DGL: Dynamic Global-Local Information Aggregation for Scalable VRP Generalization with Self-Improvement Learning

Yubin Xiao[1], **Yuesong Wu**[1], **Rui Cao**[1], **Di Wang**[2], **Zhiguang Cao**[3], **Xuan Wu**[1], **Peng Zhao**[1], **Yuanshu Li**[1], **You Zhou**[1], **Yuan Jiang**[4]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, China
[2]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore
[3]Singapore Management University, Singapore
[4]Nanyang Technological University, Singapore

Abstract

The Vehicle Routing Problem (VRP) is a critical combinatorial optimization problem with wide-reaching real-world applications, particularly in logistics and transportation. While neural network-based VRP solvers have shown impressive results on test instances similar to training data, their performance often degrades when faced with varying scales and unseen distributions, limiting their practical applicability. To overcome these limitations, we introduce DGL (Dynamic Global-Local Information Aggregation), a novel approach that combines global and local information to effectively solve VRPs. DGL dynamically adjusts local node selections within a localized range, capturing local invariance across problems of different scales and distributions, thereby enhancing generalization. At the same time, DGL integrates global context into the decision-making process, providing richer information for more informed decisions. Additionally, we propose a replacement-based self-improvement learning framework that leverages data augmentation and random replacement techniques, further enhancing DGL's robustness. Extensive experiments on synthetic datasets, benchmark datasets, and real-world country map instances demonstrate that DGL achieves state-of-the-art performance, particularly in generalizing to large-scale VRPs and real-world scenarios. These results showcase DGL's effectiveness in solving complex, realistic optimization challenges and highlight its potential for practical applications.

1 Introduction

The Vehicle Routing Problem (VRP) is a critical Combinatorial Optimization Problem (COP) with numerous real-world applications in logistics and transportation [Kim *et al.*, 2015; Alkaya and Duman, 2013]. Given its significance, many exact and heuristic algorithms have been developed over the years [Applegate *et al.*, 2007; Helsgaun, 2017; Li *et al.*, 2021]. However, these traditional algorithms often incur high computational costs. Recently, neural network (NN)-based models have emerged as a promising alternative for solving VRPs [Wu *et al.*, 2024]. These models typically leverage learned NNs to acquire heuristics for constructing solutions or improving the quality of current solutions. By leveraging the underlying patterns in training instances, neural VRP solvers [Kwon *et al.*, 2020; Xiao *et al.*, 2024a] achieve competitive or even superior solution quality compared to the traditional algorithms when solving test instances with the same pattern, while significantly reducing inference time.

Although neural VRP solvers have demonstrated impressive performance on instances matching the size and distribution pattern of the training data, their effectiveness often degrades significantly when applied to instances with varying scales or unseen distributions [Joshi *et al.*, 2022; Bdeir *et al.*, 2023]. For instance, fluctuations in customer numbers or changes in locations due to factors such as weather conditions or holidays can significantly alter problem characteristics, limiting the practical applicability of neural VRP solvers in real-world scenarios. To alleviate parts of these issues, researchers have proposed methods aimed at improving distribution and size generalization. Distribution generalization methods [Jiang *et al.*, 2022; Bi *et al.*, 2022] typically integrate various generalization algorithms into the model's learning framework. However, these approaches often struggle with cross-scale instances, particularly in large-scale VRPs. Conversely, state-of-the-art (SOTA) size generalization methods yield impressive results on large-scale VRPs by employing either a global-view [Drakulic *et al.*, 2023; Luo *et al.*, 2023] or local selection method [Fang *et al.*, 2024; Goh *et al.*, 2024]. However, global-view models suffer from excessive computational cost, while local selection models often lead to suboptimal solutions. Furthermore, most models [Kool *et al.*, 2019; Kwon *et al.*, 2020; Luo *et al.*, 2023; Gao *et al.*, 2024] rely on static embeddings during solution construction, failing to dynamically update information based on partial solutions, which limits the model's adaptability to changes in the problem state and hinders generalization.

To address the challenge of efficiently generalizing to the

diverse and complex nature of real-world VRPs, we propose a novel model named DGL (Dynamically Global-Local Information Aggregation) that enhances problem awareness by integrating dynamically adjusted global and local features into the decision-making process. Specifically, since optimal solutions often emerge in the local neighborhood—a pattern consistently observed across problems of different scales and distributions—we select neighboring nodes as candidates for subsequent node selection, thereby capturing the problem's local invariance [Gao *et al.*, 2024]. Simultaneously, global information is incorporated into the representations of these neighboring nodes and iteratively updated throughout the solution construction process, providing contextual understanding and facilitating the exploration of the solution space. By leveraging these continuously updated features, DGL captures the relationship between the current state and the broader problem context (e.g., direction to the destination), enabling more informed decision-making in node selection. Furthermore, to ensure DGL's robustness across diverse VRP instances, we propose a replacement-based self-improvement learning (SIL) method. This method promotes the diversity of datasets through data augmentation and enhances model robustness by randomly removing portions of the generated dataset during training. Finally, we evaluate DGL through extensive experiments on the Traveling Salesman Problem (TSP) and Capacitated VRP (CVRP), including synthetic datasets of varying scales and distribution patterns, two widely adopted benchmark datasets, and eight real-world country map instances. The results demonstrate that DGL exhibits excellent generalization capability, especially in solving large-scale VRP and real-world scenarios.

The key contributions of this work are as follows:

- We propose a novel model, DGL, that dynamically aggregates global and local information to enhance its perception of the current solving state, enabling it to efficiently solve diverse real-world VRPs.
- We introduce a replacement-based SIL method that uses data augmentation and random replacement to enhance DGL's robustness across different scenarios.
- We demonstrate DGL's excellent generalization performance through extensive experiments across a wide range of synthetic and real-world VRP instances, covering various scales and distribution patterns.

2 Related Work

In this section, we introduce several recent works aimed at enhancing model's generalization and review existing training paradigms for neural VRP solvers.

2.1 Generalization of Neural VRP Solvers

NN-based methods have shown promising results in solving VRPs [Kool *et al.*, 2022; Qiu *et al.*, 2022; Sun and Yang, 2023; Min *et al.*, 2023; Xia *et al.*, 2024; Xiao *et al.*, 2020; Xiao *et al.*, 2024b; Xiao *et al.*, 2025] but with suboptimal generalization in unseen (especially in real world) scenarios that may deviate from the training data [Joshi *et al.*, 2022]. Recent studies have been proposed to enhance model distribution and size generalization of these models, respectively.

