Hybrid-Balance GFlowNet for Solving Vehicle Routing Problems

Ni Zhang, Zhiguang Cao*
School of Computing and Information Systems, Singapore Management University, Singapore
ni.zhang.2025@phdcs.smu.edu.sg, zgcao@smu.edu.sg

Abstract

Existing GFlowNet-based methods for vehicle routing problems (VRPs) typically employ Trajectory Balance (TB) to achieve global optimization but often neglect important aspects of local optimization. While Detailed Balance (DB) addresses local optimization more effectively, it alone falls short in solving VRPs, which inherently require holistic trajectory optimization. To address these limitations, we introduce the Hybrid-Balance GFlowNet (HBG) framework, which uniquely integrates TB and DB in a principled and adaptive manner by aligning their intrinsically complementary strengths. Additionally, we propose a specialized inference strategy for depot-centric scenarios like the Capacitated Vehicle Routing Problem (CVRP), leveraging the depot node's greater flexibility in selecting successors. Despite this specialization, HBG maintains broad applicability, extending effectively to problems without explicit depots, such as the Traveling Salesman Problem (TSP). We evaluate HBG by integrating it into two established GFlowNet-based solvers, i.e., AGFN and GFACS, and demonstrate consistent and significant improvements across both CVRP and TSP, underscoring the enhanced solution quality and generalization afforded by our approach.

1 Introduction

Vehicle Routing Problems (VRPs) are fundamental to real-world operations, including e-commerce logistics [54, 12, 39], urban delivery [56, 7, 21], supply chain management [11, 15, 8], and ride-sharing systems [18, 30, 44]. Efficient VRP solutions directly affect cost reduction, service quality, and overall performance in transportation and supply chain networks. Over the past decades, numerous heuristic and meta-heuristic algorithms, such as the Lin-Kernighan-Helsgaun algorithm [19], ant colony optimization (ACO) [3], hybrid genetic search [46], tabu search [2], and simulated annealing [36], have been developed to address the combinatorial complexity of VRPs. However, these approaches often depend on handcrafted rules and problem-specific heuristics, which limit their adaptability and scalability across diverse VRP instances. More recently, reinforcement learning and deep learning methods have emerged as promising alternatives [38, 5, 52, 20]. Models such as POMO [25], NeuOpt [31], and DEITSP [47] show potential in reducing dependence on handcrafted components. Yet, these methods still struggle to consistently achieve desirable performance, often becoming trapped in local optima due to the limited exploration capacity.

To improve exploration, recent work has explored the use of Generative Flow Network (GFlownet) [4], which generate diverse and high-quality solutions through a probabilistic, generative process. Unlike traditional learning-based approaches that focus on optimizing a single or a few trajectories, GFlowNet aims to learn a distribution over the solution space, making them well-suited for combinatorial problems like VRPs. However, current GFlowNet-based methods for VRPs such as GFACS [22] and AGFN [55], rely exclusively on global optimization during training. Particularly, they both adopt the

*corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Trajectory Balance (TB) objective [32], which effectively aligns with global metrics like minimizing total travel distance. However, this exclusive focus on global optimization can lead to the neglect of important local optimization signals. For instance, these methods may overlook reward dependencies between a current state and its successor, due to the lack of localized training objectives. As a result, they often struggle to capture fine-grained local structures in the solution space, limiting their ability to generate high-quality routes. On the other hand, Detailed Balance (DB) [4] offers a mechanism better suited for local optimization. Nevertheless, using DB alone is equally inadequate, as VRPs fundamentally require a global perspective to achieve optimal solutions. These limitations highlight the need for a broader approach that balances both local and global optimization. We propose that a Hybrid-Balance principle, combining TB and DB in a unified and extensible manner, can significantly enhance GFlowNet-based methods for VRPs.

Guided by this Hybrid-Balance principle, we introduce the Hybrid-Balance GFlowNet (HBG) framework for solving VRPs. First, HBG unifies DB, which promotes local optimization, with TB, which captures global optimization. To fully exploit their complementary strengths, we formulate a VRP-specific version of DB that effectively facilitates local optimization through localized objectives. We also design an adaptive integration mechanism that combines DB and TB in a way that respects their theoretical underpinnings, such as forward and backward transition probability, while leveraging their complementary benefits. Second, motivated by the insight that depot nodes in depot-centric VRPs, such as the Capacitated Vehicle Routing Problem (CVRP), have greater flexibility in selecting successor nodes, we propose a depot-guided inference strategy inspired by the Hybrid-Balance principle. Notably, even in depot-free scenarios like the Traveling Salesman Problem (TSP), our framework remains effective, as the Hybrid-Balance formulation is inherently general. Third, to demonstrate the broad applicability of HBG, we integrate it into two existing GFlowNet-based solvers, i.e., AGFN and GFACS, and observe consistent improvements in routing performance across both CVRP and TSP benchmarks. In summary, our main contributions are outlined as follows:

- We propose the Hybrid-Balance GFlowNet (HBG) framework for solving VRPs, which, for the first time, introduces and formalizes the concept of DB within the VRP context. Meanwhile, it unifies the principles of TB and DB through a principled and coherent integration to process both local and global optimizations.
- We design a depot-guided inference strategy to efficiently generate and explore high-quality trajectories, specifically tailored for problems involving a designated depot like the CVRP.
- We incorporate the HBG framework into existing GFlowNet-based methods for solving VRPs, i.e., AGFN and GFACS, and evaluate it on both synthetic and real-world datasets. The results demonstrate that our method significantly improves the performance of GFlowNet-based solvers for CVRP and TSP.

2 Related Works

2.1 Learning-Based Solvers for Vehicle Routing Problems

Learning-based approaches for VRPs can generally be divided into two categories: construction-based and improvement-based methods. Construction-based solvers generate complete solution trajectories in an end-to-end manner. A seminal example is the Attention Model (AM) [23], which first applied a Transformer architecture to solve VRPs. Building on AM, Policy Optimization with Multiple Optima (POMO) extends this approach by leveraging multiple optimal policies during training and inference to improve both solution quality and robustness. This line of work has since inspired a series of end-to-end construction-based methods [43, 26, 13, 47, 14] that further improve performance. Improvement-based methods, on the other hand, enhance initial solutions through iterative refinements. These methods often integrate neural networks into classical heuristic frameworks. Notable examples include NeuroLKH [49], DeepACO [51], and NeuOpt [31], which demonstrate strong performance through learning-augmented optimization strategies. To showcase the generality of our proposed framework, we apply it to enhance both a construction-based solver (AGFN) and an improvement-based solver (GFACS).

