MVMoE: Multi-Task Vehicle Routing Solver with Mixture-of-Experts

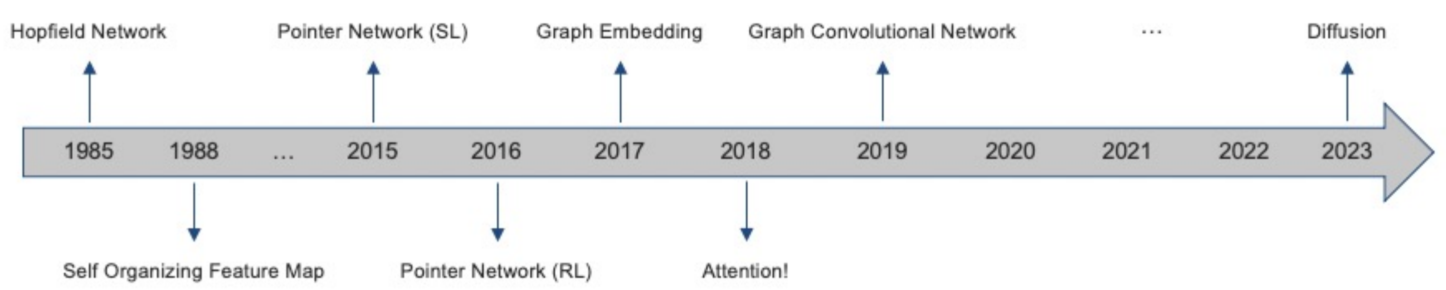
Jianan Zhou[1], Zhiguang Cao[2], Yaoxin Wu[3], Wen Song[4], Yining Ma[1], Jie Zhang[1], Chi Xu[5]

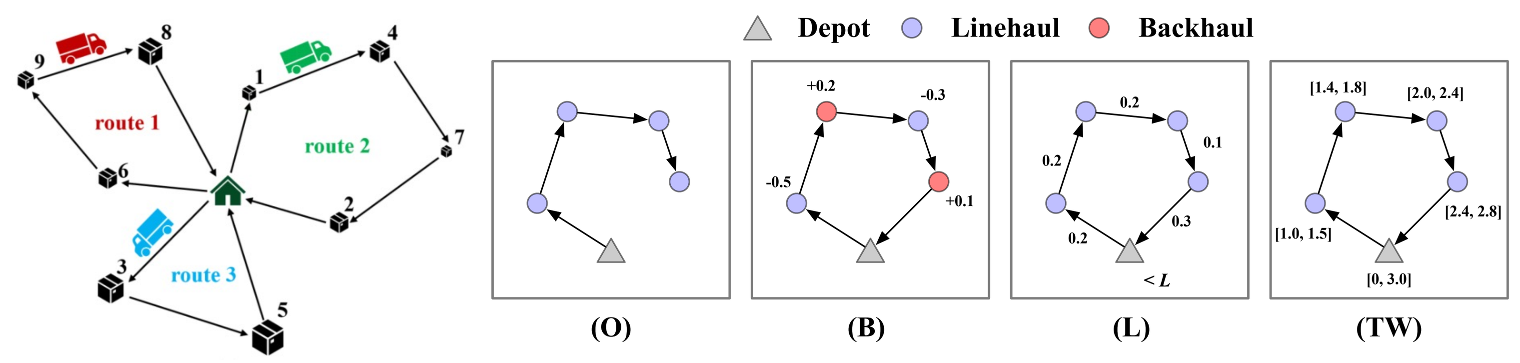
[1]Nanyang Technological University [2]Singapore Management University [3]Eindhoven University of Technology [4]Shandong University [5]A*STAR

Introduction

Vehicle Routing Problems (VRPs)