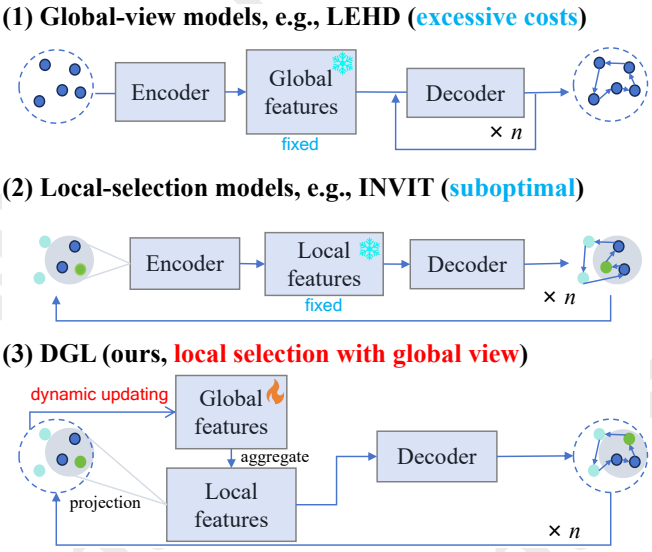

Figure 1: Comparative illustration of DGL and existing models. Global-view models incur excessive computational cost, while local selection models may lead to suboptimal solutions due to the limited views. Furthermore, The static nature of existing models limit their capacity to fully leverage updating state information. DGL enhances its perception of the current solving state by integrating dynamically updated global and local features into the decision-making process, thereby improving model generalization.

Distribution generalization methods [Zhang *et al.*, 2022; Geisler *et al.*, 2022; Jiang *et al.*, 2022; Bi *et al.*, 2022; Zhou *et al.*, 2023] aim to generalize models learned from VRPs of multiple predefined distribution patterns to instances with unseen distribution by adopting various generalization algorithms, such as adversarial training and meta-learning. However, these methods often face challenges when dealing with instances of varying scales, particularly in large-scale VRPs. As shown in Tables 1 and 2, even OMNI-VRP, the SOTA model trained on multiple distribution patterns, exhibits unsatisfactory performance on large-scale VRPs.

Prior size generalization methods [Zong *et al.*, 2022; Zheng *et al.*, 2024; Ye *et al.*, 2024] often involve decomposing a large-scale problem into smaller sub-problems, which are then solved independently. However, decomposing VRP with complex constraints can be difficult and may produce suboptimal solutions by failing to capture inter-dependencies among sub-problems [Luo *et al.*, 2024]. Recently, several SOTA VRP solvers that do not rely on D&C have been proposed. These solvers frequently utilize an encoder-decoder framework, where the encoder extracts node features, and the decoder selects the next node from a candidate set of unvisited nodes based on the encoded features. These methods have demonstrated promising results in large-scale problems and can broadly be categorized into global-view [Zhou *et al.*, 2023; Drakulic *et al.*, 2023; Li *et al.*, 2024; Gao *et al.*, 2024] and local selection [Fang *et al.*, 2024; Lyu *et al.*, 2024; Goh *et al.*, 2024]) models. Global-view models consider all unvisited nodes as candidates, while local selection models restrict candidate nodes to the k-nearest neighbors. Furthermore, existing models often employ static embeddings, where

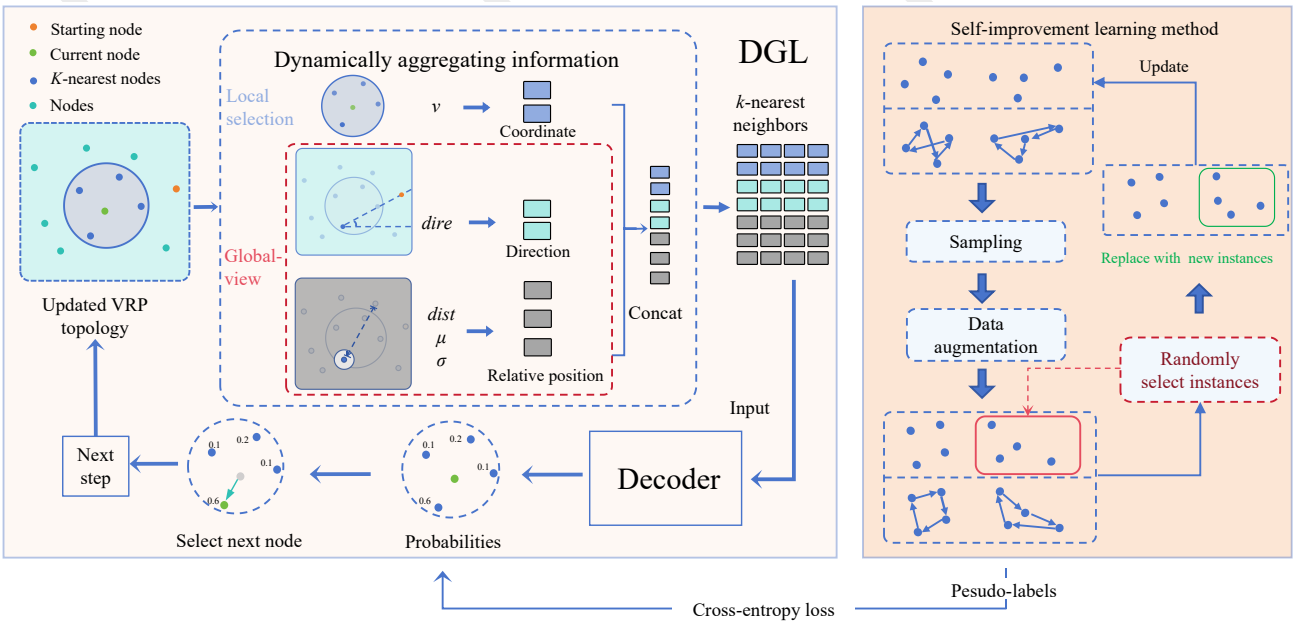

Figure 2: Overall framework of our proposed DGL and self-improvement learning method.

the node features to determine the next node remain fixed throughout the solution process. This static nature limits the solver's ability to dynamically adapt to changes in the solution stage. We illustrate these limitations and the distinctions between DGL and the existing model in Figure 1.