2.2 GFlowNet for Combinatorial Optimization Problems

GFlowNet has been applied across a wide range of structured generation and decision-making tasks. In molecular and drug discovery [57, 34, 24, 41, 16, 40], they are used to sample diverse, high-reward molecules from complex solution spaces. In causal structure learning [28, 9], GFlowNet facilitates exploration over multiple plausible directed acyclic graphs (DAGs), while in Bayesian inference [10, 42, 35], they serve as alternative samplers for discrete posteriors. Additional applications include symbolic reasoning [45, 27], robotics planning [29, 33], and solving maximum independent set (MIS) [53], where modeling solution diversity is essential. Recently, GFlowNet has also been applied to VRPs [55, 22], including TSP and CVRP. In this context, learning a distribution over feasible routes offers a flexible and effective alternative to deterministic solvers. Two representative models are AGFN and GFACS. AGFN incorporates adversarial training to improve trajectory construction in an end-to-end fashion, making it the first to apply GFlowNet to VRPs directly. In contrast, GFACS integrates GFlowNet with ant colony optimization, marking the first attempt to augment heuristic search with GFlowNet-based learning. In this paper, we further enhance both AGFN and GFACS for solving VRPs using our proposed Hybrid-Balance GFlowNet framework.

3 Hybrid-Balance GFlowNet

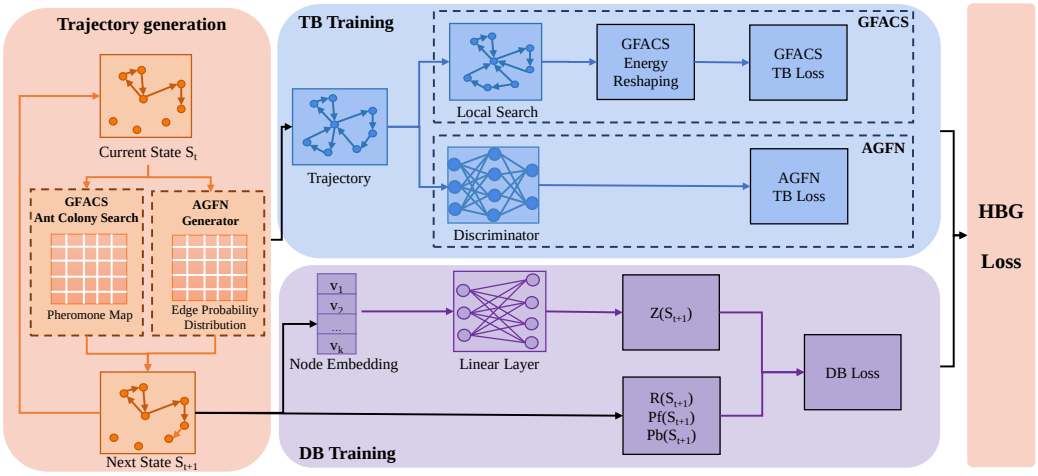

Figure 1: The Overall Framework of Our Hybrid-Balance GFlowNet for Solving VRPs.

As illustrated in Fig.1, the Hybrid-Balance GFlowNet (HBG) begins with trajectory generation using either AGFN or GFACS, during which state transition information is recorded at each step. The generated next state is then treated as the current state in the following step, and this process repeats until a complete trajectory is constructed. The blue region in Fig.1 corresponds to the original components from AGFN and GFACS, responsible for the processing of the complete trajectory and the computation of the TB loss, which captures global optimization signals. However, relying solely on the TB loss can cause the model to overlook important relationships between individual states. To bridge this gap, our proposed HBG introduces additional components, highlighted in purple, where the DB loss is computed for each individual state transition to enhance local optimization. The final training objective is a combination of two loss, which together guide the optimization of the model.

To better illustrate the motivation for introducing DB, consider a long vehicle routing trajectory that is incrementally constructed as A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E $\rightarrow$ ... $\rightarrow$ U $\rightarrow$ V $\rightarrow$ W $\rightarrow$ X $\rightarrow$ Y $\rightarrow$ Z. Assume this complete route yields a high total cost (bad performance), primarily due to suboptimal decisions made in the early stages, such as traversing a high-cost edge from node C to node D. In contrast, the latter portion of the tour (e.g., from node U to Z) may follow a more cost-effective and well-structured pattern. Under Trajectory Balance (TB), the final reward is determined by the overall trajectory cost, and is proportionally assigned to all transitions. Consequently, even high-quality local transitions, such as W $\rightarrow$ X $\rightarrow$ Y, may receive weak or misleading training signals simply because they are embedded in a globally suboptimal trajectory. This would hinder the model's ability to learn and reinforce desirable local patterns. On the other hand, Detailed Balance (DB) operates at a

step-wise granularity, evaluating the expected outcomes of individual transitions. For instance, at node W, DB can assess whether transitioning to X leads to better outcomes compared to alternative choices like Z, regardless of earlier suboptimal steps. This localized and reward-sensitive feedback enables the model to more accurately learn local quality from global performance, and promotes stronger learning signals for valuable decisions even within imperfect trajectories.

This example illustrates a core limitation of Trajectory Balance (TB) in long-horizon combinatorial tasks like VRP: when the overall trajectory is suboptimal, TB lacks the ability to identify and preserve well-structured local segments within it. As a result, valuable local patterns may be overlooked or penalized. By incorporating Detailed Balance (DB) into the training objective, we address this limitation by providing fine-grained, step-level singal that helps isolate and reinforce high-quality local decisions, even when the global trajectory does not show good performance.

3.1 Modeling Basics

Problem Definition. For a CVRP instance, $\mathcal{G}$ denotes the input graph, which includes the coordinates and demands of customers, as well as the depot location. Formally, the instance is represented as a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{U})$, where $\mathcal{V} = \{v_0, v_1, \ldots, v_n\}$ denotes the set of nodes, with v_0 as the depot and the remaining nodes representing customers, and $\mathcal{U}$ is the set of edges. Each customer node v_i ($i \geq 1$) is associated with a demand d_i and a location in Euclidean space. Each edge $(v_i, v_j) \in \mathcal{U}$ has an associated cost c_{ij}, typically defined as the Euclidean distance between v_i and v_j. The goal of CVRP is to determine a set of vehicle routes that start and end at the depot, such that each customer is visited exactly once, the total demand on each route does not exceed the vehicle's capacity, and the total routing cost is minimized.

State s: In a trajectory set $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_h\}$, the state s^i denotes the sequence of nodes visited in trajectory τ_i. At decision step t in τ_i, the state is defined as $s_t^i = \{x_0^i, x_1^i, x_2^i, \ldots, x_t^i\}$, where x_t^i is the most recently visited node, and x_0^i represents the depot which serves as both the starting and ending point of the route.

Action a: An action a_t^i transitions the system from state s_t^i to s_{t+1}^i. Given $s_t^i = \{x_0^i, x_1^i, \ldots, x_t^i\}$, the action selects the next node x_{t+1}^i from the set of unvisited nodes, adhering to feasibility constraints such as vehicle capacity. Once all customers are visited, the route terminates in a final state, forming a complete trajectory $\tau_i = \{x_0^i, x_1^i, \ldots, x_m^i\}$.