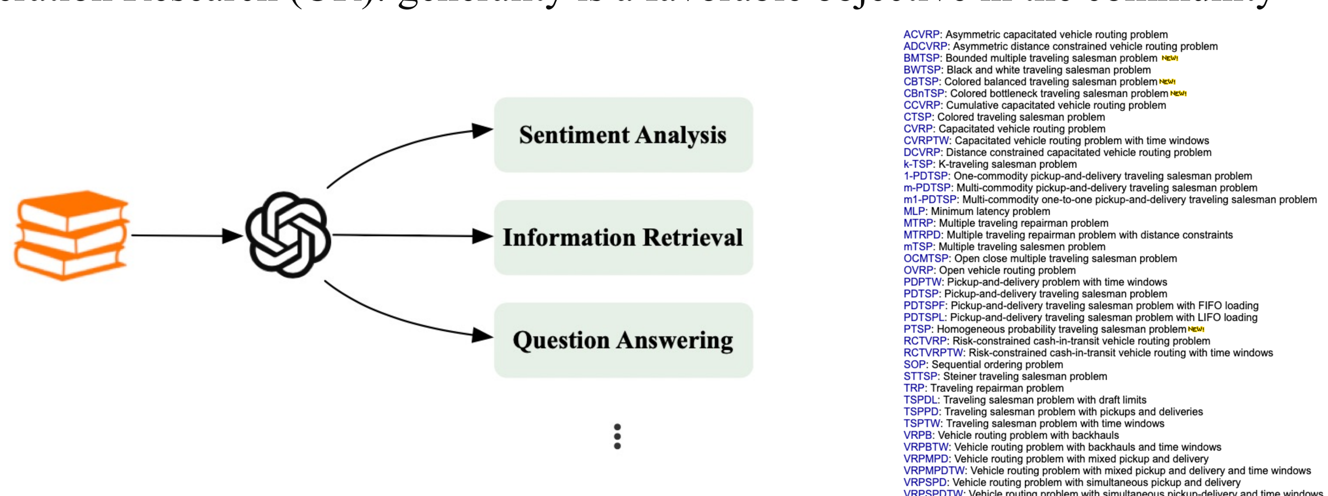
- A class of NP-hard combinatorial optimization problems
- Find an optimal solution that satisfies all constraints
- Exact methods (e.g., B&B framework); Heuristic methods (e.g., LKH3)
- Neural methods





Motivation

- Recent neural methods are only structured and trained independently on a specific problem, making them less generic and practical
- Machine Learning (ML): foundation models (FMs) have shown great success in language and vision domains
- Operation Research (OR): generality is a favorable objective in the community



Contribution

- An early work studying the cross-problem generalization in VRPs
- A unified neural solver to deal with multiple VRPs
- A hierarchical gating for a trade-off between performance and computation
- Strong empirical results on 16 VRPs
- Extensive studies on the effect of MoE configurations

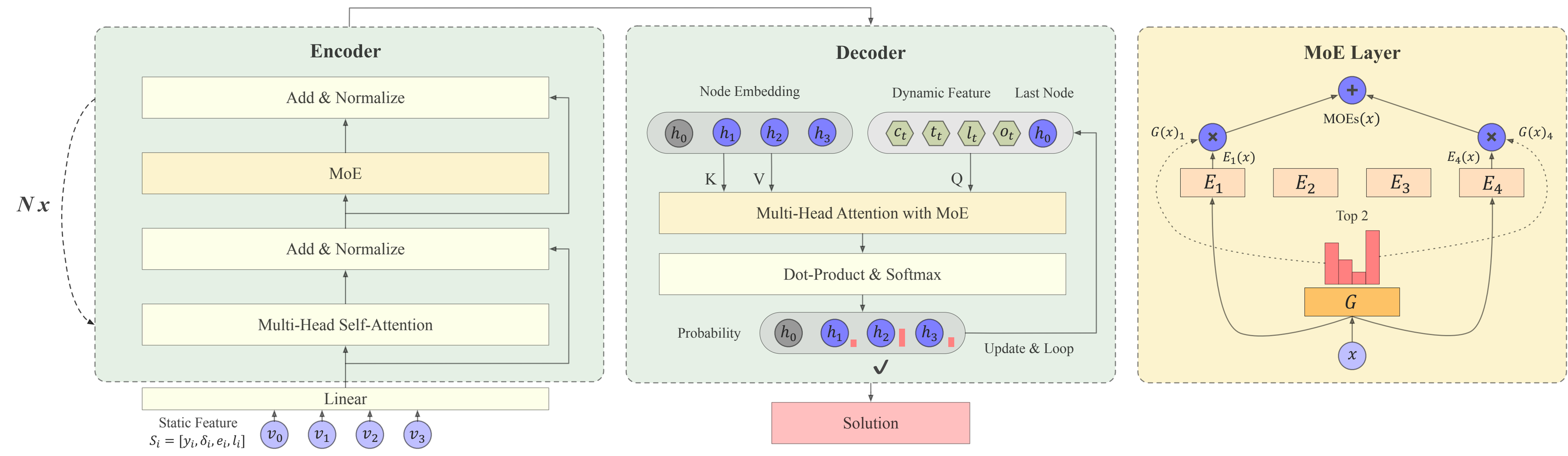
	Capacity (C)	Open Route (O)	Backhaul (B)	Duration Limit (L)	Time Window (TW)
CVRP	✓				
OVRP	✓	✓			
VRPB	✓		✓		
VRPL	✓			✓	
VRPTW	✓				✓
OVRPTW	✓	✓			✓
OVRPB	✓	✓	✓		
OVRPL	✓	✓		✓	
VRPBL	✓		✓	✓	
VRPBTW	✓		✓		✓
VRPLTW	✓			✓	✓
OVRPBL	✓	✓	✓	✓	
OVRPBTW	✓	✓	✓		✓
OVRPLTW	✓	✓		✓	✓
VRPBLTW	✓		✓	✓	✓
OVRPBLTW	✓	✓	✓	✓	✓