2.2 Training Paradigms

Neural VRP solvers typically rely on either Supervised Learning (SL) or Reinforcement Leaning (RL). SL-based methods [Drakulic *et al.*, 2023; Luo *et al.*, 2023] require optimal solutions as training labels, which are often challenging to obtain and computationally expensive. In contrast, RL-based approaches [Gao *et al.*, 2024; Fang *et al.*, 2024] suffer from sparse rewards and excessive computing costs, making them difficult to scale to larger instances [Luo *et al.*, 2023]. Recently, a novel SIL paradigm has been explored, which selects the best solution from multiple sampled solutions to serve as an expert trajectory for supervised imitation learning. SIL methods avoid the need for expensive labels required by SL and address some of the challenges associated with RL, achieving promising results. However, existing SIL methods [Pirnay and Grimm, 2024; Luo *et al.*, 2024] rely on fixed training instances without replacement during each epoch, which limits the model's exploration capabilities and reduces robustness. Additionally, these SIL methods determine policy updates based solely on performance over a validation set, which not only reduces training efficiency but also risks overfitting to the validation set, potentially leading to suboptimal solutions.

Unlike existing training methods, in this paper, we propose a replacement-based SIL approach that eliminates the dependence on validation sets and introduces random replacement of training instances at each epoch. This strategy enhances model robustness and expands the exploration space, improving overall training efficiency and solution quality.

3 Methodology

This section first presents the formulation of VRPs and then proposes a novel model named DGL that dynamically aggregates global and local information to efficiently solve VRPs of varying scales and distributions. Furthermore, we propose a replacement-based SIL training method to enhance DGL's robustness. The framework of DGL is presented in Figure 2.

3.1 Problem Setting

We define a VRP-n instance as a graph with n nodes, where each node is represented by its h-dimensional coordinates $v_i \in \mathbb{R}^{1\times h}$. The optimal solution to a VRP is the tour π^* that visits all nodes with the minimum cost, $c(\pi^*)$, corresponding to the shortest overall length of the tour. Solving different VRP variants involves adhering to various problem-specific constraints. This study focuses on two prominent VRP variants, i.e., TSP and CVRP, due to their representativeness and extensive applications across diverse domains [Kim *et al.*, 2015]. In TSP, each node must be visited exactly once. CVRP extends TSP by introducing a depot, vehicle capacity constraints, and node demands smaller than the vehicle's capacity. A CVRP tour comprises multiple sub-tours, each representing a vehicle that starts and ends at the depot, visiting a subset of nodes, with the total demand of each sub-tour not exceeding the vehicle's capacity.

3.2 DGL: Dynamic Information Aggregation

Neural VRP solvers often exhibit unsatisfactory performance when applied to problems with varying sizes and distributions. While local selection-based approaches are effective for large-scale problems, they tend to converge to suboptimal solutions due to their reliance on limited local information.

To address these challenges, we propose a novel and efficient DGL model that incorporates global information into the features of local candidate nodes, which are then processed by a decoder to construct the solution. This strategy combines the computational efficiency of local selection with the broader perspective provided by global information. Furthermore, instead of using static embeddings to predict the next node, DGL dynamically updates global features at each step of the solution process, offering a more comprehensive representation of the current state and improving the model's ability to capture state-dependent changes.

Information Aggregation from Local and Global Views

Local View. A common observation in solving VRPs is that the optimal action (i.e., selecting the next node) lies within a small local neighborhood of the current node [Helsgaun, 2017]. This local neighborhood pattern has been consistently observed across VRPs with diverse scales and distributions [Gao *et al.*, 2024]. To leverage this property, we focus DGL's action space on the local k-nearest neighbors, capturing local invariance across different VRPs and then enhancing model generalization capabilities.

Global View. However, focusing solely on local view often leads to suboptimal solutions, as shown by our ablation study (Section 4). Conversely, simply using the features of all nodes to capture global information as previous methods disrupts local invariance and significantly increases computational cost, making it impractical for large-scale problems. Consequently, the primary challenge lies in integrating local selection with global information effectively. The key challenge, therefore, is effectively integrating local selection with global information. To address this, we incorporate four global features—spatial relationships, distribution patterns, and relative positioning—into the candidate node representation within the local selection range.

1. *Distance to the current node*: We prioritize candidate nodes based on their distance to the current node to minimize cost and prevent distant routing. The global feature *dist* for candidate node i is computed as the euclidean distance $\mathrm{d}(i,j)$ between nodes i and j. π_t denotes the current node at step t.

2. *Average distance to all unvisited nodes*: This feature allows the model to anticipate the impact of future steps by considering the spatial distribution of unvisited nodes. By integrating this information, we guide the model to consider not only the immediate next step but also the broader VRP topology. We compute this feature for candidate node i as:

$$\mu_i = \frac{\sum_{j=1}^{n} m(j) \cdot \mathrm{d}(i,j)}{\sum_{j=1}^{n} m(j)}, \tag{1}$$

where $m(i) = 1$ if node i has been visited and $m(i) = 0$ otherwise.

3. *Standard deviation of distances to unvisited nodes*: We consider the variability in distances from candidate node i to unvisited nodes. A higher standard deviation indicates greater variability in the node's spatial relationships, which could affect routing efficiency. We compute this feature as follows:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{n} m(j) \cdot \left(\mathrm{d}(i,j) - \mu_i\right)^2}{\sum_{j=1}^{n} m(j)}}. \tag{2}$$

4. *Directional vector to the starting node (or depot in CVRP)*: We include the directional information from the candidate node to the starting point (or depot) to align the model's decisions with the overall route direction, helping it avoid detours and move efficiently toward the destination (i.e., the starting node). The global feature for candidate node i is computed as follows:

$$\mathit{dire}_i = \frac{[v_i^1 - v_{\pi_0}^1, \ldots, v_i^h - v_{\pi_0}^h]}{\mathrm{d}(i,\pi_0)}, \tag{3}$$

where π_0 denotes the starting node, v_i^j denotes the coordinates of the jth dimension of node i, and $[\cdot,\cdot]$ denotes the horizontal concatenation operator.

The distance-related features 1–3 provide insights into node positions within the VRP topology, ensuring the model can assess both local proximity and global distribution. The directional feature 4 adds orientation awareness, ensuring decisions align with the optimal path from the starting node. By integrating these position and orientation features into the k-nearest candidate nodes, DGL effectively maintains a global perspective when deciding the next node in the solution.

