ICML
International Conference
On Machine Learning

TRANSACTIONS tmlr on ML RESEARCH

RouteFinder:

Towards Foundation Models for Vehicle Routing Problems

*Federico Berto** · *Chuanbo Hua** · *Nayeli Gast Zepeda** · *André Hottung* · *Niels Wouda* · *Leon Lan* · *Junyoung Park* · *Kevin Tierney* · *Jinkyoo Park*

KAIST · radical numerics · universität wien · UNIVERSITÄT BIELEFELD

Omelet · Applied Routing

Funded by DFG Deutsche Forschungsgemeinschaft German Research Foundation

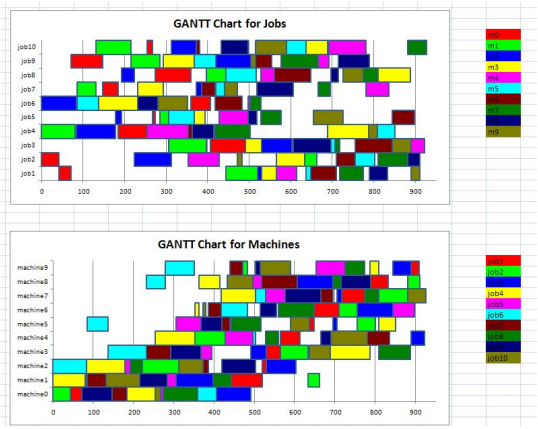
AI4CO

Combinatorial Optimization (CO)

AI4CO

Goal 🎯: finding an optimal set of actions from a finite set of discrete objects

Routing Problems



Scheduling Problems

The logistics industry is worth over 10 Trillion USD! (Statista, 2025)

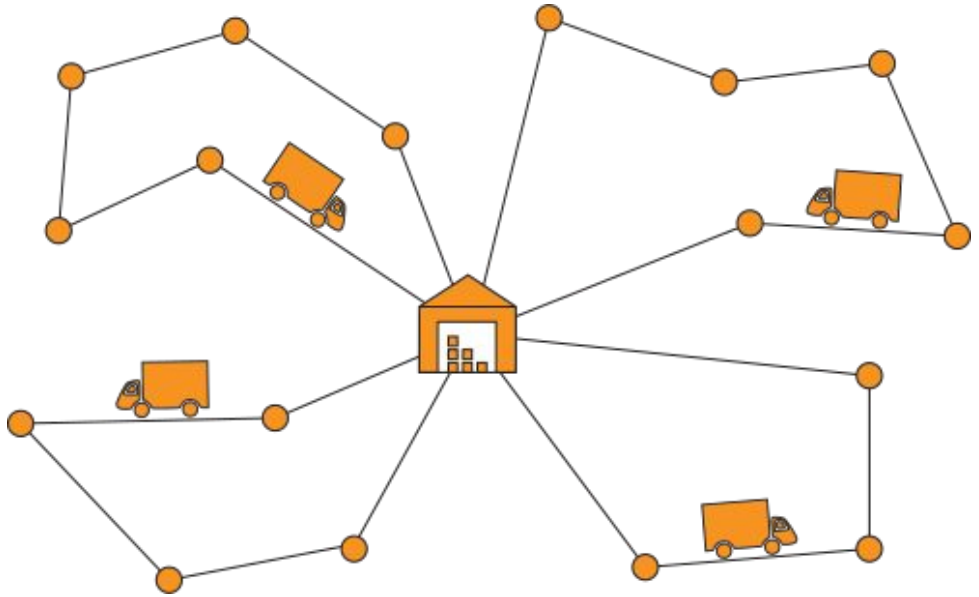
Problem: CO is NP-hard!

Solution: Reinforcement Learning (RL)

➔ Better solutions than traditional methods
➔ Fast and scalable solvers
➔ Less or no reliance on manual design

Vehicle Routing Problems (VRPs)

Vehicle Routing Problems (VRPs) are NP-hard combinatorial problems with impactful applications in the wild



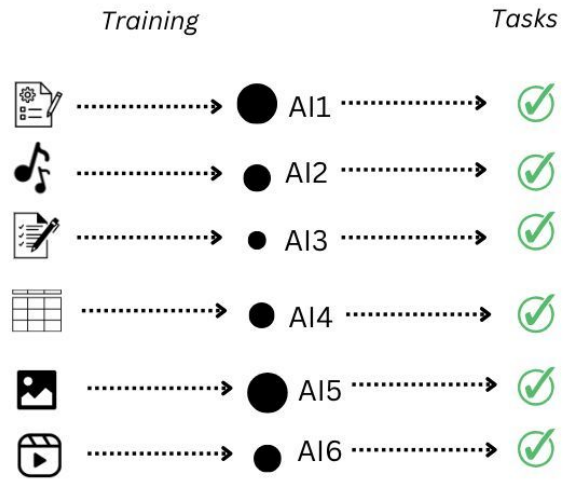
CVRP example from PyVRP

Example: CVRP (Capacitated Vehicle Routing Problem) involves finding the shortest path for vehicles with limited capacities → There are many extensions to this problem!

