGY

2.1 VRPs as Autoregressive Sequence Generation

We learn to construct solutions in a sequence. We start by encoding the problem instance $\boldsymbol{x}$ (e.g., node and global attributes) through a trainable encoder f_θ, yielding an embedding $\boldsymbol{h} = f_\theta(\boldsymbol{x})$. The solution $\boldsymbol{a}$ is then decoded from $\boldsymbol{h}$ with actions based on the current sequence. Each action is chosen by the decoder g_θ, following:

$$a_t \sim g_\theta(a_t | a_{t-1}, ..., a_0, \boldsymbol{h}), \tag{1a}$$

$$\pi_\theta(\boldsymbol{a}|\boldsymbol{x}) \triangleq \prod_{t=1}^{T-1} g_\theta(a_t | a_{t-1}, ..., a_0, \boldsymbol{h}), \tag{1b}$$

where $\boldsymbol{a} = (a_1, ..., a_T)$ is a solution and π_θ maps the problem $\boldsymbol{x}$ to a solution $\boldsymbol{a}$.

2.2 Unified VRP Environment

We construct a unified VRP environment for autoregressive solution construction that can model any combination of VRP attributes, including capacity constraints, time windows, open routes, backhauls, and duration limits. By representing VRP variants as subsets of this generalized problem with different attributes activated or deactivated, we enable the model to learn a wide range of VRP variants within a single environment as shown in Fig. 1. In this environment,

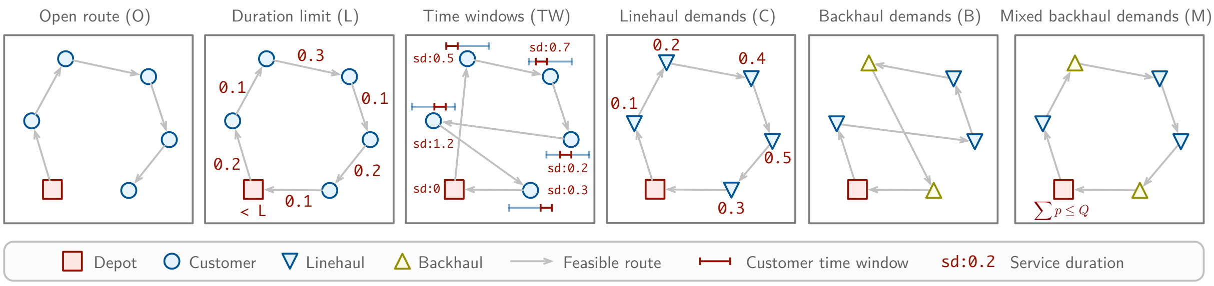


Figure 1 – *VRP variants we consider with respective attributes.*

attributes can be turned "on" or "off" by setting specific parameters. For instance, if a variant does not include time windows, we set the time windows of all customers to $[0, \infty]$. This flexibility allows us to model up to 2^n (with n number of attributes) different VRP variants.

2.3 Transformer-Based Model with Global Attribute Embeddings

We design a transformer-based model architecture inspired by recent advances in large language models (Dubey *et al.*, 2024). The encoder processes the input VRP instance, including node features (e.g., customer locations, demands) and global attributes (e.g., whether the problem includes time windows). To enhance the model's ability to differentiate between VRP variants, we introduce *Global Attribute Embeddings*. These embeddings encode the global attributes of the VRP instance and are integrated into the transformer's encoding process, allowing the model to better understand and adapt to different VRP variants.

2.4 Reinforcement Learning Techniques

Training a model to solve multiple VRP variants presents challenges due to the diversity of tasks and reward scales. We propose two reinforcement learning techniques to address these challenges. Firstly, *Mixed Batch Training* makes it possible to sample problem instances with different combinations of attributes within each training batch. This encourages the model to