Finally, we concatenate these features for the k-nearest neighbor node i of the current node to form the model inputs as follows:

$$x_i = [v_i, \mathit{dist}_i, \mu_i, \sigma_i, \mathit{dire}_i]. \tag{4}$$

Normalization. To unify the scale of features and improve the robustness of the model, we normalize the node's features as follows:

$$\hat{x_i} = \frac{x_i - \min(x)}{\max(x) - \min(x)}. \tag{5}$$

Efficient Dynamic Feature Update. As the solution is constructed, the state at each step evolves, and the corresponding global features should be updated accordingly. However, recalculating the global features 2 and 3 at each step is computationally expensive and time-consuming due to their quadratic time complexity ($\mathcal{O}(n^2)$). To address this issue, we propose using a recursive equation to update the mean and variance at each step $t < n$, preserving accuracy while reducing the time complexity to $\mathcal{O}(n)$ as follows:

$$\mu_i^{(t)} = \frac{(n-t+1)\mu_i^{(t-1)} - \mathrm{d}(i,\pi_t)}{n-t}, \tag{6}$$

$$\sigma_i^{(t)} = \sqrt{\frac{(n-t+1)(\sigma_i^{(t-1)})^2 - (\mathrm{d}(i,\pi_t) - \mu_i^{(t-1)})^2}{n-t}}. \tag{7}$$

Model Architecture

Unlike conventional encoder-decoder architectures, we adopt a decoder-only architecture to better capture the dynamic properties of the state during VRP solution construction. As demonstrated by the design of the GPT family [Chen *et al.*, 2024], decoders, compared to encoders, are more effective in capturing dynamic patterns from the changing state, leading to improved generalization. Specifically, we use the features of the k-nearest nodes, processed via dynamic feature computation, as input. These features are then mapped to a

probability distribution for the next node selection through L attention layers, as shown in eq. 9.

The initial embeddings are generated using a linear layer, expressed as $\{s_i^1\}_{i=1}^k = \{W\hat{x}_i\}_{i=1}^k$, where W denotes learnable parameters. The embedding at the first attention layer is then defined as follows:

$$S^1 = [Ws_{\pi_t}, s_1^1, \ldots, s_k^1], \tag{8}$$

where s_{π_t} denotes the embeddings of current node π_t at step t. Notably, instead of explicitly including the starting node information in the input as done in conventional methods [Drakulic *et al.*, 2023], we implicitly encode it into the global feature representation, i.e., through features 3 and 4. This design choice is based on the observation that the starting node often falls outside the local view of the current node after several steps. Explicit inclusion of the starting node could therefore interfere with the local focus on the current node.

Finally, DGL maps these embeddings to the probability distribution of each candidate node being the next node through L attention layers $\mathrm{Attnlayer}$ as follows:

$$S^l = \mathrm{Attnlayer}(S^{l-1}), l \in \{1, \ldots, L\}, \tag{9}$$

$$p_i(\theta) = \mathrm{Softmax}(Ws_i^L), \tag{10}$$

where θ denotes model parameters. In each step of t, DGL selects an unvisited node π_t according to θ, while masking invalid nodes (those already visited or exceeding capacity constraints) to ensure solution feasibility. This process continues until the tour is complete.

3.3 Replacement-based SIL Method

Considering the limitations of conventional learning paradigms (see Section 2.2), we propose a replacement-based SIL method to improve the diversity of the dataset through data augmentation and enhance DGL's robustness by randomly removing portions of the generated dataset during training. We outline the five key steps as follows:

Initialization. We randomly generate B VRP instances from a uniform distribution $\mathcal{X}$ and use a simple nearest-neighbor heuristic algorithm to generate initial solutions, which serve as initial pseudo-labels to warm up DGL.

Sampling. At each update step, we employ the current policy to sample M solutions for each instance via beam search. If the best solution among the M sampled solutions improves upon the current best solution for a given instance, it replaces the current solution.

Data Augmentation. To enhance model robustness, we apply data augmentation techniques to both the problem and solution sides, thereby increasing the diversity of training instances. For the problem side, we use transformations such as rotation, reflection, and normalization on the VRP topology. On the solution side, we use problem-specific enhancement methods: for TSP, considering the solution forms a cyclic sequence, we randomly apply an integer offset ϵ to the solution, i.e., $\pi_i = \pi_{i+\epsilon}$; for CVRP, considering the solution comprises multiple sub-tours, we randomly flip sub-tours and swap the order of adjacent sub-tours. The augmented problem and solution then replace the current dataset.

Algorithm 1 Replacement-based SIL method of DGL

Require: Uniform distribution $\mathcal{X}$, number of epochs E, batch size B, sampling number M, replacement rate α.

Initialize policy θ
$x^i \leftarrow$ Generate B instances from $\mathcal{X} \quad \forall i \in \{1, \ldots, B\}$
$\pi^i \leftarrow$ Nearest-neighbor$(x^i) \quad \forall i \in \{1, \ldots, B\}$
for $e = 1, \ldots, E$ **do**
  for $i = 1, \ldots, B$ **do**
    $\pi^j \leftarrow$ Sampling(**DGL**, $x^i, M, \theta) \quad \forall j \in \{1, \ldots, M\}$
    if $\pi^i =$ Null **or** $\mathrm{c}(\pi^i) > \min_{j \in \{1,2,\ldots,M\}} \mathrm{c}(\pi^j)$ **then**
      $\pi^i \leftarrow \pi^j$
    end if
  end for
  $x^i, \pi^i \leftarrow$ Augmentation$(x^i, \pi^i) \quad \forall i \in \{1, \ldots, B\}$
  $\theta \leftarrow$ **DGL**$(x^i, \pi^i, \theta) \quad \forall i \in \{1, \ldots, B\}$
  $\hat{x^j} \leftarrow$ Generate $\alpha \cdot B$ instances from $\mathcal{X} \quad \forall j \in \{1, \ldots, \alpha \cdot B\}$
  $j \leftarrow$ Randomly select $\alpha \cdot B$ indices $\quad \forall j \in \{1, \ldots, \alpha \cdot B\}$
  $x^j, \pi^j \leftarrow \hat{x^j}$, Null $\quad \forall i \in \{1, \ldots, \alpha \cdot B\}$
end for

Imitation Learning. Using the augmented dataset as training instances with pseudo-labels y_i, we apply the cross-entropy loss function $\mathcal{L}(\theta) = -\sum_{i=1}^n y_i \log(p_i(\theta))$ to maximize the likelihood of selecting the current best node at each step.

Random Replacement. To mitigate overfitting to a fixed dataset, which often impedes generalization to unseen scenarios, we propose a data replacement strategy that reduces the model's dependence on specific instances, promoting more flexible and generalizable learning. Specifically, we randomly replace a fraction α of the training instances with resampled data and set the solution length of these instances to infinity during the subsequent update. This introduces controlled variability into the dataset and encourages the model to learn from a broader range of data distributions, thereby enhancing its generalization capability.

In contrast to existing SIL methods that rely on static datasets without replacement, our approach eliminates the need for a separate validation set. This enables the model to improve continuously throughout training, avoiding the computational overhead of cross-validation. Consequently, our method streamlines the training process and boosts the model's robustness by exposing it to a more diverse range of data variations. We present the overall SIL strategy of DGL in Algorithm 1. The source code of DGL is accessible online[1].