Motivation: Foundation Models

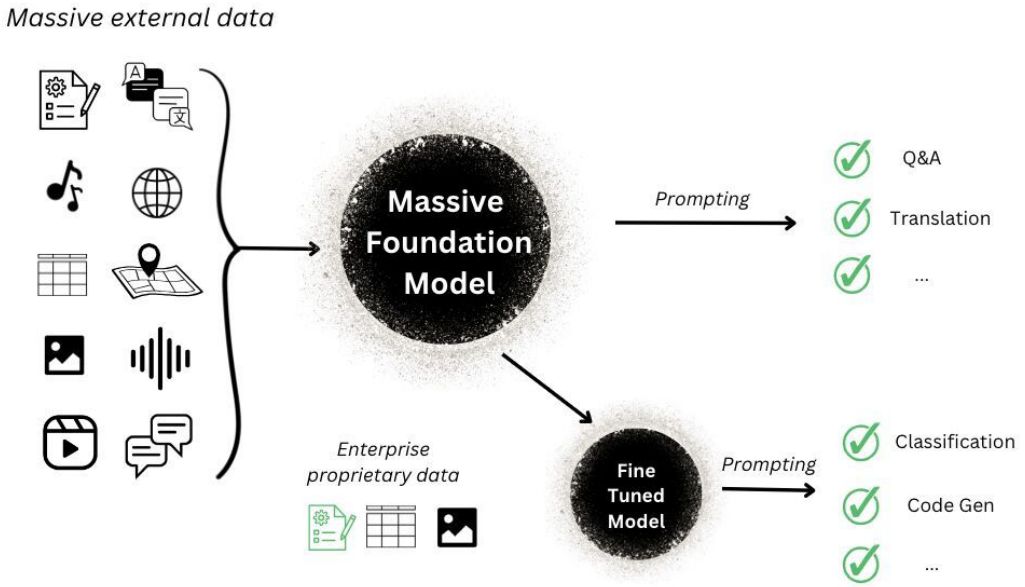
AI4CO

Traditional ML



- Train *K* models for *K* task
- Easy
- (-) Generalization
- (-) Cannot finetune easily to new tasks

Foundation Models



- Train a single model for all tasks
- Needs more data and care
- (+) Superior generalization
- (+) Can efficiently finetune to new tasks

ICML International Conference On Machine Learning

VRP Variants

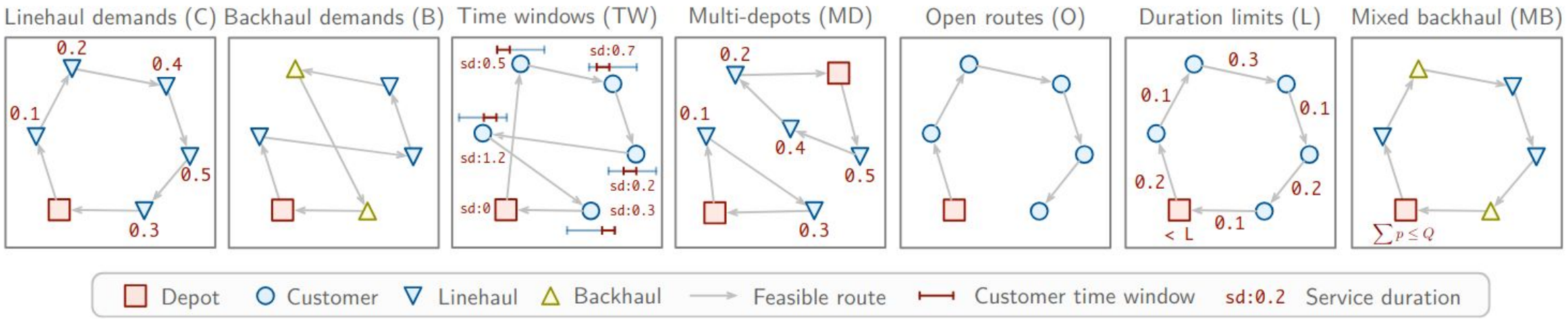
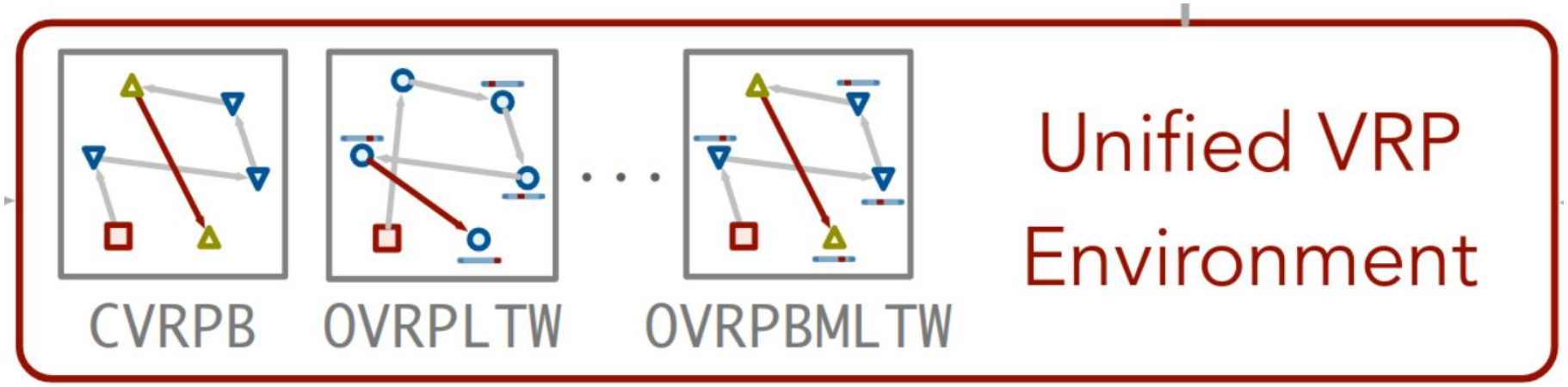