Reward R: The reward $R(\tau_i)$ is determined by the quality of the generated trajectory τ_i. We define two types of rewards: $R(\tau_i)$ and $R(s_t^i)$. The former, $R(\tau_i)$, evaluates the entire trajectory, while the latter, $R(s_t^i)$, reflects local reward signals at individual state transitions. These are defined as: $R(\tau_i) = \sum_{k=0}^{m-1} d(x_k^i, x_{k+1}^i)$, $R(s_t^i) = d(x_{t-1}^i, x_t^i)$, where $d(x_k^i, x_{k+1}^i)$ denotes the Euclidean distance between consecutive nodes.

Graph Neural Network (GNN). We integrate a GNN module [48] into the GFlowNet framework to more effectively capture the complex relational structures inherent in VRP instances. The detailed architecture and formulation are provided in Appendix A.1. Following the designs of AGFN and GFACS, we sparsify the fully connected graph into a k-nearest-neighbor graph $\mathcal{G}$ to improve scalability and reduce computational cost. The graph $\mathcal{G}$ is embedded into a high-dimensional feature space, encoding node coordinates and edge distances as node and edge features, respectively. The GNN, parameterized by $\boldsymbol{\theta}$, processes these features through multiple layers to produce rich representations. The resulting edge embeddings are passed through a multi-layer perceptron (MLP) to generate edge probability distribution $\eta(\mathcal{G}^*, \boldsymbol{\theta})$ for decision making by GFlowNet, while the node embeddings $\mathcal{Q} = \{q_1, q_2, \ldots, q_b\}$ are retained for computing state flows.

3.2 Hybrid-Balance GFlowNet

3.2.1 Global Optimization via TB

In VRPs, the objective is to determine the shortest route while satisfying various operational constraints, which necessitates evaluating solutions from a global perspective. Both AGFN and GFACS adopt the Trajectory Balance (TB) objective to address this requirement, as it enables the GFlowNet to be trained over entire trajectories, naturally aligning with global optimization goals.

As illustrated in Fig. 1, AGFN generates an edge probability distribution $\eta(\mathcal{G}^*, \boldsymbol{\theta}_{\text{generator}})$ using GFlowNet, which is then used to sample the next node in the route. A discriminator, trained with false labels from GFlowNet-generated trajectories and true labels from near-optimal trajectories, evaluates the quality of sampled trajectory set $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_h\}$. It assigns a quality score to each trajectory, which is then combined with the raw trajectory length $R(\tau)$ to compute the final AGFN reward $\widetilde{R}(\tau)$. These rewards $\widetilde{R}(\tau)$, along with the source flow $Z(\boldsymbol{\theta}_{\text{generator}})$, forward probability $P_F(\tau; \boldsymbol{\theta}_{\text{generator}})$, and backward probability $P_B(\tau)$ obtained from the GFlowNet, are used to compute the AGFN TB loss $\ell_{\text{TB}}^{\text{AG}}$, defined as:

$$\ell_{\text{TB}}^{\text{AG}}(\mathcal{T}; \boldsymbol{\theta}_{\text{generator}}) = \frac{1}{h} \sum_{k=1}^{h} \left(\log \frac{Z(\boldsymbol{\theta}_{\text{generator}}) * P_F(\tau_k; \boldsymbol{\theta}_{\text{generator}})}{\widetilde{R}(\tau_k) * P_B(\tau_k)} \right)^2. \tag{1}$$

For GFACS, the GFlowNet is used to generate a heuristic matrix $\eta(\mathcal{G}^*, \boldsymbol{\theta})$, which is subsequently transformed into a pheromone map to guide the ant colony optimization (ACO) in trajectory construction. Once the trajectories are generated, a local search is applied for refinement, followed by an energy reshaping step. The TB loss for GFACS, denoted $\ell_{\text{TB}}^{\text{GF}}$, is then computed in a similar form to AGFN, as both approaches adopt the TB loss formulation to optimize their models.

3.2.2 Local-Global Optimization through Hybrid-Balance

While global optimization is essential for solving VRPs, local optimization is also important as it helps the model to capture fine-grained patterns, such as transitions between neighboring nodes. However, local information alone is insufficient for modeling global objective and constraints like total cost and capacity. To address this, we propose to unify both global and local objectives within a Hybrid-Balance GFlowNet framework. Specifically, we integrate the DB mechanism into the original TB framework of the GFACS and AGFN models to further enhance the modeling of local transitions, particularly the relationship between the current state s_t^i and the next state s_{t+1}^i.

As shown in Fig. 1, the model records relevant information at each step, including the current state's reward $R(s_t^i)$ and flow $F(s_t^i; \boldsymbol{\theta})$, the next state's reward $R(s_{t+1}^i)$ and flow $F(s_{t+1}^i; \boldsymbol{\theta})$, as well as the forward and backward transition probability $P_f(s_{t+1}^i | s_t^i; \boldsymbol{\theta})$ and $P_b(s_t^i | s_{t+1}^i)$. Once a trajectory is completed, we apply a forward-looking technique [37] to compute the DB loss ℓ_{DB} between two successive states as:

$$\ell_{\text{DB}}(s_t^i, s_{t+1}^i; \boldsymbol{\theta}) = \left(\log \frac{P_f(s_{t+1}^i | s_t^i; \boldsymbol{\theta}) \cdot F(s_t^i; \boldsymbol{\theta}) \cdot \exp(\tilde{\mathcal{E}}(s_{t+1}^i))}{P_b(s_t^i | s_{t+1}^i) \cdot F(s_{t+1}^i; \boldsymbol{\theta}) \cdot \exp(\tilde{\mathcal{E}}(s_t^i))} \right)^2. \tag{2}$$

Here, $P_f(s_{t+1}^i | s_t^i; \boldsymbol{\theta})$ denotes the forward transition probability derived from the edge probability distribution $\eta(\mathcal{G}^*, \boldsymbol{\theta})$ in AGFN or the pheromone map in GFACS. The relationship between the trajectory-level forward probability $P_F(\tau_i; \boldsymbol{\theta})$ used in TB loss and the step-wise forward probability $P_f(s_{t+1}^i | s_t^i; \boldsymbol{\theta})$ used in DB loss is given by:

$$P_F(\tau_i; \boldsymbol{\theta}) = \prod_{t=1}^{m} P_f(s_t^i | s_{t-1}^i; \boldsymbol{\theta}). \tag{3}$$

To ensure consistency with the trajectory-level backward probability $P_B(\tau_i)$ used in TB loss, we design the step-wise backward probability $P_b(s_t^i | s_{t+1}^i)$ to reflect the structure of sub-trajectories within τ_i. Specifically, we assume that each complete trajectory τ_i consists of a multi-node sub-trajectories and j single-node sub-trajectories, and parameter P_b is accordingly determined by the varied transition structures.