4 Experimental Results

In this section, we conduct extensive experiments on both synthetic and real-world dataset to evaluate the generalization performance of our DGL compared to baselines.

DGL Settings. To demonstrate superior generalization ability of our method, we train a single model exclusively on ran-

[1]https://github.com/wuyuesong/DGL

Distribution	**Uniform**				**Clustered**			
Category	**TSP-100**	**TSP-1000**	**TSP-5000**	**TSP-10000**	**TSP-100**	**TSP-1000**	**TSP-5000**	**TSP-10000**
Measurements	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time
(Near-)Optimality	0.00, 23.8m	0.00, 16.3h	0.00, 2.1h	0.00, 1.4d	0.00, 34.4m	0.00, 16.9h	0.00, 4.6h	0.00, 1.7d
Omni-TSP(ICML'23)	1.28, 8.4s	19.56, 1.3m	49.43, 16.1m	61.39, 2.0h	1.38, 8.3s	21.23, 1.3m	54.49, 16.1m	71.60, 2.0h
ELG(IJCAI'24)	**0.24**, 23.5s	12.22, 13.4s	18.84, 39.7s	18.32, 5.6m	1.45, 23.3s	15.27, 13.4s	25.74, 39.6s	31.01, 5.6m
INVIT(ICML'24)	2.08, 46.8s	5.13, 2.6m	6.49, 6.6m	4.82, 20.2m	3.05, 48.6s	6.39, 2.5m	7.20, 7.0m	6.02, 20.8m
UDC(NeurIPS'24)	0.40, 1.4m	2.06, 1.9m	6.99, 2.0m	8.73, 3.7m	2.54, 1.4m	8.26, 1.9m	15.19, 2.0m	15.41, 3.1m
LEHD(NeurIPS'23)	0.57, 4.8s	2.76, 1.9m	15.80, 18.2m	24.10, 2.3h	1.83, 4.2s	8.56, 1.9m	23.46, 18.2m	35.33, 2.3h
BQ(NeurIPS'23)	5.14, 13.5s	3.82, 9.6m	12.68, 1.9h	18.74, 13.5h	5.73, 13.0s	9.43, 9.6m	27.65, 1.9h	41.80, 13.5h
GD(TMLR'24)	0.72, 1.0m	4.26, 11.8m	60.26, 1.1h	198.65, 6.7h	2.29, 1.0m	25.26, 11.8m	329.10, 1.1h	627.83, 6.8h
DGL(ours)	0.60, 1.1m	2.36, 41.4s	5.20, 38.4s	4.46, 5.9m	1.76, 1.1m	4.26, 41.4s	6.12, 38.4s	6.02, 5.9m
DGL(ours)+BS(4)	0.27, 4.8m	**1.47**, 2.7m	**3.42**, 1.9m	**2.58**, 6.4m	**1.19**, 4.8m	**3.24**, 4.4m	**4.81**, 1.9m	**4.49**, 6.4m
Distribution	**Explosion**				**Implosion**			
Category	**TSP-100**	**TSP-1000**	**TSP-5000**	**TSP-10000**	**TSP-100**	**TSP-1000**	**TSP-5000**	**TSP-10000**
Measurements	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time
(Near-)Optimality	0.00, 28.3m	0.00, 17.5h	0.00, 2.0h	0.00, 1.4d	0.00, 28.7m	0.00, 17.5h	0.00, 3.5h	0.00, 1.4d
Omni-TSP(ICML'23)	1.22, 8.3s	19.96, 1.3m	51.28, 16.1m	65.37, 2.0h	1.21, 8.3s	19.20, 1.3m	50.37, 16.1m	62.58, 2.0h
ELG(IJCAI'24)	0.36, 23.3s	13.67, 13.4s	22.79, 39.7s	23.45, 5.6m	**0.31**, 23.4s	12.40, 13.4s	18.95, 39.7s	18.73, 5.6m
INVIT(ICML'24)	2.24, 49.2s	7.85, 2.6m	10.04, 6.9m	8.70, 20.2m	2.37, 45.0s	5.92, 2.5m	6.79, 6.3m	5.31, 19.3m
UDC(NeurIPS'24)	0.66, 1.4m	6.96, 1.9m	16.15, 2.0m	17.44, 2.4m	0.54, 1.4m	3.74, 1.9m	7.74, 2.0m	10.04, 2.4m
LEHD(NeurIPS'23)	0.67, 4.2s	5.76, 1.8m	21.07, 18.2m	30.55, 2.3h	1.11, 4.2s	4.10, 1.8m	17.48, 18.2m	26.46, 2.3h
BQ(NeurIPS'23)	5.57, 12.9s	7.11, 9.6m	29.39, 1.9h	51.54, 13.5h	5.63, 12.9s	5.22, 9.6m	16.42, 1.9h	25.23, 13.5h
GD(TMLR'24)	0.68, 1.0m	12.33, 11.8m	271.55, 1.1h	682.40, 6.7h	1.45, 1.1m	8.68, 11.9m	100.05, 1.0h	259.46, 6.7h
DGL(ours)	0.72, 1.1m	4.25, 41.4s	7.56, 38.4s	6.87, 5.9m	0.94, 1.1m	3.14, 41.4s	5.45, 38.4s	4.92, 5.9m
DGL(ours)+BS(4)	**0.35**, 4.8m	**3.43**, 2.7m	**6.07**, 1.9m	**5.22**, 6.4m	0.60, 4.8m	**2.23**, 4.4m	**3.68**, 1.9m	**3.11**, 6.4m

Table 1: Performance comparison of different methods on synthetic TSP instances. Symbol"BS(4)" denotes Beam search with width of 4. The best and second results are bolded and underlined, respectively. Furthermore, UDC fails to solve TSPs under 100 nodes and CVRPs under 200 nodes due to unknown errors.

domly generated instances with 100 nodes and a uniform distribution, then directly evaluate it on test instances of varying sizes and distributions. Our proposed DGL consists of 4 decoder layers with a hidden dimension of 128 and 8 attention heads. The learning rate is 1e-4 with a decay rate of 0.97. Training spans 100 epochs with 100 iterations per epoch, a batch size of 256, 64 samples per iteration, and a replacement rate of 12.5%. The hyperparameter k is set to 30 for TSP and 50 for CVRP.