Figure 1: VRP attributes. Linehaul demands (C), backhaul demands (B), time windows (TW), and multi-depot (MD) are *node attributes*, whereas open routes (O), duration limits (L), and mixed backhaul (MB) mode are *global attributes*. Attribute combinations can define new VRP variants.

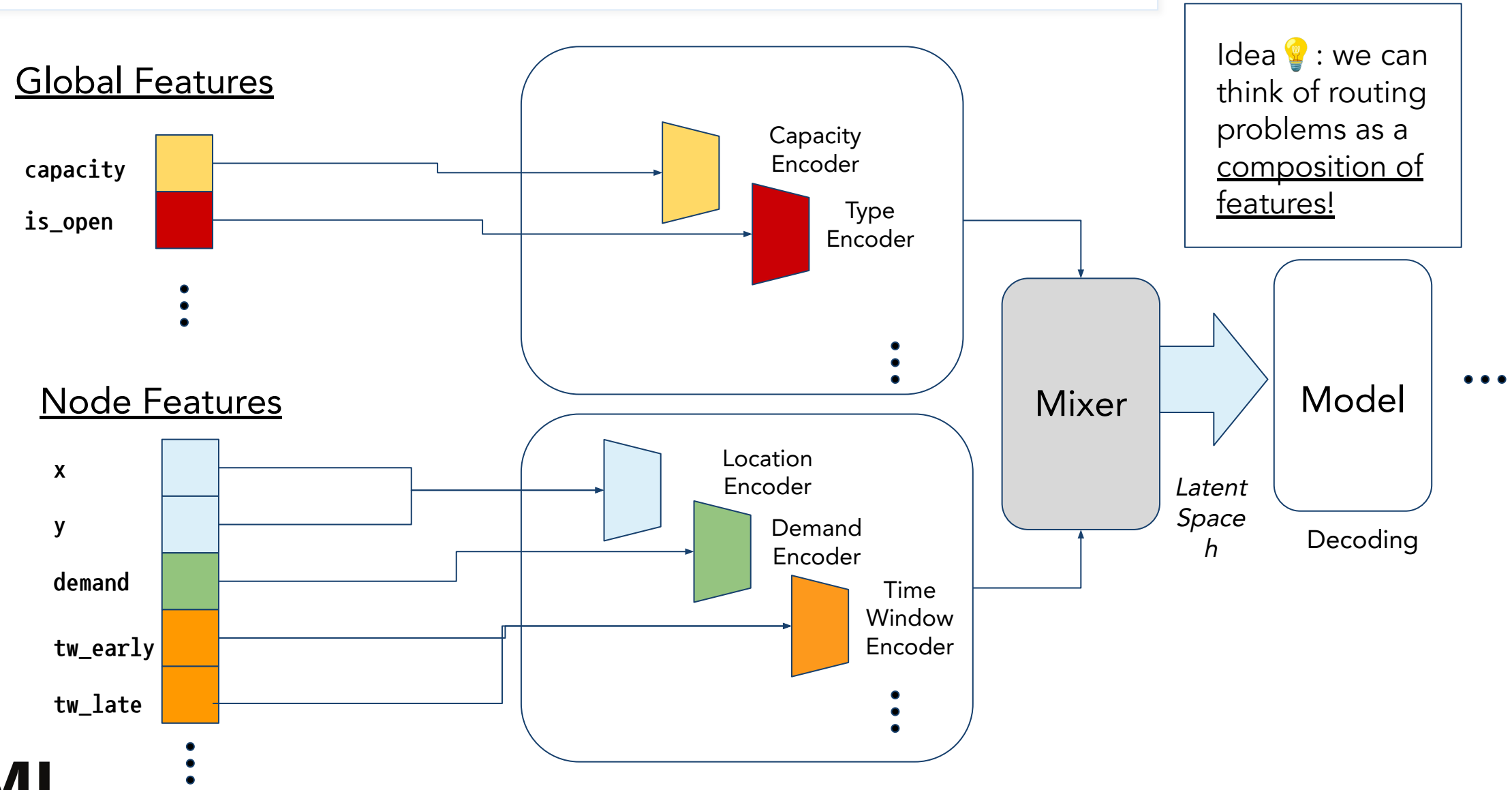
Ingredient 1: Environment



- We create an environment that can handle any number of features
  - With hardware acceleration (GPU)
  - Even on the same batch