Definition 1 (Trajectory Composition and Ordering Count)**.** *We define $\mathcal{A}_a$ as the set of a multi-node trajectories, and $\mathcal{J}_j$ as the set of j single-node trajectories. Together, these sequences are combined to form a complete trajectory τ_i. Let $B(\mathcal{A}_a, \mathcal{J}_j)$ denote the number of distinct orderings of sub-trajectories in $\mathcal{A}_a$ and $\mathcal{J}_j$ that result in the same complete trajectory τ_i.*

We next present the following statement, which describes the recurrence relation for $B(\mathcal{A}_a, \mathcal{J}_j)$.

Statement 1 (Trajectory Orders' Count Recurrence)**.** *The number of distinct trajectories composed of a multi-node trajectories and j single-node trajectories arranged in different orders, denoted by $B(\mathcal{A}_a, \mathcal{J}_j)$, satisfies the following recurrence relation:*

$$B(\mathcal{A}_a, \mathcal{J}_j) = 2a \cdot B(\mathcal{A}_{a-1}, \mathcal{J}_j) + j \cdot B(\mathcal{A}_a, \mathcal{J}_{j-1}). \tag{4}$$

This recurrence arises from the backward destruction of CVRP trajectories, where we consider how $B(\mathcal{A}_a, \mathcal{J}_j)$ reached its predecessors. Suppose the current state corresponds to $B(\mathcal{A}_a, \mathcal{J}_j)$, where there are a remaining multi-node trajectories and j remaining single-node trajectories to be disconnected. There are two possible types of backward transitions from this state to reach its predecessor $B(\mathcal{A}_{a-1}, \mathcal{J}_j)$ or $B(\mathcal{A}_a, \mathcal{J}_{j-1})$:

(1) Multi-node trajectory: If a multi-node trajectory is selected for backward destruction from the depot, either of its two nodes can serve as the immediate predecessor to the depot. Therefore, each of the a multi-node trajectories contributes two valid backward transitions, resulting in a total contribution of $2a \cdot B(\mathcal{A}_{a-1}, \mathcal{J}_j)$, where the recursion proceeds with $a-1$ remaining multi-node trajectories and j unchanged single-node trajectories.

(2) Single-node trajectory: If a single-node trajectory is chosen, it contains only one node, which uniquely determines the depot's predecessor. Thus, each of the j single-node trajectories contributes one backward transition, resulting in $j \cdot B(\mathcal{A}_a, \mathcal{J}_{j-1})$, where the recursion continues with a multi-node trajectories and $j-1$ single-node trajectories.

We combine both types of transitions to derive the recurrence relation as presented in Eq. 4. Subsequently, we deduce the closed-form expression of $B(\mathcal{A}_a, \mathcal{J}_j)$ from Eq. 4. The proof is provided in Appendix Sec. A.2, and the corresponding formulation is presented below:

$$B(\mathcal{A}_a, \mathcal{J}_j) = (a+j)! \cdot 2^a, \quad \text{for } a, j \geq 0. \tag{5}$$

Physically, the term $(a+j)!$ accounts for all possible orderings of the $a+j$ sub-trajectories, where each of them is treated as an atomic step in the destruction process. Each multi-node trajectory has 2 possible directions for destruction, contributing an additional 2^a multiplicative factor. In contrast, single-node trajectories allow only one valid direction. Therefore, the total number of reward-equivalent permutations is the product of these two factors.

Statement 2 (TB Backward Probability)**.** *We denote $P_B(\tau_i)$ as the backward policy probability of a complete trajectory τ_i in the GFlowNet framework under TB, formulated as:*

$$P_B(\tau_i) = \frac{1}{(a+j)! \cdot 2^a}, \tag{6}$$

where the denominator reflects the total number of distinguishable trajectory permutations given a multi-node and j single-node trajectories to achieve complete trajectory τ_i.

Statement 3 (DB Backward Probability)**.** *We denote $P_b(s_t^i \mid s_{t+1}^i)$ as the probability of a single backward transition from state s_{t+1}^i to its predecessor s_t^i, and under the DB formulation, the backward probability is defined conditionally:*

$$P_b(s_t^i | s_{t+1}^i) = \begin{cases} \frac{1}{2a+j} & \textit{if the current node is the depot}, \\ 1 & \textit{otherwise}. \end{cases} \tag{7}$$

This probability formulation originates from Eq. 4, which defines the total number of sub-trajectory backward destruction orderings. Physically, each multi-node trajectory offers two possible predecessor nodes for backward disconnection from the depot, thereby contributing the $2a$ term, while each single-node trajectory provides one such option, contributing the j term. The resulting probability $\frac{1}{2a+j}$ reflects a uniform selection over all valid backward transitions at the current decision step. In contrast, for all other nodes in the trajectory, only a single predecessor is feasible, and thus the backward transition becomes fully deterministic with probability 1.

Meanwhile, $F(s_t^i; \boldsymbol{\theta})$ in Eq. 2 represents the flow of current state, and is derived from the node embedding q at state s_t^i, which is calculated as follow:

$$F(s_t^i; \boldsymbol{\theta}) = \frac{1}{t} \sum_{x_k \in s_t^i} (W_2 \cdot \text{ReLU}(W_1 \cdot q_k + b_1) + b_2), \tag{8}$$

where W_1, W_2, b_1 and b_2 are learnable parameters and ReLU [6] is the activation function. To handle the local objective associated with state transitions, we define the reward of the predecessor state s_t^i as zero. Consequently, the energy term $\tilde{\mathcal{E}}(s_t^i)$ in Eq. 2 is also set to zero. The successor state s_{t+1}^i, in contrast, receives a non-zero transition reward. Accordingly, the energy term $\tilde{\mathcal{E}}(s_{t+1}^i)$ represents the local reward signal, and its negative is defined as follow:

$$\tilde{\mathcal{E}}(s_t^i) = R(s_t^i) - \frac{1}{h}\sum_{k=1}^{h} R(s_t^k). \tag{9}$$

As training progresses, the quality of each trajectory steadily improves, resulting in smaller values of $R(s_t^i)$ as the generated routes become shorter. In Eq. 9, we compute the energy $\tilde{\mathcal{E}}(s_t^i)$ for state s_t^i by subtracting the average reward of other trajectories at the same decision step. This formulation effectively captures the relative advantage of a given state compared to its peers, encouraging the model to assign higher energy to better-performing states. As the variance across rewards decreases during training, the energy values naturally increase.