Baselines Settings. We select seven SOTA models for comparison, including 1) RL-based models: Omni-VRP [Zhou *et al.*, 2023], ELG [Gao *et al.*, 2024], INVIT-3V [Fang *et al.*, 2024] and UDC [Zheng *et al.*, 2024]; 2) SL-based models: LEHD [Luo *et al.*, 2023] and BQ [Drakulic *et al.*, 2023]; and 3) SIL based models: GD [Pirnay and Grimm, 2024]. We follow the publicly available pre-trained parameters and default settings for all baselines. We report results for LEHD and BQ with the greedy search for a fair comparison in terms of computational efficiency. We apply the same inference setup for both DGL and INVIT. All experiments were conducted on an Intel(R) Xeon(R) Platinum 8352V CPU and an NVIDIA RTX 4090 GPU (24GB).

Dataset. For the synthetic data, we adopt the MSVDRP dataset [Fang *et al.*, 2024], which comprises 16 subsets for TSP, encompassing 4 distributions (uniform, clustered, explosion, and implosion) and 4 scales (TSP-100, TSP-1000, TSP-5000 and TSP-10000), and 12 subsets for CVRP under the same distributions but at three scales (CVRP-50, CVRP-500, and CVRP-5000).

For real-world data, we adopt the widely recognized TSPLIB and CVRPLIB benchmarks. TSPLIB95 contains 77 instances varying in scale from 51 to 18512. CVRPLIB Set-X [Uchoa *et al.*, 2017] contains 100 instances varying in scale from 100 to 1,000. Furthermore, we test on eight country maps to further evaluate model's performance in real-world scenarios with city counts from 194 to 22775.

For all baselines and DGL, we report their average gap to the (near-)optimal solutions, i.e., LKH results for synthetic data and the best known solutions for real-world data.

Performance on Synthetic Datasets. We compare DGL against baseline models on synthetic datasets. The key findings are as follows: 1) Most models experience performance degradation when generalizing to VRPs with varying scales and distributions. 2) As shown in Table 1, DGL achieves state-of-the-art performance compared to global-view-based baselines like Omni-VRP, LEHD, and BQ, demonstrating the effectiveness of local selection. 3) DGL outperforms INViT in both solution quality and inference speed, leveraging global information and dynamically updating features to avoid local optima and better adapt to problem variations. 4) Compared to other SIL models like GD, DGL with replacement-based SIL significantly improves solution quality. 5) The increased number of nodes leads to more complex route patterns and greater diversity (CVRP-5000 in Table 2). DGL efficiently integrating information and handling this increased complexity compared to baselines.

Performance on Real-world Datesets. We compare DGL against baseline models on TSPLIB, CVRPLIB, and Country maps datasets (see Table 3). On these real-world instances, DGL consistently demonstrates superior generalization performance compared to the baselines. Real-world instances have more complex distributions than synthetic datasets, and DGL effectively integrates global information via dynamic feature computation while leveraging local invariance to han-

Distribution	**Uniform**			**Clustered**		
Category	**CVRP-50**	**CVRP-500**	**CVRP-5000**	**CVRP-50**	**CVRP-500**	**CVRP-5000**
Measurements	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time
(Near-)Optimality	0.00, 5.1h	0.00, 1.1d	0.00, 3.3d	0.00, 5.7h	0.00, 2.3d	0.00, 3.3d
Omni-TSP(ICML'23)	4.04, 2.4s	6.32, 1.6m	48.74, 21.5m	**2.81**, 2.1s	**4.43**, 1.6m	24.21, 20.8m
ELG(IJCAI'24)	8.88, 6.9s	7.86, 12.5s	8.20, 4.7m	13.49, 6.8s	12.17, 12.6s	25.34, 4.7m
INVIT(ICML'24)	5.06, 1.9m	9.57, 2.4m	7.62, 9.8m	5.15, 1.9m	9.15, 2.4m	7.48, 10.3m
UDC(NeurIPS'24)	N/A	5.62, 7.4m	3.33, 16.7m	N/A	4.51, 7.4m	3.46, 16.6m
LEHD(NeurIPS'23)	5.29, 2.4s	3.63, 1.8s	7.86, 19.4m	4.91, 1.8s	5.00, 1.8s	24.00, 19.3m
BQ(NeurIPS'23)	**3.18**, 4.4s	**3.31**, 1.4m	5.19, 2.0h	3.12, 4.0s	4.54, 1.3m	14.60, 2.0h
GD(TMLR'24)	5.40, 19.1s	8.71, 1.5m	13.48, 4.3m	5.55, 19.2s	9.87, 1.5m	16.65, 4.3m
DGL(ours)	4.69, 1.1m	8.38, 1.2m	3.08, 1.2m	4.33, 1.1m	6.26, 1.2m	2.35, 1.2m
DGL(ours)+BS(4)	3.29, 4.5m	7.20, 4.5m	**2.83**, 3.3m	3.23, 4.5m	5.50, 4.5m	**2.22**, 3.3m
Distribution	**Explosion**			**Implosion**		
Category	**CVRP-50**	**CVRP-500**	**CVRP-5000**	**CVRP-50**	**CVRP-500**	**CVRP-5000**
Measurements	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time	gap(%)↓, time
(Near-)Optimality	0.00, 5.0h	0.00, 2.8d	0.00, 3.3d	0.00, 5.0h	0.00, 2.0d	0.00, 3.3d
Omni-TSP(ICML'23)	3.76, 2.1s	6.16, 1.6m	34.62, 21.1m	3.89, 2.1s	5.77, 1.6m	41.04, 21.0m
ELG(IJCAI'24)	9.76, 6.6s	8.50, 12.1s	17.92, 4.7m	9.56, 6.7s	7.97, 12.1s	9.68, 4.6m
INVIT(ICML'24)	5.34, 1.9m	9.58, 2.4m	7.69, 9.7m	5.27, 1.9m	9.14, 2.4m	7.20, 9.8m
UDC(NeurIPS'24)	N/A	5.57, 7.4m	3.65, 16.5m	N/A	5.45, 7.4m	3.34, 16.7m
LEHD(NeurIPS'23)	5.20, 1.8s	**4.27**, 1.8s	16.78, 19.4m	5.56, 1.8s	4.70, 1.8s	16.28, 19.4m
BQ(NeurIPS'23)	3.26, 4.0s	4.50, 1.3m	14.94, 2.0h	3.48, 4.0s	**3.87**, 1.3m	6.47, 2.0h
GD(TMLR'24)	5.66, 19.1s	10.06, 1.5m	19.84, 4.3m	5.73, 19.1s	9.50, 1.5m	13.74, 4.3m
DGL(ours)	4.40, 1.1m	8.09, 1.2m	3.20, 1.3m	4.46, 1.1m	7.95, 1.2m	2.77, 1.3m
DGL(ours)+BS(4)	**3.10**, 4.5m	7.03, 4.5m	**2.98**, 3.3m	**3.22**, 4.5m	6.90, 4.6m	**2.54**, 4.3m