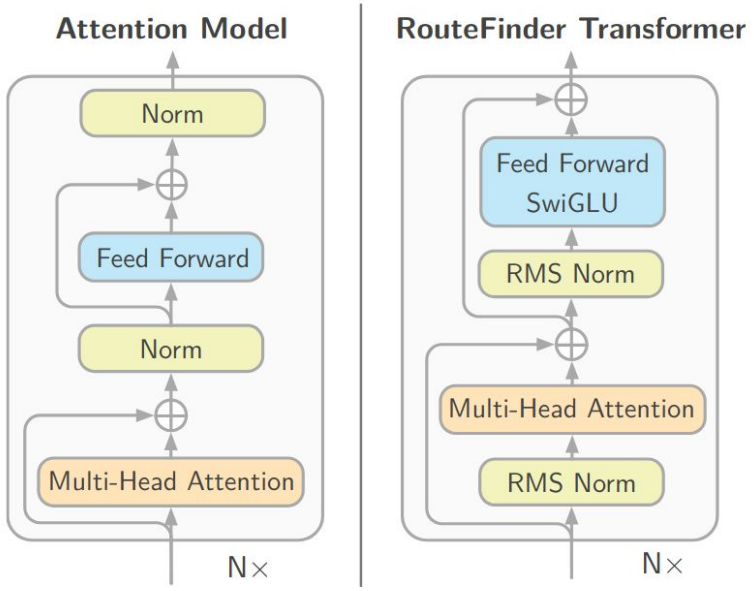
Ingredient 2: Unified Representation

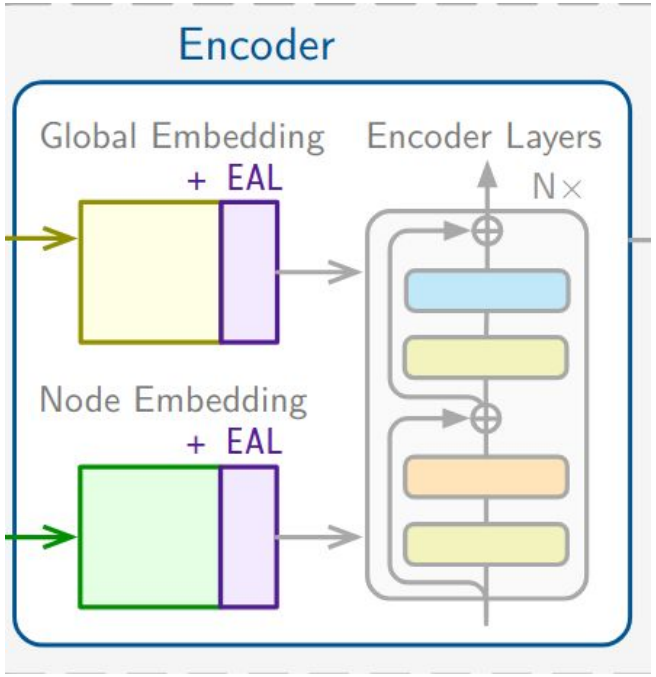
AI4CO



ICML
International Conference On Machine Learning

Ingredient 3: Model





1. We use a modern architecture with a Transformer Encoder with Pre-Layer Normalization (as in Llama, DeepSeek…)
2. We model global features with ad-hoc Global Embeddings