Then, the DB loss of the completed trajectory $\tau_i = \{x_0^i, x_1^i, x_2^i, \ldots, x_m^i\}$ can be calculated as:

$$\ell_{\text{DB}}(\tau_i; \boldsymbol{\theta}) = \sum_{t=0}^{m-1} \ell_{\text{DB}}(s_t^i, s_{t+1}^i; \boldsymbol{\theta}), \tag{10}$$

where $\ell_{\text{DB}}(s_t^i, s_{t+1}^i; \boldsymbol{\theta})$ is derived from Eq. 2. The overall loss for the Hybrid-Balance GFlowNet, denoted by $\ell_{\text{HB}}(\mathcal{T}; \boldsymbol{\theta})$, is computed by aggregating both the TB loss $\ell_{\text{TB}}(\tau; \boldsymbol{\theta})$ and the DB loss $\ell_{\text{DB}}(\tau; \boldsymbol{\theta})$ over all trajectories:

$$\ell_{\text{HB}}(\mathcal{T}; \boldsymbol{\theta}) = \sum_{i=1}^{h} \ell_{\text{HB}}(\tau_i; \boldsymbol{\theta}) = \sum_{i=1}^{h} (\ell_{\text{TB}}(\tau_i; \boldsymbol{\theta}) + \ell_{\text{DB}}(\tau_i; \boldsymbol{\theta})). \tag{11}$$

This unified objective enables the model to simultaneously capture global trajectory-level structure and fine-grained local transitions, leading to more effective and robust optimization in VRPs.

3.2.3 Depot-Guided Inference

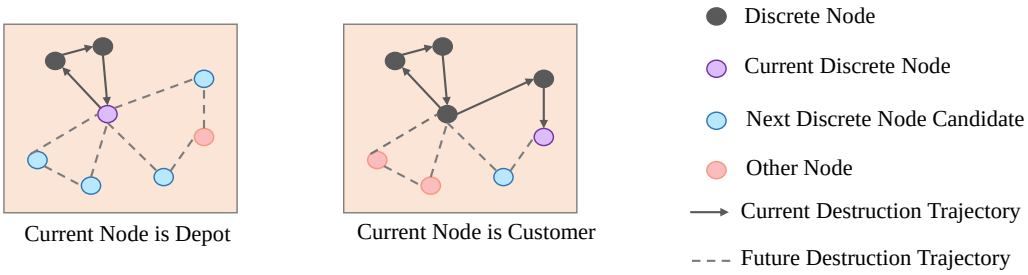

Figure 2: Illustration for Depot-Guide Inference.

The design of Hybrid-Balance GFlowNet's backward policy reveals a key insight: as illustrated in Fig. 2, only the depot node retains flexibility in choosing among multiple predecessor candidates during trajectory destruction. This flexibility stems from the construction of sub-trajectories, each of which begins and ends at the depot. In contrast, for all customer nodes, the backward transition path is uniquely defined by the trajectory structure, i.e., once a customer node is reached, its predecessor is deterministically identified. This determinism also holds during forward trajectory construction. To leverage this structural characteristic, we propose a depot-guided inference mechanism defined as:

$$x_{t+1} = \begin{cases} x & \text{if the current node } x_t \text{ is depot,} \\ x^* & \text{if the current node } x_t \text{ is customer,} \end{cases} \tag{12}$$

where $x \sim P_f(s_{t+1} \mid s_t; \boldsymbol{\theta})$ denotes sampling from the forward transition probability, and $x^* = \arg\max P_f(s_{t+1} \mid s_t; \boldsymbol{\theta})$ corresponds to greedy selection. Here, $P_f(s_{t+1} \mid s_t; \boldsymbol{\theta})$ is derived from

Table 1: Comparison on CVRP datasets of different sizes: Objective (Obj.) values and inference times (in seconds) are shown, and Gap(%) is computed with respect to LKH.

Method	200			500			1000		
	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓
LKH	28.04	59.81	-	63.32	233.72	-	120.53	433.90	-
ACO	71.46	3.36	154.85	189.79	11.14	199.73	371.30	24.50	208.06
POMO (*8)	29.22	0.29	4.21	79.86	0.84	26.12	192.18	3.06	59.45
POMO	29.45	0.23	5.03	82.92	0.59	30.95	231.88	1.48	92.38
GANCO	29.01	0.46	3.57	71.30	148.91	12.60	145.84	4.02	21.00
NeuOpt	38.42	17.19	37.02	186.17	38.05	194.01	-	-	-
AGFN	31.26	0.14	11.48	71.05	0.40	12.21	133.97	0.65	11.15
HBG-AGFN	**30.83**	0.15	**9.95**	**69.93**	0.42	**10.44**	**131.78**	0.65	**9.34**
GFACS	34.52	4.65	23.11	78.41	12.76	23.83	149.24	26.32	23.82
HBG-GFACS	**32.66**	4.67	**16.48**	**71.89**	12.77	**13.53**	**133.32**	26.33	**10.61**
GFACS (local search)	28.63	12.18	2.10	65.24	34.19	3.03	124.15	80.52	3.00
HBG-GFACS (local search)	**28.59**	12.20	**1.96**	**65.10**	34.21	**2.81**	**123.85**	80.53	**2.75**

the edge probability distribution $\eta(\mathcal{G}^*, \boldsymbol{\theta})$ in AGFN or the pheromone map in GFACS. Under this strategy, exploration through sampling is applied only at the depot, while customer nodes follow a deterministic, greedy policy.

It is important to note that depot-guided inference is specifically designed for problems featuring a designated depot node, such as the CVRP. For problems lacking a depot or node-role differentiation, such as the TSP, we retain their original inference procedures, including hybrid decoding strategy [55] for AGFN and the ant clony search [22] for GFACS.

4 Experiment

We conduct experiments to validate the effectiveness of the Hybrid-Balance GFlowNet (HBG) in enhancing two representative GFlowNet-based solvers, i.e., AGFN and GFACS, on CVRP. We first present comparison results, followed by ablation studies to analyze the contribution of individual components. Lastly, we extend the evaluation to other vehicle routing problem.

Dataset: We adopt synthetic CVRP datasets following standard settings used in prior work [22, 25, 55, 49]. Each instance features a single depot and multiple customers served by a vehicle with fixed capacity C. The depot and customer coordinates are sampled uniformly from the unit square $[0, 1]^2$, and customer demands follow a uniform distribution $U[a, b]$ with $a = 1$ and $b = 9$. The vehicle capacity is fixed at $C = 50$ across all problem sizes: 100, 200, 500, and 1,000 nodes. For testing, we generate 128 synthetic instances for each of the 200-, 500-, and 1,000-node settings, aligned with evaluation rules established in AGFN and GFACS. The code is available at `https://github.com/ZHANG-NI/HBG`

Hyperparameters: We adopt the same model configurations and training settings as AGFN and GFACS, including network architecture, batch size, learning rate, optimizer, and other hyperparameters. Training is conducted using sampling-based decoding with $\mathcal{N} = 20$ routes per instance. During inference, AGFN uses depot-guided inference, and GFACS applies an ant colony search with depot-guided node selection. All models are trained on 100-node instances. The experiments are conducted on a server equipped with an NVIDIA A100 GPU and an Intel Xeon 6342 CPU.