Methodology

Multi-Task VRP Solver

- Autoregressive Construction Solver: AM, **POMO**, LEHD, …
- Attribute Composition
  - Static features of node i: $S_i = [y_i, \delta_i, e_i, l_i]$
  - Dynamic features of decoding step t: $D_t = [c_t, t_t, l_t, o_t]$
  - Missing features are padded to the default values
  - CVRP Example: $S_i = [y_i, \delta_i, 0, 0]$, $D_t = [c_t, 0, l_t, 0]$
  - Can solve 2^n problem variants
- RL Training

$$\nabla_\theta \mathcal{L}(\theta|\mathcal{G}) = \mathbb{E}_{p_\theta(\tau|\mathcal{G})}\left[\left(c(\tau) - b(\mathcal{G})\right)\nabla_\theta \log p_\theta(\tau|\mathcal{G})\right]$$

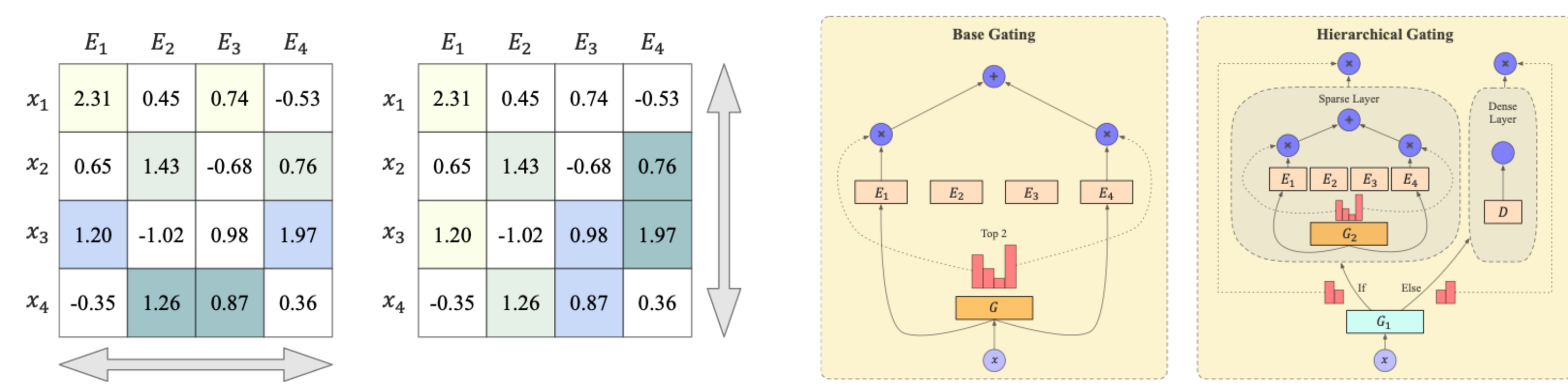


Mixture-of-Experts

- An MoE layer consists of
  - m experts $\{E_1, E_2, \ldots, E_m\}$, each of which is a linear layer or FFN
  - a gating network G

$$MoE(x) = \sum_{j=1}^{m} G(x)_j E_j(x)$$

- A sparse vector $G(x)$ only activates a small subset of experts with partial parameters
- Load balancing is required for MoE's training (e.g., by using an auxiliary loss)