Ingredient 4: Training

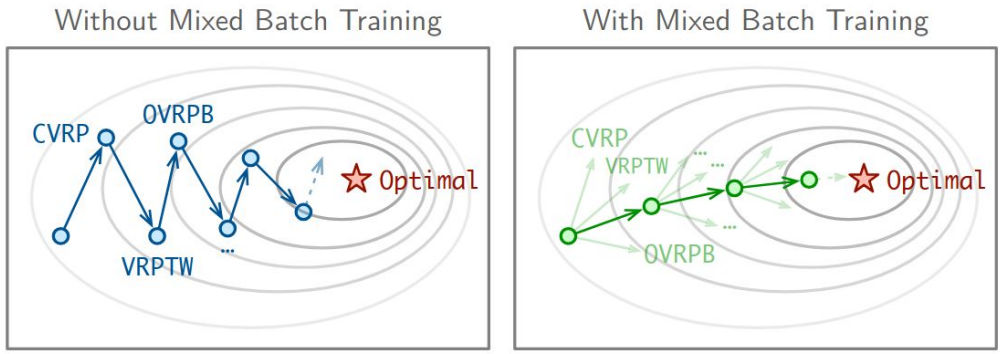
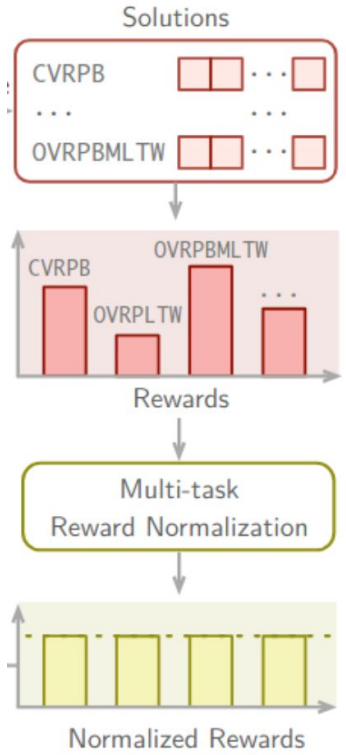
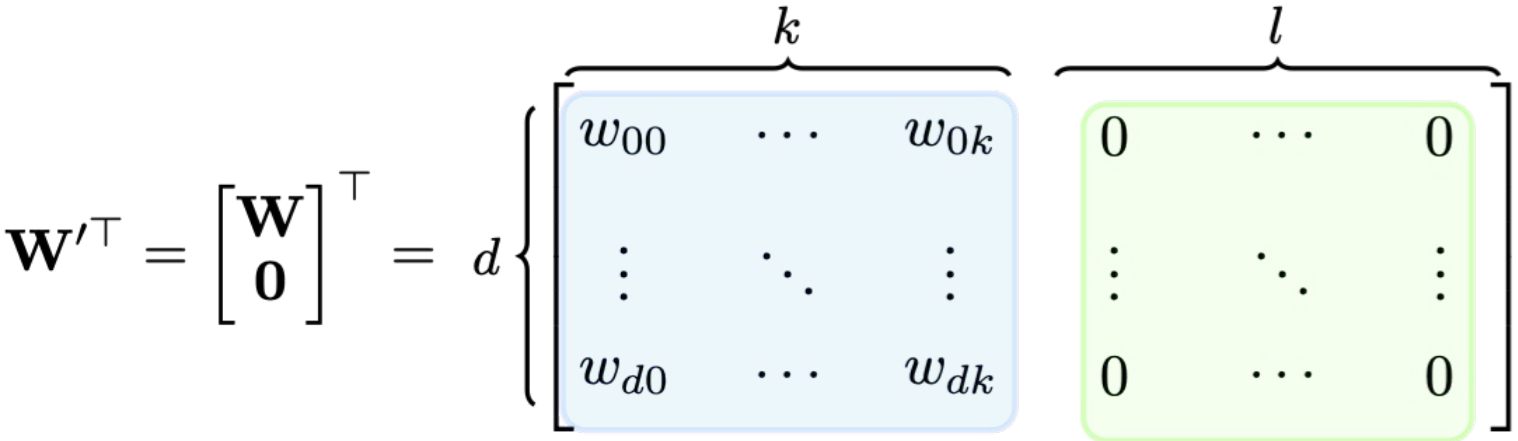


Figure 4: **[Left]** Training without MBT leads to instability since at each step, the optimization is biased toward a single task. **[Right]** Training RouteFinder with MBT allows for stable training.



1. We use Mixed Batch Training to stabilize multi-task learning
2. Multi-task Reward Normalization further stabilized different reward scales (different tasks)

Ingredient 5: Finetuning

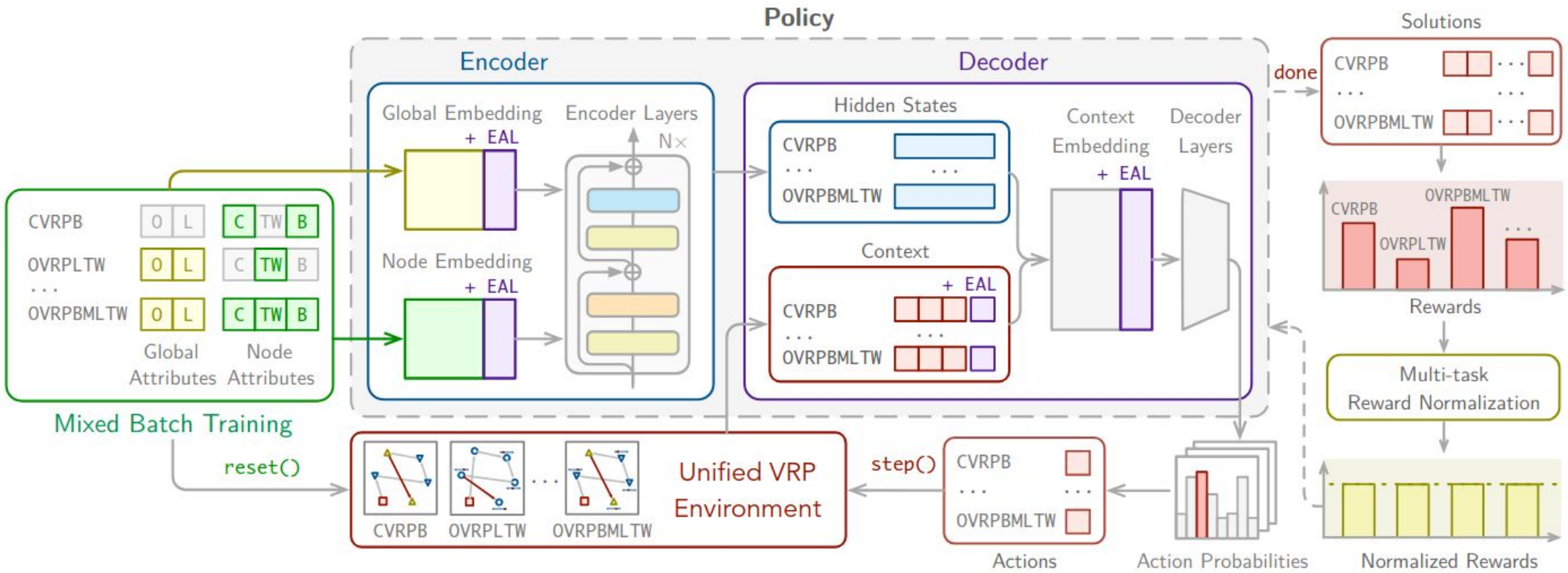


- Problem: we have a new attribute *outside* of initial set we need to "expand our dictionary" (linear projections to embedding space)

→ Efficient Adaptive Layers (EAL): simple 💡

1. Initialize new layer with previous weights and append 0 to new attributes
2. Finetune, fast!

RouteFinder: Final Recipe



Results: In-distribution

Table 1: Performance on 1000 test VRP instances. The lower, the better (↓). "∗" represents the best-known solutions. ROUTEFINDER (RF) models outperforms state-of-the-art neural baselines in all settings.