Table 2: Performance comparison of different methods on synthetic CVRP instances

Node number	**TSPLIB**			**Country Maps**							
(Count)	≤1000 (48)	1001-10000 (24)	>10000 (5)	**QA194**	**UY734**	**MU1979**	**EG7146**	**YM7663**	**AR9152**	**GR9882**	**VM22775**
Omni-TSP	9.66%	34.89%	OOM	10.44%	14.83%	52.06%	151.05%	79.25%	72.70%	70.10%	OOM
ELG	4.39%	12.99%	OOM	7.06%	10.77%	22.54%	209.58%	60.21%	21.66%	23.95%	OOM
INVIT	3.20%	7.63%	7.57%	2.88%	5.38%	13.06%	12.88%	13.37%	12.63%	13.25%	9.75%
UDC	3.10% (42)	14.11%	23.36% (1)	0.58%	5.54%	15.41%	27.11%	28.03%	29.17%	19.44%	OOM
LEHD	2.51%	12.34%	50.21% (3)	27.15%	20.98%	42.65%	42.15%	93.84%	56.54%	74.69%	OOM
BQ	8.42%	14.50%	45.21%(1)	12.18%	9.26%	48.92%	170.87%	82.37%	64.43%	78.22%	OOM
GD	4.19%	74.95%	991.24%	236.40%	284.43%	1351.43%	1281.81%	3632.62%	1753.86%	2245.37%	OOM
DGL	2.42%	6.49%	8.24%	1.27%	3.11%	10.53%	8.56%	11.10%	**9.51%**	10.35%	7.78%
DGL+BS(4)	**2.23%**	**5.64%**	**5.77%**	**0.50%**	**1.55%**	**10.46%**	**7.60%**	**10.21%**	12.46%	**9.11%**	**6.54%**

Table 3: Performance comparison of different methods on real-world instances. Symbol "OOM" is used to indicate cases where the model fails to solve all instances in the set due to GPU memory constraints. Symbol "($\underline{i}$)" denotes the number of instances the model successfully solves in this set. In Country Maps, the first two character **EG** is the abbreviation of 'Egypt' and **7146** is the node number.

dle these complexities, showcasing its robust applicability in unseen and pratical scenarios.

Ablation Studies. We conduct extensive ablation studies to evaluate the design of DGL, focusing on: 1) the effectiveness of the four global features and 2) the sensitivity to the replacement rate α and local range k. (Please contact the first author for the detailed experimental results).

The results of the ablation study on global features indicate that their inclusion significantly enhances performance, while their exclusion results in notable degradation, underscoring their critical importance. Sensitivity analysis for α shows that performance remains stable across configurations, except when $\alpha = 0$, where a sharp decline is observed, validating the necessity of the replace operation. Similarly, varying k demonstrates DGL's robustness across different local range settings.

5 Conclusion

This study presents DGL, a novel approach that integrates global and local information to effectively solve VRPs. Extensive experimental evaluations on VRPs of varying sizes and distribution patterns demonstrate DGL's superior generalization capabilities and its promise for real-world applications. In future work, we aim to further improve DGL's computational efficiency. Additionally, we plan to leverage learning-based methods to automatically derive and better fuse global features, and to extend DGL to more complex combinatorial optimization problems.

Acknowledgments

This work is supported by the Jilin Provincial Department of Science and Technology Project (20240302086GX). Any opinions, findings and conclusions or recommendations ex-

pressed in this material are those of authors and do not reflect the views of the funding agencies.

Contribution Statement

Yubin Xiao, Yuesong Wu, and Rui Cao made equal contributions to this study. You Zhou and Yuan Jiang serve as the corresponding authors.

References

[Alkaya and Duman, 2013] Ali Fuat Alkaya and Ekrem Duman. Application of sequence-dependent traveling salesman problem in printed circuit board assembly. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 3(6):1063–1076, 2013.

[Applegate *et al.*, 2007] David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William John Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.

[Bdeir *et al.*, 2023] Ahmad Bdeir, Jonas K. Falkner, and Lars Schmidt-Thieme. Attention, filling in the gaps for generalization in routing problems. In *Proceedings of Machine Learning and Knowledge Discovery in Databases*, pages 505–520, 2023.

[Bi *et al.*, 2022] Jieyi Bi, Yining Ma, Jiahai Wang, Zhiguang Cao, Jinbiao Chen, Yuan Sun, and Yeow Meng Chee. Learning generalizable models for vehicle routing problems via knowledge distillation. In *Proceedings of Advances in Neural Information Processing Systems*, pages 31226–31238, 2022.

[Chen *et al.*, 2024] Yuxuan Chen, Guangsheng Ou, Mingwei Liu, Yanlin Wang, and Zibin Zheng. Are decoder-only large language models the silver bullet for code search?, 2024.

[Drakulic *et al.*, 2023] Darko Drakulic, Sofia Michel, Florian Mai, Arnaud Sors, and Jean-Marc Andreoli. BQ-NCO: Bisimulation quotienting for efficient neural combinatorial optimization. In *Proceedings of Advances in Neural Information Processing Systems*, pages 77416–77429, 2023.

[Fang *et al.*, 2024] Han Fang, Zhihao Song, Paul Weng, and Yutong Ban. INViT: A generalizable routing problem solver with invariant nested view transformer. In *Proceedings of International Conference on Machine Learning*, volume 235, pages 12973–12992, 2024.

[Gao *et al.*, 2024] Chengrui Gao, Haopu Shang, Ke Xue, Dong Li, and Chao Qian. Towards generalizable neural solvers for vehicle routing problems via ensemble with transferrable local policy. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 6914–6922, 2024.

[Geisler *et al.*, 2022] Simon Geisler, Johanna Sommer, Jan Schuchardt, Aleksandar Bojchevski, and Stephan Günnemann. Generalization of neural combinatorial solvers through the lens of adversarial robustness. In *Proceedings of International Conference on Learning Representations*, 2022.

[Goh *et al.*, 2024] Yong Liang Goh, Zhiguang Cao, Yining Ma, Yanfei Dong, Mohammed Haroon Dupty, and Wee Sun Lee. Hierarchical neural constructive solver for real-world tsp scenarios. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 884–895, 2024.

[Helsgaun, 2017] Keld Helsgaun. An extension of the Lin–Kernighan–Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, pages 24–50, 2017.

[Jiang *et al.*, 2022] Yuan Jiang, Yaoxin Wu, Zhiguang Cao, and Jie Zhang. Learning to solve routing problems via distributionally robust optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9786–9794, 2022.