4.1 Performance on Synthetic CVRP Instances

We compare HBG-enhanced models, i.e., HBG-AGFN and HBG-GFACS, with their original TB-based counterparts, AGFN [55] and GFACS [22]. AGFN constructs routes in an end-to-end manner, while GFACS searches for solutions by combining GFlowNet with ant colony optimization. We also include classical heuristics (LKH [19], ACO [3]) and learning-based baselines (POMO [25], GANCO [50], NeuOpt [31]) for comparison. All methods are trained on 100-node instances and evaluated on CVRP200, CVRP500, and CVRP1000 datasets, following AGFN and GFACS evaluation protocols. Additional experiments on the public benchmark CVRPLib are reported in Appendix B.1.

Table 1 shows that HBG consistently improves performance across all problem sizes for AGFN, GFACS, and GFACS with local search. The performance gains are significant, with gap reductions of up to 16.23%, 55.46%, and 8.33%, respectively. Improvements become more pronounced as instance size increases, indicating strong scalability. Inference incurs only minor overhead (0.01–0.04

Table 2: Ablation Study on AGFN and GFACS: Gap(%) is computed with respect to LKH.

(a) AGFN

Method	200		500		1000	
	Obj. ↓	Gap(%) ↓	Obj. ↓	Gap(%) ↓	Obj. ↓	Gap(%) ↓
LKH	28.04	-	63.32	-	120.53	-
AGFN	31.26	11.48	71.05	12.21	133.97	11.15
+ HB	31.08	10.84	69.99	10.53	131.94	9.47
+ Depot-Guide Inference	30.83	9.95	69.93	10.44	131.78	9.34

(b) GFACS

Method	200		500		1000	
	Obj. ↓	Gap(%) ↓	Obj. ↓	Gap(%) ↓	Obj. ↓	Gap(%) ↓
LKH	28.04	-	63.32	-	120.53	-
GFACS	34.52	23.11	78.41	23.83	149.24	23.82
+ HB	34.01	21.29	76.67	21.08	144.48	19.87
+ Depot-Guide Inference	32.66	16.48	71.89	13.53	133.32	10.61

Table 3: Comparison of DB, TB, and HB: Gap(%) is computed with respect to LKH.

(a) AGFN

Method	200		500		1000	
	Obj.	Gap(%)	Obj.	Gap(%)	Obj.	Gap(%)
LKH	28.04	–	63.32	–	120.53	–
DB	34.41	22.72	76.78	21.26	143.25	18.85
TB	31.26	11.48	71.05	12.21	133.97	11.15
HB	**31.08**	**10.84**	**69.99**	**10.53**	**131.94**	**9.47**

(b) GFACS

Method	200		500		1000	
	Obj. ↓	Gap(%) ↓	Obj. ↓	Gap(%) ↓	Obj. ↓	Gap(%) ↓
LKH	28.04	–	63.32	–	120.53	–
DB	43.28	54.36	94.20	48.77	181.87	50.89
TB	34.52	23.11	78.41	23.83	149.24	23.82
HB	**34.01**	**21.29**	**76.67**	**21.08**	**144.48**	**19.87**

seconds) due to temporary loading of flow parameters, which does not impact overall runtime or scalability. Compared to other heuristic and learning-based methods, HBG-AGFN and HBG-GFACS achieve competitive or superior solution quality across all scales. On CVRP200, both methods outperform ACO and NeuOpt. On CVRP500 and CVRP1000, they continue to generalize effectively, outperforming ACO, POMO, GANCO, and NeuOpt. These results highlight the robustness, efficiency, and strong generalization capabilities of the proposed HBG framework.

4.2 Ablation Study

Comparison of Component Contributions. We evaluate the contribution of each component in the HBG framework for both AGFN and GFACS. First, we incorporate the Hybrid-Balance (HB) module into the original models. Then, we add the depot-guided inference mechanism on top of the HB-enhanced variants. As shown in Table 2, each component contributes significantly to performance. Incorporating the HB module alone reduces the optimality gap by up to 15.07% in AGFN and 16.58% in GFACS. Adding depot-guided inference provides further gains, especially for larger instances. These results confirm that the HB module offers consistent improvements and depot-guided inference delivers additional benefits in depot-centric tasks.

Comparison of Balance Strategies. To further validate the effectiveness of Hybrid Balance (HB), we conduct a comparison against Trajectory Balance (TB) and Detailed Balance (DB) under identical training settings on 100-node instances, evaluated on CVRP200, CVRP500, and CVRP1000. As shown in Table 3, the HB module consistently outperforms both TB and DB across all instance sizes for both AGFN and GFACS. Notably, HB achieves up to a 15.07% improvement over TB in AGFN and up to 16.58% in GFACS. These results highlight the superior effectiveness of Hybrid Balance as a unifying optimization strategy.

Depot-Guided Inference Variants. We assess four variants of the depot-guided inference strategy by applying either sampling or greedy decoding at the depot and customer nodes. Tests are conducted using both AGFN and GFACS on CVRP200, CVRP500, and CVRP1000. As shown in Table 4, the combination of sampling at the depot and greedy decoding at customers yields the best performance. This setting consistently outperforms all other variants, including depot greedy + customer sampling, depot greedy + customer greedy, and depot sampling + customer sampling. These results validate the effectness of our depot-guided inference mechanism.

4.3 Generalization to Other Vehicle Routing Problem

We further evaluate our framework on the Traveling Salesman Problem (TSP), a key VRP variant. Baselines include GFlowNet-based solvers (AGFN [55], GFACS [22]), classical heuristics (LKH [19], ACO [3]), and learning-based models (POMO [25], GANCO [50], NeuOpt [31]). All models are trained on 100-node instances and evaluated on 200-, 500-, and 1,000-node settings. Since TSP lacks a depot node, depot-guided inference is not used. Table 5 shows that HBG-AGFN consistently outperforms AGFN, reducing the gap by up to 17.64%. HBG-GFACS also achieves

Table 4: Ablation Study on Depot-Guided Inference. Gap(%) is computed with respect to the LKH. DG represents depot greedy, DS represents depot sampling, CG represents customer greedy, CS represents customer sampling.