	Solver	$n = 50$			$n = 100$			Solver	$n = 50$			$n = 100$		
		Obj.	Gap	Time	Obj.	Gap	Time		Obj.	Gap	Time	Obj.	Gap	Time
CVRP	HGS-PyVRP	10.372	*	10.4m	15.628	*	20.8m	VRPTW	16.031	*	10.4m	25.423	*	20.8m
	OR-Tools	10.572	1.907%	10.4m	16.280	4.178%	20.8m		16.089	0.347%	10.4m	25.814	1.506%	20.8m
	MTPOMO	10.518	1.411%	2s	15.934	1.988%	7s		16.410	2.364%	1s	26.412	3.873%	7s
	MVMoE	10.501	1.242%	2s	15.888	1.694%	9s		16.404	2.329%	2s	26.389	3.788%	9s
	RF-POMO	10.508	1.314%	2s	15.908	1.826%	7s		16.367	2.094%	1s	26.336	3.575%	7s
	RF-MoE	**10.499**	**1.226%**	2s	15.876	1.622%	9s		16.389	2.234%	2s	26.322	3.519%	9s
	RF-TE	10.504	1.274%	2s	**15.857**	**1.505%**	7s		**16.364**	**2.077%**	1s	**26.235**	**3.178%**	7s
OVRP	HGS-PyVRP	6.507	*	10.4m	9.725	*	20.8m	VRPL	10.587	*	10.4m	15.766	*	20.8m
	OR-Tools	6.553	0.686%	10.4m	9.995	2.732%	20.8m		10.570	2.343%	10.4m	16.466	5.302%	20.8m
	MTPOMO	6.718	3.209%	1s	10.210	4.965%	6s		10.775	1.734%	1s	16.149	2.434%	7s
	MVMoE	6.702	2.965%	2s	10.177	4.621%	9s		10.751	1.505%	2s	16.099	2.115%	9s
	RF-POMO	6.698	2.904%	1s	10.180	4.659%	6s		10.751	1.523%	1s	16.107	2.174%	6s
	RF-MoE	6.697	2.886%	2s	10.139	4.229%	9s		**10.737**	**1.388%**	2s	16.070	1.941%	9s
	RF-TE	**6.684**	**2.687%**	1s	**10.121**	**4.055%**	6s		10.749	1.502%	1s	**16.051**	**1.827%**	6s
VRPB	HGS-PyVRP	9.687	*	10.4m	14.377	*	20.8m	OVRPTW	10.510	*	10.4m	16.926	*	20.8m
	OR-Tools	9.802	1.159%	10.4m	14.933	3.853%	20.8m		10.519	0.078%	10.4m	17.027	0.583%	20.8m
	MTPOMO	10.033	3.564%	1s	15.082	4.922%	6s		10.668	1.479%	1s	17.420	2.892%	7s
	MVMoE	10.005	3.270%	2s	15.023	4.508%	8s		10.669	1.492%	2s	17.416	2.872%	10s
	RF-POMO	9.996	3.174%	1s	15.016	4.468%	6s		10.657	1.378%	1s	17.391	2.720%	7s
	RF-MoE	9.980	3.015%	2s	14.973	4.164%	8s		10.674	1.539%	2s	17.387	2.697%	10s
	RF-TE	**9.977**	**2.989%**	1s	**14.942**	**3.952%**	6s		**10.652**	**1.326%**	1s	**17.327**	**2.346%**	7s