Gating Mechanism

- Gating Level: **Node** / Instance / Problem
- Gating Algorithm: Random / **Input-Choice** / Expert-Choice / Soft MoE
- Trade-off between performance and computation: Employing MoEs in the decoder
  - can significantly boost the empirical performance
  - can significantly increase the training overheads due to the autoregressive nature
- An MoE layer with **Hierarchical Gating** consists of
  - an upper gating network G_1
  - a sparse layer: m experts $\{E_1, E_2, \ldots, E_m\}$ + a lower gating network G_2
  - a dense layer: D

$$Hierarchical\_MoE(x) = \begin{cases} G_1(x)_0 \sum_{j=1}^{m} G_2(x)_j E_j(x) \\ G_1(x)_1 D(x) \end{cases}$$

- The hierarchical gating enhances computational efficiency with a minor loss on in-distribution performance but significantly improves out-of-distribution performance

Experiment

Performance Comparison & Ablation

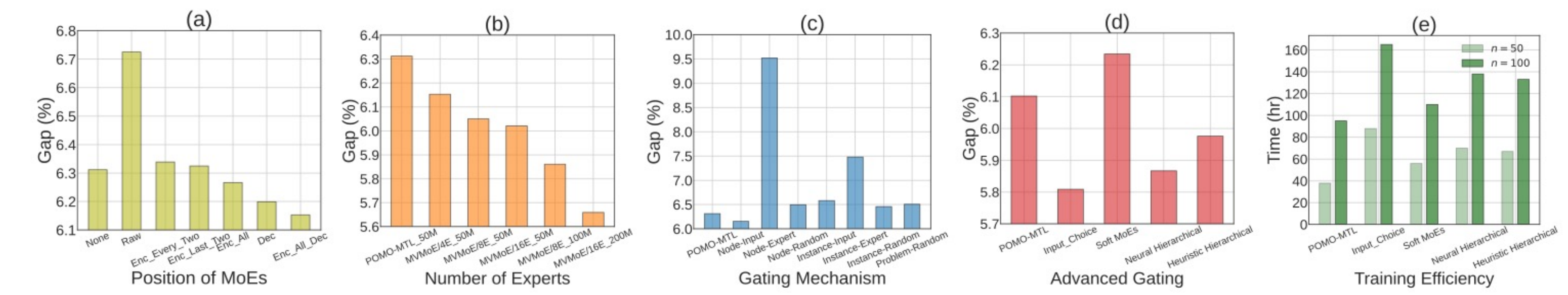


Table 1. Performance on 1K test instances of trained VRPs. * represents 0.000%, with which the gaps are computed.

	Method	n = 50 Obj.	n = 50 Gap	n = 50 Time	n = 100 Obj.	n = 100 Gap	n = 100 Time
CVRP	HGS	10.334	*	4.6m	15.504	*	9.1m
	LKH3	10.346	0.115%	9.9m	15.590	0.556%	18.0m
	OR-Tools	10.540	1.962%	10.4m	16.381	5.652%	20.8m
	OR-Tools (x10)	10.418	0.788%	1.7h	15.935	2.751%	3.5h
	POMO	10.418	0.806%	3s	15.734	1.488%	9s
	POMO-MTL	10.437	0.987%	3s	15.790	1.846%	9s
	MVMoE/4E	**10.428**	**0.896%**	4s	**15.760**	**1.653%**	11s
	MVMoE/4E-L	10.434	0.955%	4s	15.771	1.728%	10s
OVRP	LKH3	6.511	0.198%	4.5m	9.828	*	5.3m
	OR-Tools	6.531	0.495%	10.4m	10.010	1.806%	20.8m
	OR-Tools (x10)	6.498	*	1.7h	9.842	0.122%	3.5h
	POMO	6.609	1.685%	2s	10.044	2.192%	8s
	POMO-MTL	6.671	2.634%	2s	10.169	3.458%	8s
	MVMoE/4E	**6.655**	**2.402%**	3s	**10.138**	**3.136%**	10s
	MVMoE/4E-L	6.665	2.548%	3s	10.145	3.214%	9s
VRPB	OR-Tools	8.127	0.989%	10.4m	12.185	2.594%	20.8m
	OR-Tools (x10)	8.046	*	1.7h	11.878	*	3.5h
	POMO	8.149	1.276%	2s	11.993	0.995%	7s
	POMO-MTL	8.182	1.684%	2s	12.072	1.674%	7s
	MVMoE/4E	**8.170**	**1.540%**	3s	**12.027**	**1.285%**	9s
	MVMoE/4E-L	8.176	1.605%	3s	12.036	1.368%	8s