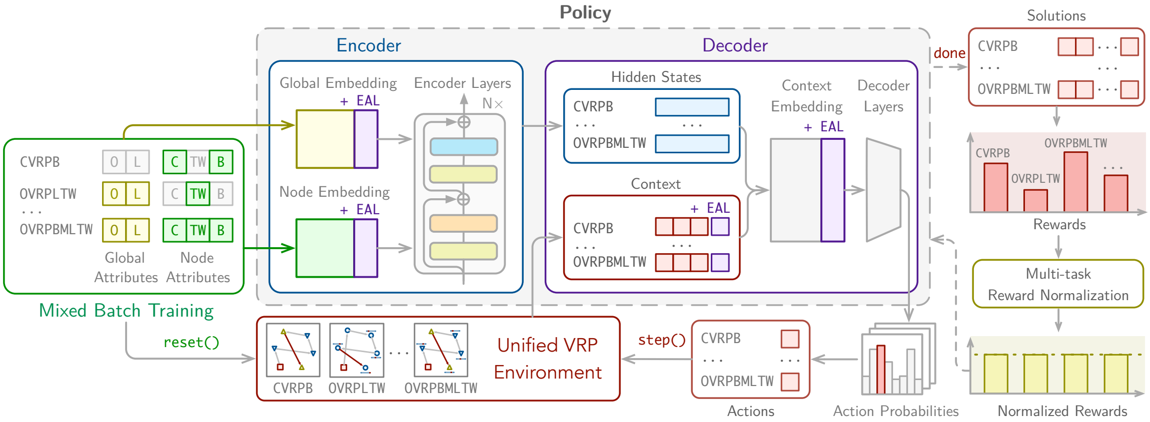


Figure 2 – *Overview of RouteFinder.*

generalize across variants and prevents bias towards specific tasks. Furthermore, *Multi-Variant Reward Normalization* alleviates the different reward scales of diverse variants by adjusting the rewards for each variant, ensuring that the learning signal is balanced across tasks. The final training objective becomes:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\xi(\boldsymbol{a}^i, \boldsymbol{x}) - \frac{1}{N} \sum_{j=1}^{N} \xi(\boldsymbol{a}^j, \boldsymbol{x}) \right) \nabla_\theta \log p_\theta(\boldsymbol{a}^i | \boldsymbol{x}), \tag{2}$$

where $\xi(\boldsymbol{a}, \boldsymbol{x})$ is a function calculating the normalized reward for VRP instance $\boldsymbol{x}$ that additionally maps instance $\boldsymbol{x}$ to variant k. We employ the REINFORCE objective with POMO (Kwon *et al.*, 2020) as shared mean baseline to improve convergence, where both the reward and the shared baseline are normalized by ξ to calculate the policy gradients advantage.

2.5 Efficient Adapter Layers for Fine-Tuning

To enable efficient fine-tuning of the pre-trained model to new VRP variants with unseen attributes, we introduce *Efficient Adapter Layers* (EAL). Unlike the recently proposed adapter layers in Lin *et al.* (2024), EALs augment existing projection layers by adding new parameters initialized with zeroes for the unseen attributes while keeping the existing parameters unchanged. This allows the model to incorporate new attributes without affecting the learned representations of existing attributes, facilitating rapid adaptation to new variants.

3 RESULTS

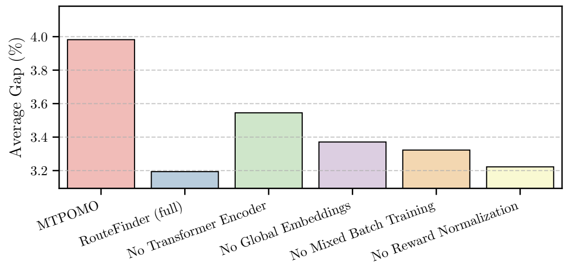
We train RouteFinder on 16 VRP variants, including combinations of attributes like capacity constraints, time windows, open routes, and backhauls. We run all neural models on a single NVIDIA RTX 3090 GPU and OR methods HGS-PyVRP (Wouda *et al.*, 2024) and OR-Tools (Perron & Furnon, 2023) on a single CPU core (parallelized in a batch of 16 cores) and compare results on 1000 test instances for each variant. Our experiments demonstrate the effectiveness of RouteFinder in solving multiple VRP variants within a single model.

Compared to state-of-the-art multi-task learning models like MTPOMO (Liu *et al.*, 2024) and MVMoE (Zhou *et al.*, 2024), RouteFinder achieves superior performance across all VRP variants tested. We observe that RouteFinder reduces the optimality gap by more than 10% compared to these baselines (left of Table 1).