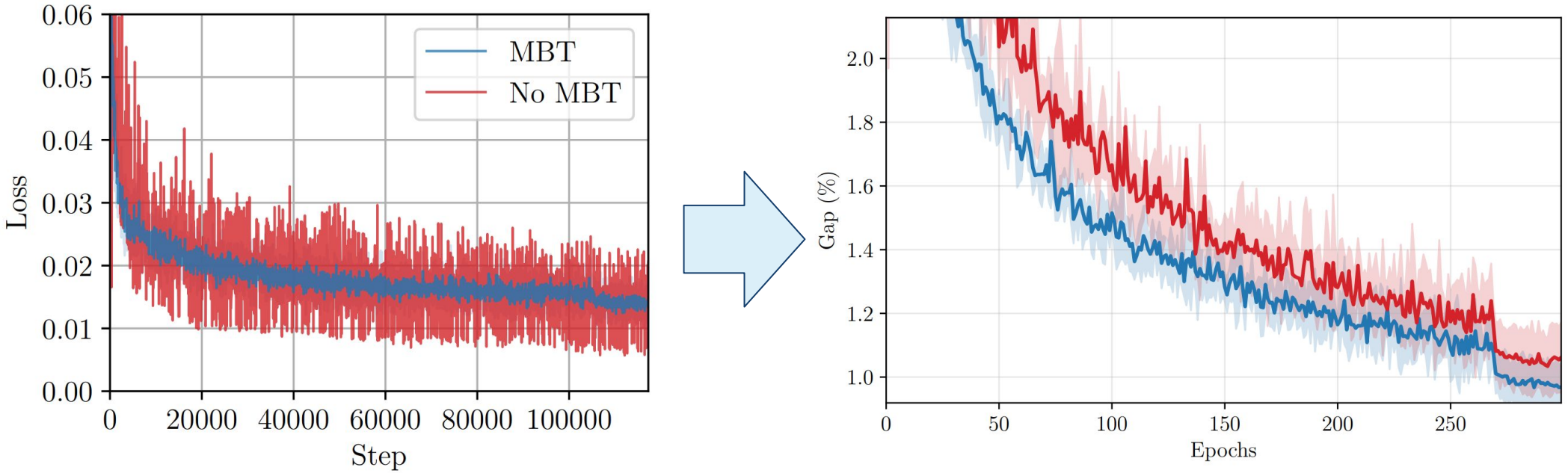
→ RouteFinder (RF) variants achieve SOTA results among learning methods

Results: Out-of-Distribution

Set-X		POMO†		MTPOMO †		MVMoE†		MVMoE-L†		MTPOMO		MVMoE		RF-TE	
Instance	Opt.	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap
X-n502-k39	69226	75617	9.232%	77284	11.640%	73533	6.222%	74429	7.516%	69226	9.410%	76338	10.274%	71791	3.705%
X-n513-k21	24201	30518	26.102%	28510	17.805%	32102	32.647%	31231	29.048%	24201	42.511%	32639	34.866%	28465	17.619%
X-n524-k153	154593	201877	30.586%	192249	24.358%	186540	20.665%	182392	17.982%	154593	14.771%	170999	10.612%	174381	12.800%
X-n536-k96	94846	106073	11.837%	106514	12.302%	109581	15.536%	108543	14.441%	94846	16.109%	105847	11.599%	103272	8.884%
X-n936-k151	132715	237625	79.049%	186262	40.347%	220926	66.466%	190407	43.471%	132715	50.654%	161343	21.571%	163162	22.942%
X-n957-k87	85465	130850	53.104%	98198	14.898%	113882	33.250%	105629	23.593%	85465	48.127%	123633	44.659%	102689	20.153%
X-n979-k58	118976	147687	24.132%	138092	16.067%	146347	23.005%	139682	17.404%	118976	16.711%	131754	10.740%	129952	9.225%
X-n1001-k43	72355	100399	38.759%	87660	21.153%	114448	58.176%	94734	30.929%	72355	82.677%	88969	22.962%	85929	18.760%
Avg. Gap		29.658%		16.796%		26.408%		19.607%		30.202%		18.795%		**12.303%**	

OOD generalization is better compared to models trained on single task variants!

Results: Training Stability



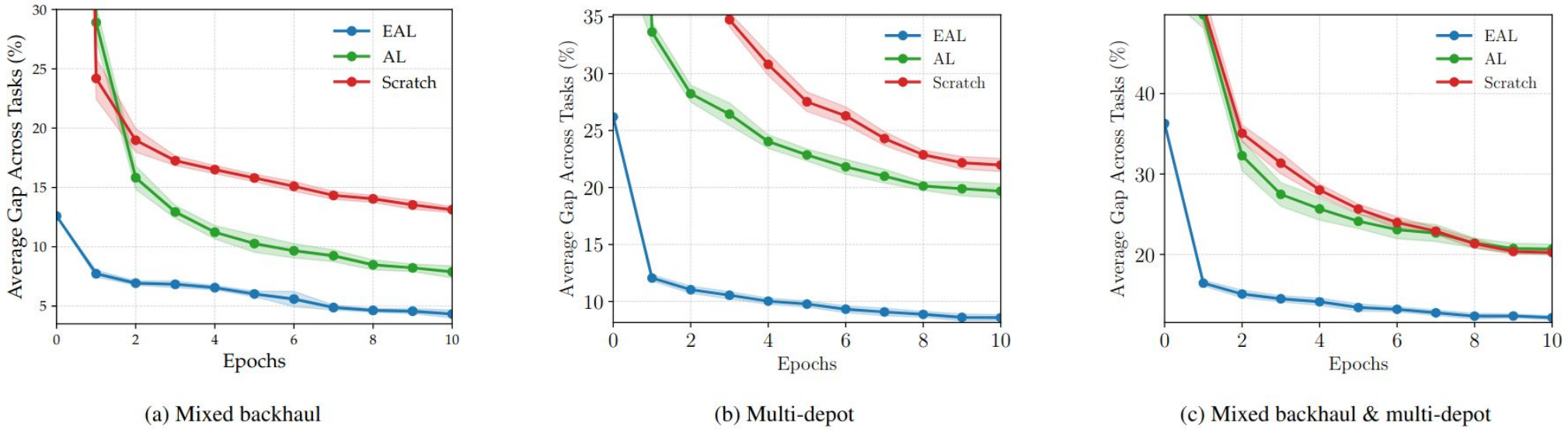
Mixed Batch Training (MBT) greatly improves convergence speed by aggregating gradients from multiple tasks

Results: Efficiency

	# models	# num parameters	# total training GPU time
POMO (Single Task x 16)	16	28.2M	16 days
RouteFinder (Multi-Task x 16)	1	2.3M	1 day

Instead of specializing a model for one task among 16, our Foundation Model trained on all tasks a single unified model and significantly save resources!