[Joshi *et al.*, 2022] Chaitanya K. Joshi, Quentin Cappart, Louis-Martin Rousseau, and Thomas Laurent. Learning the travelling salesperson problem requires rethinking generalization. *Constraints*, 27:70–98, 2022.

[Kim *et al.*, 2015] Gitae Kim, Yew-Soon Ong, Chen Kim Heng, Puay Siew Tan, and Nengsheng Allan Zhang. City vehicle routing problem (City VRP): A review. *IEEE Transactions on Intelligent Transportation Systems*, 16:1654–1666, 2015.

[Kool *et al.*, 2019] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *Proceedings of International Conference on Learning Representations*, 2019.

[Kool *et al.*, 2022] Wouter Kool, Herke van Hoof, Joaquim Gromicho, and Max Welling. Deep policy dynamic programming for vehicle routing problems. In *Proceedings of International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 190–213, 2022.

[Kwon *et al.*, 2020] Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems*, pages 21188–21198, 2020.

[Li *et al.*, 2021] Sirui Li, Zhongxia Yan, and Cathy Wu. Learning to delegate for large-scale vehicle routing. In *Proceedings of Advances in Neural Information Processing Systems*, volume 34, pages 26198–26211, 2021.

[Li *et al.*, 2024] Yuanshu Li, Yubin Xiao, Xuan Wu, Lei Song, Yanchun Liang, and You Zhou. Leveraging hierarchical similarities for contrastive clustering. In *Proceedings of Neural Information Processing*, pages 148–168, 2024.

[Luo *et al.*, 2023] Fu Luo, Xi Lin, Fei Liu, Qingfu Zhang, and Zhenkun Wang. Neural combinatorial optimization with heavy decoder: Toward large scale generalization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 36, pages 8845–8864, 2023.

[Luo *et al.*, 2024] Fu Luo, Xi Lin, Zhenkun Wang, Xialiang Tong, Mingxuan Yuan, and Qingfu Zhang. Self-improved learning for scalable neural combinatorial optimization, 2024. arXiv:2403.19561.

[Lyu *et al.*, 2024] Zefeng Lyu, Md. Zahidul Islam, and Andrew Junfang Yu. A scalable and adaptable supervised learning approach for solving the traveling salesman problems. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):17092–17104, 2024.

[Min *et al.*, 2023] Yimeng Min, Yiwei Bai, and Carla P Gomes. Unsupervised learning for solving the travelling salesman problem. In *Proceedings of Advances in Neural Information Processing Systems*, volume 36, pages 47264–47278, 2023.

[Pirnay and Grimm, 2024] Jonathan Pirnay and Dominik G. Grimm. Self-improvement for neural combinatorial optimization: Sample without replacement, but improvement. *Transactions on Machine Learning Research*, 2024.

[Qiu *et al.*, 2022] Ruizhong Qiu, Zhiqing Sun, and Yiming Yang. Dimes: A differentiable meta solver for combinatorial optimization problems. In *Proceedings of Advances in Neural Information Processing Systems*, volume 35, pages 25531–25546, 2022.

[Sun and Yang, 2023] Zhiqing Sun and Yiming Yang. DIFUSCO: Graph-based diffusion solvers for combinatorial optimization. In *Proceedings of Advances in Neural Information Processing Systems*, volume 36, pages 3706–3731, 2023.

[Uchoa *et al.*, 2017] Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017.

[Wu *et al.*, 2024] Xuan Wu, Di Wang, Lijie Wen, Yubin Xiao, Chunguo Wu, Yuesong Wu, Chaoyu Yu, Douglas L. Maskell, and You Zhou. Neural combinatorial optimization algorithms for solving vehicle routing problems: A comprehensive survey with perspectives, 2024. arXiv:2406.00415.

[Xia *et al.*, 2024] Yifan Xia, Xianliang Yang, Zichuan Liu, Zhihao Liu, Lei Song, and Jiang Bian. Position: Rethinking post-hoc search-based neural approaches for solving large-scale traveling salesman problems. In *Proceedings of International Conference on Machine Learning*, pages 54178–54190, 2024.

[Xiao *et al.*, 2020] Yubin Xiao, Zheng Xiao, Xiang Feng, Zhiping Chen, Linai Kuang, and Lei Wang. A novel computational model for predicting potential lncRNA-disease associations based on both direct and indirect features of lncRNA-disease pairs. *BMC Bioinformatics*, 21:1–22, 2020.

[Xiao *et al.*, 2024a] Yubin Xiao, Di Wang, Boyang Li, Huanhuan Chen, Wei Pang, Xuan Wu, Hao Li, Dong Xu, Yanchun Liang, and You Zhou. Reinforcement learning-based non-autoregressive solver for traveling salesman problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Xiao *et al.*, 2024b] Yubin Xiao, Di Wang, Boyang Li, Mingzhao Wang, Xuan Wu, Changliang Zhou, and You Zhou. Distilling autoregressive models to obtain high-performance non-autoregressive solvers for vehicle routing problems with faster inference speed. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20274–20283, 2024.

[Xiao *et al.*, 2025] Yubin Xiao, Di Wang, Xuan Wu, Yuesong Wu, Boyang Li, Wei Du, Liupu Wang, and You Zhou. Improving generalization of neural vehicle routing problem solvers through the lens of model architecture. *Neural Networks*, 187:107380, 2025.

[Ye *et al.*, 2024] Haoran Ye, Jiarui Wang, Helan Liang, Zhiguang Cao, Yong Li, and Fanzhang Li. GLOP: Learning global partition and local construction for solving large-scale routing problems in real-time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20284–20292, 2024.

[Zhang *et al.*, 2022] Zeyang Zhang, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning to solve travelling salesman problem with hardness-adaptive curriculum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9136–9144, 2022.

[Zheng *et al.*, 2024] Zhi Zheng, Changliang Zhou, Xialiang Tong, Mingxuan Yuan, and Zhenkun Wang. UDC: A unified neural divide-and-conquer framework for large-scale combinatorial optimization problems. In *Proceedings of Advances in Neural Information Processing Systems*, 2024.

[Zhou *et al.*, 2023] Jianan Zhou, Yaoxin Wu, Wen Song, Zhiguang Cao, and Jie Zhang. Towards omni-generalizable neural methods for vehicle routing problems. In *Proceedings of International Conference on Machine Learning*, volume 202, pages 42769–42789, 2023.

[Zong *et al.*, 2022] Zefang Zong, Hansen Wang, Jingwei Wang, Meng Zheng, and Yong Li. Rbg: Hierarchically solving large-scale routing problems in logistic systems via reinforcement learning. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 4648–4658, 2022.