(a) AGFN

Method	200			500			1000		
	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓
LKH	28.04	59.81	-	63.32	233.72	-	120.53	433.90	-
DG and CS	32.78	0.16	16.90	76.42	0.41	20.69	146.14	0.65	21.25
DG and CG	31.96	0.16	13.98	71.35	0.41	12.68	133.32	0.65	10.61
DS and CS	31.88	0.16	13.69	74.49	0.41	17.64	144.74	0.65	20.09
DS and CG	**30.83**	0.16	**9.95**	**69.93**	0.41	**10.44**	**131.78**	0.65	**9.34**

(b) GFACS

Method	200			500			1000		
	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓
LKH	28.04	59.81	-	63.32	233.72	-	120.53	433.90	-
DG and CS	34.87	4.67	24.36	75.99	12.78	20.01	148.53	26.33	23.23
DG and CG	34.58	4.67	23.32	74.12	12.78	17.06	135.33	26.33	12.28
DS and CS	33.38	4.67	19.04	75.79	12.78	19.69	143.46	26.33	19.02
DS and CG	**32.66**	4.67	**16.48**	**71.89**	12.78	**13.53**	**133.32**	26.33	**10.61**

Table 5: Comparison of performance and runtime on TSP with 200, 500, and 1000 nodes. Gap(%) is computed relative to LKH (10000).

Method	200			500			1000		
	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓	Obj. ↓	Time(s) ↓	Gap(%) ↓
LKH	10.62	38.80	–	16.30	75.29	–	22.68	149.36	–
ACO	45.72	1.79	330.51	149.62	5.87	817.91	315.42	13.20	1290.74
POMO	10.97	0.12	3.30	20.85	0.39	27.91	33.94	0.59	49.65
POMO (*8)	10.90	0.20	2.64	20.44	0.55	25.40	32.60	3.42	43.74
NeuOpt	13.22	6.39	24.48	138.15	14.54	747.55	325.28	27.84	1334.22
GANCO	11.30	0.11	6.40	19.69	0.36	20.80	29.97	0.85	32.14
AGFN	11.85	0.08	11.58	19.08	0.26	17.06	27.15	0.70	19.71
Our-AGFN	**11.73**	0.11	**10.45**	**18.59**	0.27	**14.05**	**26.87**	0.71	**18.47**
GFACS	13.04	1.64	22.79	24.41	9.42	49.76	41.86	20.79	84.57
Our-GFACS	**12.68**	1.66	**19.40**	**24.19**	9.43	**48.41**	**39.90**	20.81	**75.93**
GFACS (local search)	10.78	6.67	1.51	17.10	27.76	4.91	24.45	58.42	7.80
Our-GFACS (local search)	**10.78**	6.68	**1.50**	**17.05**	27.78	**4.60**	**24.42**	58.42	**7.67**

notable improvements, with the gap on 200-node instances reduced from 22.79% to 19.40%. With local search, HBG-GFACS achieves further improvements, with the best gap reduction reaching 6.31%. Compared to classical heuristics and learning-based methods such as ACO, POMO, NeuOpt, and GANCO, both HBG-AGFN and HBG-GFACS achieve competitive results on TSP tasks. These results confirm the generalizability and strong performance of HBG on TSP tasks.

5 Conclusion

In this paper, we introduced the Hybrid-Balance GFlowNet (HBG) framework to enhance the performance of GFlowNet-based solvers for vehicle routing problems. HBG unifies Trajectory Balance and Detailed Balance in a principled and adaptive manner to jointly optimize local and global objectives. We also proposed a depot-guided inference strategy aligned with the Hybrid-Balance principle, specifically tailored for depot-centric problems. Extensive experiments on both CVRP and TSP benchmarks demonstrate that HBG significantly improves the performance of two representative GFlowNet-based solvers, i.e., AGFN and GFACS, showcasing improved solution quality, scalability, and generalization. A current limitation of HBG is its reliance on existing GFlowNet-based models, as its performance depends in part on the underlying solver, which might be inferior to others. In future work, we plan to integrate it with alternative stronger generative policies and solvers.

Acknowledgments and Disclosure of Funding

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-RP-2022-031, AISG3-RP-2025-036-USNSF), and the Lee Kong Chian Fellowship awarded to CAO Zhiguang by Singapore Management University.

References

[1] Florian Arnold, Michel Gendreau, and Kenneth Sörensen. Efficiently solving very large-scale routing problems. *Computers & operations research*, 107:32–42, 2019.

[2] Gulay Barbarosoglu and Demet Ozgur. A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research*, 26(3):255–270, 1999.

[3] John E Bell and Patrick R McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced engineering informatics*, 18(1):41–48, 2004.

[4] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

[5] 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. *Advances in Neural Information Processing Systems*, 35:31226–31238, 2022.

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic relu. In *European conference on computer vision*, pages 351–367. Springer, 2020.

[7] Younghoon Choi, Bradford Robertson, Youngjun Choi, and Dimitri Mavris. A multi-trip vehicle routing problem for small unmanned aircraft systems-based urban delivery. *Journal of Aircraft*, 56(6):2309–2323, 2019.

[8] Zeynel Abidin Çil, Hande Öztop, Zülal Diri Kenger, and Damla Kizilay. Integrating distributed disassembly line balancing and vehicle routing problem in supply chain: Integer programming, constraint programming, and heuristic algorithms. *International Journal of Production Economics*, 265:109014, 2023.

[9] Tiago da Silva, Eliezer Silva, António Góis, Dominik Heider, Samuel Kaski, Diego Mesquita, and Adèle Ribeiro. Human-in-the-loop causal discovery under latent confounding using ancestral gflownets. *arXiv preprint arXiv:2309.12032*, 2023.

[10] Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pages 518–528. PMLR, 2022.

[11] Rodolfo Dondo, Carlos A Méndez, and Jaime Cerdá. The multi-echelon vehicle routing problem with cross docking in supply chain management. *Computers & Chemical Engineering*, 35(12):3002–3024, 2011.

[12] Verena Ch Ehrler, Dustin Schöder, and Saskia Seidel. Challenges and perspectives for the use of electric vehicles for last mile logistics of grocery e-commerce–findings from case studies in germany. *Research in Transportation Economics*, 87:100757, 2021.

[13] Han Fang, Zhihao Song, Paul Weng, and Yutong Ban. Invit: a generalizable routing problem solver with invariant nested view transformer. In *Proceedings of the 41st International Conference on Machine Learning*, pages 12973–12992, 2024.

[14] 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 *International Joint Conference on Artificial Intelligence*, 2024.

[15] Antonio Giallanza and Gabriella Li Puma. Fuzzy green vehicle routing problem for designing a three echelons supply chain. *Journal of Cleaner Production*, 259:120774, 2020.

[16] Akshat Santhana Gopalan and Sowmya Ramaswamy Krishnan. Generative flow networks for lead optimization in drug design (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29484–29486, 2025.

[17] Ronald L Graham. *Concrete mathematics: a foundation for computer science*. Pearson Education India, 1994.

[18] Jiaqi Guo, Jiancheng Long, Xiaoming Xu, Miao Yu, and Kai Yuan. The vehicle routing problem of intercity ride-sharing between two cities. *Transportation Research Part B: Methodological*, 158:113–139, 2022.

[19] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000.