Results: Finetuning with EAL

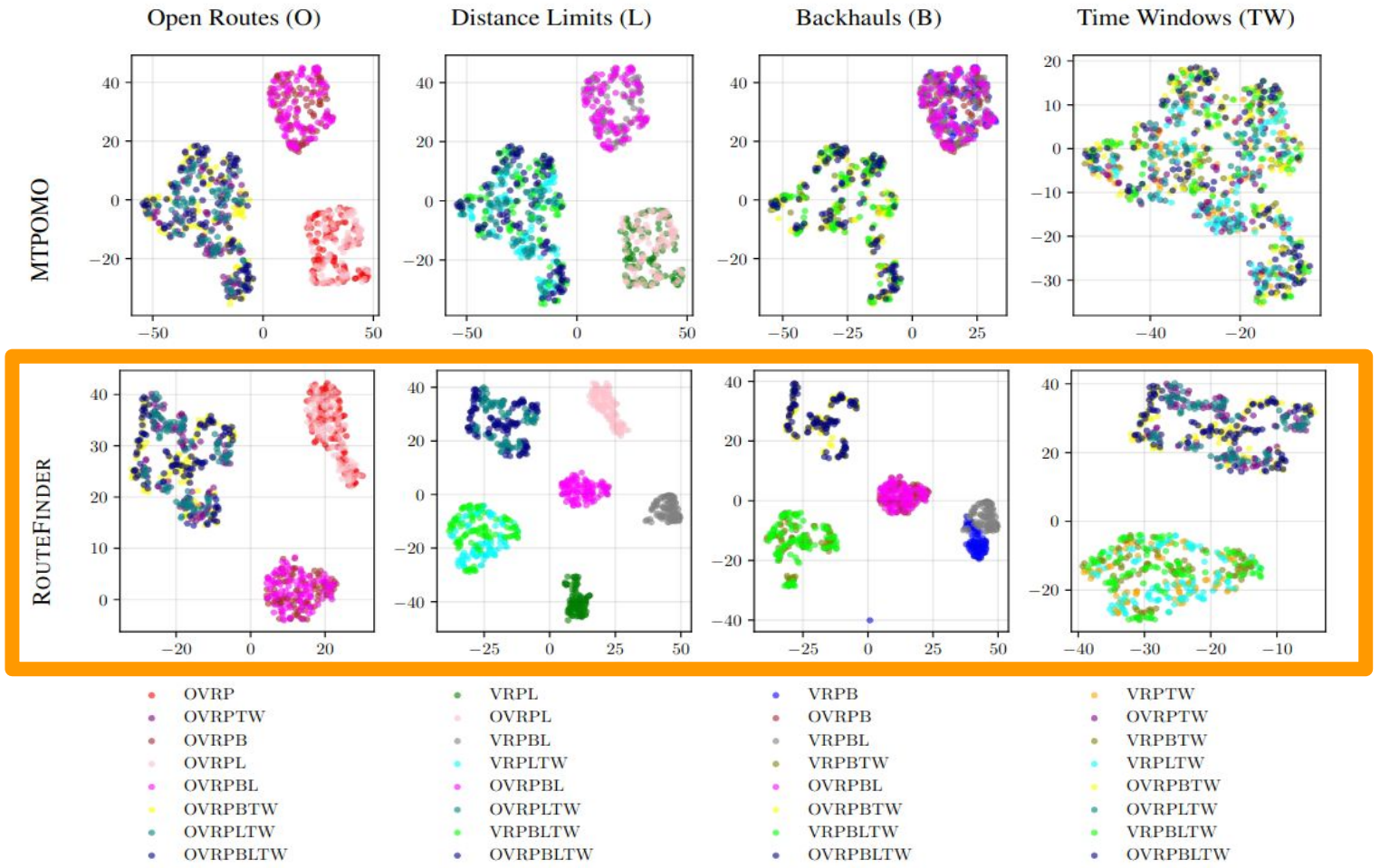


(a) Mixed backhaul (b) Multi-depot (c) Mixed backhaul & multi-depot

- Few-shot finetuning to *unseen attributes*
- Efficient Adaptive Layers (EAL) converge much faster than creating new layers [1] → model has learned transferable knowledge to new tasks
- Also, much better than finetuning a specialized single-task model!

[1] Lin, Z., Wu, Y., Zhou, B., Cao, Z., Song, W., Zhang, Y., & Jayavelu, S. (2024). Cross-problem learning for solving vehicle routing problems. IJCAI 2024

Latent Space Analysis



RouteFinder learns more complex relationships across tasks !

Thank you!

AI4CO

Star us on Github⭐

Join AI4CO 🤝

ICML
International Conference
On Machine Learning