We conduct ablation studies to assess the impact of each component of RouteFinder, which we show in Table 1 (right). Each component, from modeling to training method, has a significant impact on the performance.

Table 1 – *Left: results on 16 VRPs. Right: ablation study.*

	Solver	$n = 50$ Obj.	Gap	Time	$n = 100$ Obj.	Gap	Time
16 VRPs	HGS-PyVRP	11.315	*	10.4m	17.657	*	20.8m
	OR-Tools	11.371	0.682 %	10.4m	18.048	2.404 %	20.8m
	MTPOMO	11.593	2.410 %	1s	18.363	3.873 %	7s
	MVMoE	11.582	2.287 %	2s	18.337	3.682%	9s
	RF-POMO	11.568	2.140 %	1s	18.309	3.565 %	7s
	RF-MoE	11.567	2.161 %	2s	18.286	3.384 %	9s
	RF-TE	11.556	2.063 %	1s	18.236	3.124 %	7s



Finally, we test RouteFinder's ability to adapt to new VRP variants with unseen attributes using EALs. We introduce *mixed backhauls* as a new attribute allowing modeling new problems such as the VRP with mixed backhauls (VRPMB). RouteFinder with EALs rapidly adapts to the new variant with more than 30% improvements compared to adapter layers (Lin *et al.*, 2024).

4 CONCLUSION

We have presented RouteFinder, a foundation model framework for solving multiple VRP variants within a unified environment. By integrating a transformer-based architecture with global attribute embeddings and novel reinforcement learning techniques, RouteFinder effectively learns to solve diverse VRP variants. The introduction of Efficient Adapter Layers enables efficient adaptation to new variants with unseen attributes.

Our work is an early attempt to develop foundation models for VRPs. In future works, we intend to extend RouteFinder to support further variants of the vast VRP literature. We also aim at improving model performance to eventually outperform state-of-the-art OR solvers – exciting directions include decomposition methods and end-to-end (de)construction and improvement.

References

Dubey, Abhimanyu, Jauhri, Abhinav, Pandey, Abhinav, Kadian, Abhishek, Al-Dahle, Ahmad, Letman, Aiesha, Mathur, Akhil, Schelten, Alan, Yang, Amy, Fan, Angela, *et al.* 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783.*

Hottung, André, Mahajan, Mridul, & Tierney, Kevin. 2024. PolyNet: Learning Diverse Solution Strategies for Neural Combinatorial Optimization. *arXiv preprint arXiv:2402.14048.*

Kool, Wouter, Van Hoof, Herke, & Welling, Max. 2019. Attention, learn to solve routing problems! *International Conference on Learning Representations.*

Kwon, Yeong-Dae, Choo, Jinho, Kim, Byoungjip, Yoon, Iljoo, Gwon, Youngjune, & Min, Seungjai. 2020. Pomo: Policy optimization with multiple optima for reinforcement learning. *Advances in Neural Information Processing Systems*, **33**, 21188–21198.

Lin, Zhuoyi, Wu, Yaoxin, Zhou, Bangjian, Cao, Zhiguang, Song, Wen, Zhang, Yingqian, & Jayavelu, Senthilnath. 2024. Cross-Problem Learning for Solving Vehicle Routing Problems. *IJCAI.*

Liu, Fei, Lin, Xi, Zhang, Qingfu, Tong, Xialiang, & Yuan, Mingxuan. 2024. Multi-Task Learning for Routing Problem with Cross-Problem Zero-Shot Generalization. *arXiv preprint arXiv:2402.16891.*

Perron, Laurent, & Furnon, Vincent. 2023. *OR-Tools.* Google.

Wouda, Niels A, Lan, Leon, & Kool, Wouter. 2024. PyVRP: A high-performance VRP solver package. *INFORMS Journal on Computing.*

Zhou, Jianan, Cao, Zhiguang, Wu, Yaoxin, Song, Wen, Ma, Yining, Zhang, Jie, & Xu, Chi. 2024. MVMoE: Multi-Task Vehicle Routing Solver with Mixture-of-Experts. *In: International Conference on Machine Learning.*