	Method	n = 50 Obj.	n = 50 Gap	n = 50 Time	n = 100 Obj.	n = 100 Gap
VRPTW	HGS	14.509	*	8.4m	24.339	*
	LKH3	14.607	0.664%	5.5m	24.721	1.584%
	OR-Tools	14.915	2.694%	10.4m	25.894	6.297%
	OR-Tools (x10)	14.665	1.011%	1.7h	25.212	3.482%
	POMO	14.940	2.990%	3s	25.367	4.307%
	POMO-MTL	15.032	3.637%	3s	25.610	5.313%
	MVMoE/4E	**14.999**	**3.410%**	4s	**25.512**	**4.903%**
	MVMoE/4E-L	15.013	3.500%	3s	25.519	4.927%
VRPL	LKH3	10.571	0.790%	7.8m	15.771	*
	OR-Tools	10.677	1.746%	10.4m	16.496	4.587%
	OR-Tools (x10)	10.495	*	1.7h	16.004	1.444%
	POMO	10.491	-0.008%	2s	15.785	0.093%
	POMO-MTL	10.513	0.201%	2s	15.846	0.479%
	MVMoE/4E	**10.501**	**0.092%**	3s	**15.812**	**0.261%**
	MVMoE/4E-L	10.506	0.131%	3s	15.821	0.323%
OVRPTW	OR-Tools	8.737	0.592%	10.4m	14.635	1.756%
	OR-Tools (x10)	8.683	*	1.7h	14.380	*
	POMO	8.891	2.377%	3s	14.728	2.467%
	POMO-MTL	8.987	3.470%	3s	15.008	4.411%
	MVMoE/4E	**8.964**	**3.210%**	4s	**14.927**	**3.852%**
	MVMoE/4E-L	8.974	3.322%	4s	14.940	3.941%

Table 2. Zero-shot generalization on 1K test instances of unseen VRPs. * represents 0.000%, with which the gaps are computed.

	Method	n = 50 Obj.	n = 50 Gap	n = 50 Time	n = 100 Obj.	n = 100 Gap	n = 100 Time
OVRPB	OR-Tools	5.764	0.332%	10.4m	8.522	1.852%	20.8m
	OR-Tools (x10)	5.745	*	1.7h	8.365	*	3.5h
	POMO-MTL	6.116	6.430%	2s	8.979	7.335%	8s
	MVMoE/4E	**6.092**	**5.999%**	3s	**8.959**	**7.088%**	9s
	MVMoE/4E-L	6.122	6.522%	3s	8.972	7.243%	9s
VRPBL	OR-Tools	8.131	1.254%	10.4m	12.095	2.586%	20.8m
	OR-Tools (x10)	8.029	*	1.7h	11.790	*	3.5h
	POMO-MTL	8.188	1.971%	2s	11.998	1.793%	8s
	MVMoE/4E	**8.172**	**1.776%**	3s	**11.945**	**1.346%**	9s
	MVMoE/4E-L	8.180	1.872%	3s	11.960	1.473%	9s
VRPLTW	OR-Tools	14.815	1.432%	10.4m	25.823	2.534%	20.8m
	OR-Tools (x10)	14.598	*	1.7h	25.195	*	3.5h
	POMO-MTL	14.961	2.586%	3s	25.619	1.920%	12s
	MVMoE/4E	**14.937**	**2.421%**	4s	**25.514**	**1.471%**	13s
	MVMoE/4E-L	14.953	2.535%	4s	25.529	1.545%	12s
OVRPBTW	OR-Tools	8.758	0.927%	10.4m	14.713	2.268%	20.8m
	OR-Tools (x10)	8.675	*	1.7h	14.384	*	3.5h
	POMO-MTL	9.514	9.628%	3s	15.879	10.453%	10s
	MVMoE/4E	**9.486**	**9.308%**	4s	**15.808**	**9.948%**	11s
	MVMoE/4E-L	9.515	9.630%	3s	15.841	10.188%	10s
VRPBLTW	OR-Tools	14.890	1.402%	10.4m	25.979	2.518%	20.8m
	OR-Tools (x10)	14.677	*	1.7h	25.342	*	3.5h
	POMO-MTL	15.980	9.035%	3s	27.247	7.746%	11s
	MVMoE/4E	**15.945**	**8.775%**	4s	**27.142**	**7.332%**	12s
	MVMoE/4E-L	15.963	8.915%	4s	27.177	7.473%	11s

	Method	n = 50 Obj.	n = 50 Gap	n = 50 Time	n = 100 Obj.	n = 100 Gap	n = 100 Time
OVRPL	OR-Tools	6.522	0.480%	10.4m	9.966	1.783%	20.8m
	OR-Tools (x10)	6.490	*	1.7h	9.790	*	3.5h
	POMO-MTL	6.668	2.734%	2s	10.126	3.441%	9s
	MVMoE/4E	**6.650**	**2.454%**	3s	**10.097**	**3.148%**	10s
	MVMoE/4E-L	6.659	2.597%	3s	10.106	3.244%	9s
VRPBTW	OR-Tools	15.053	1.857%	10.4m	26.217	2.858%	20.8m
	OR-Tools (x10)	14.771	*	1.7h	25.496	*	3.5h
	POMO-MTL	16.055	8.841%	3s	27.319	7.413%	10s
	MVMoE/4E	**16.022**	**8.600%**	4s	**27.236**	**7.078%**	11s
	MVMoE/4E-L	16.041	8.745%	4s	27.265	7.190%	10s
OVRPBL	OR-Tools	5.771	0.549%	10.4m	8.555	2.459%	20.8m
	OR-Tools (x10)	5.739	*	1.7h	8.348	*	3.5h
	POMO-MTL	6.104	6.306%	2s	8.961	7.343%	8s
	MVMoE/4E	**6.076**	**5.843%**	3s	**8.942**	**7.115%**	9s
	MVMoE/4E-L	6.104	6.310%	3s	8.957	7.300%	9s
OVRPLTW	OR-Tools	8.728	0.656%	10.4m	14.535	1.779%	20.8m
	OR-Tools (x10)	8.669	*	1.7h	14.279	*	3.5h
	POMO-MTL	8.987	3.633%	3s	14.896	4.374%	11s
	MVMoE/4E	**8.966**	**3.396%**	4s	**14.828**	**3.903%**	12s
	MVMoE/4E-L	8.974	3.488%	4s	14.839	3.971%	10s
OVRPBLTW	OR-Tools	8.729	0.624%	10.4m	14.496	1.724%	20.8m
	OR-Tools (x10)	8.673	*	1.7h	14.250	*	3.5h
	POMO-MTL	9.532	9.851%	3s	15.738	10.498%	10s
	MVMoE/4E	**9.503**	**9.516%**	4s	**15.671**	**10.009%**	11s
	MVMoE/4E-L	9.518	9.682%	4s	15.706	10.263%	10s

Summary

Towards Foundation Models for Combinatorial Optimization (FM4CO)

- LLMs (FMs from other domains) for CO
  - Generate solutions or algorithms (i.e., code)
  - Automate problem formulation or simplify tools' usage
  - Enhance interpretability
- Domain FMs for CO
  - A unified model to solve various optimization tasks

Limitation

- Scalability
  - The number of model parameters
  - Training on large-scale instances
- Generality
- Interpretability

Ads & Links

- **Jianan Zhou is looking for a postdoc position in 2025**
- **Email: jianan004@e.ntu.edu.sg**
- **Links of WeChat, Paper, and Code:**