[20] Ziwei Huang, Jianan Zhou, Zhiguang Cao, and Yixin Xu. Rethinking light decoder-based solvers for vehicle routing problems. *The Thirteenth International Conference on Learning Representations*, 2025.

[21] 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(4):1654–1666, 2015.

[22] Minsu Kim, Sanghyeok Choi, Hyeonah Kim, Jiwoo Son, Jinkyoo Park, and Yoshua Bengio. Ant colony sampling with gflownets for combinatorial optimization. In *The 28th International Conference on Artificial Intelligence and Statistics*.

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

[24] Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Chenghao Liu, Mike Tyers, and Robert Batey. Rgfn: Synthesizable molecular generation using gflownets. *Advances in Neural Information Processing Systems*, 37:46908–46955, 2024.

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

[26] Yeong-Dae Kwon, Jinho Choo, Iljoo Yoon, Minah Park, Duwon Park, and Youngjune Gwon. Matrix encoding networks for neural combinatorial optimization. *Advances in Neural Information Processing Systems*, 34:5138–5149, 2021.

[27] Sida Li, Ioana Marinescu, and Sebastian Musslick. Gfn-sr: Symbolic regression with generative flow networks. In *NeurIPS 2023 AI for Science Workshop*.

[28] Wenqian Li, Yinchuan Li, Shengyu Zhu, Yunfeng Shao, Jianye Hao, and Yan Pang. Gflowcausal: Generative flow networks for causal discovery. *arXiv preprint arXiv:2210.08185*, 2022.

[29] Yinchuan Li, Shuang Luo, Haozhi Wang, and HAO Jianye. Cflownets: Continuous control with generative flow networks. In *The Eleventh International Conference on Learning Representations*.

[30] Yeqian Lin, Wenquan Li, Feng Qiu, and He Xu. Research on optimization of vehicle routing problem for ride-sharing taxi. *Procedia-Social and Behavioral Sciences*, 43:494–502, 2012.

[31] Yining Ma, Zhiguang Cao, and Yeow Meng Chee. Learning to search feasible and infeasible regions of routing problems with flexible neural k-opt. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967, 2022.

[33] Kishan Reddy Nagiredla, Arun Kumar AV, Thommen George Karimpanal, and Santu Rana. Robonet: A sample-efficient robot co-design generator. In *[CoRL 2024] Morphology-Aware Policy and Design Learning Workshop (MAPoDeL)*.

[34] Andrei Cristian Nica, Moksh Jain, Emmanuel Bengio, Cheng-Hao Liu, Maksym Korablyov, Michael M Bronstein, and Yoshua Bengio. Evaluating generalization in gflownets for molecule design. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

[35] Mizu Nishikawa-Toomey, Tristan Deleu, Jithendaraa Subramanian, Yoshua Bengio, and Laurent Charlin. Bayesian learning of causal structure and mechanisms with gflownets and variational bayes. *arXiv preprint arXiv:2211.02763*, 2022.

[36] Ibrahim Hassan Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, 41:421–451, 1993.

[37] Ling Pan, Nikolay Malkin, Dinghuai Zhang, and Yoshua Bengio. Better training of gflownets with local credit and incomplete trajectories. In *International Conference on Machine Learning*, pages 26878–26890. PMLR, 2023.

[38] Xuanhao Pan, Yan Jin, Yuandong Ding, Mingxiao Feng, Li Zhao, Lei Song, and Jiang Bian. H-tsp: Hierarchically solving the large-scale traveling salesman problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9345–9353, 2023.

[39] MID Ranathunga, AN Wijayanayake, and DHH Niwunhella. Solution approaches for combining first-mile pickup and last-mile delivery in an e-commerce logistic network: A systematic literature review. In *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, volume 4, pages 267–275. IEEE, 2021.

[40] Seonghwan Seo, Minsu Kim, Tony Shen, Martin Ester, Jinkyoo Park, Sungsoo Ahn, and Woo Youn Kim. Generative flows on synthetic pathway for drug design. In *NeurIPS 2024 Workshop on AI for New Drug Modalities*.

[41] Tony Shen, Mohit Pandey, Jason R Smith, Artem Cherkasov, and Martin Ester. Tacogfn: Target conditioned gflownet for structure-based drug design. *CoRR*, 2023.

[42] Tiago Silva, Daniel Augusto de Souza, and Diego Mesquita. Streaming bayes gflownets. *Advances in Neural Information Processing Systems*, 37:27153–27177, 2024.

[43] Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in neural information processing systems*, 36:3706–3731, 2023.

[44] Amirmahdi Tafreshian, Neda Masoud, and Yafeng Yin. Frontiers in service science: Ride matching for peer-to-peer ride sharing: A review and future directions. *Service Science*, 12(2-3):44–60, 2020.

[45] Christo Kurisummoottil Thomas and Walid Saad. Neuro-symbolic causal reasoning meets signaling game for emergent semantic communications. *IEEE Transactions on Wireless Communications*, 23(5):4546–4563, 2023.

[46] Thibaut Vidal. Hybrid genetic search for the cvrp: Open-source implementation and swap* neighborhood. *Computers & Operations Research*, 140:105643, 2022.

[47] Mingzhao Wang, You Zhou, Zhiguang Cao, Yubin Xiao, Xuan Wu, Wei Pang, Yuan Jiang, Hui Yang, Peng Zhao, and Yuanshu Li. An efficient diffusion-based non-autoregressive solver for traveling salesman problem. In *Proceedings of the 31th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025.

[48] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[49] Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Neurolkh: Combining deep learning model with lin-kernighan-helsgaun heuristic for solving the traveling salesman problem. *Advances in Neural Information Processing Systems*, 34:7472–7483, 2021.

[50] Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Generative adversarial training for neural combinatorial optimization models. 2022.

[51] Haoran Ye, Jiarui Wang, Zhiguang Cao, Helan Liang, and Yong Li. Deepaco: Neural-enhanced ant systems for combinatorial optimization. *Advances in neural information processing systems*, 36:43706–43728, 2023.

[52] 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.

[53] Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron C Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial problems with gflownets. *Advances in neural information processing systems*, 36:11952–11969, 2023.

[54] Mengdi Zhang, Saurabh Pratap, Zhiheng Zhao, Dhirendra Prajapati, and George Q Huang. Forward and reverse logistics vehicle routing problems with time horizons in b2c e-commerce logistics. *International Journal of Production Research*, 59(20):6291–6310, 2021.

[55] Ni Zhang, Jingfeng Yang, Zhiguang Cao, and Xu Chi. Adversarial generative flow network for solving vehicle routing problems. In *The Thirteenth International Conference on Learning Representations*, 2025.

[56] Changjiang Zheng, Yuhang Gu, Jinxing Shen, and Muqing Du. Urban logistics delivery route planning based on a single metro line. *IEEE Access*, 9:50819–50830, 2021.

[57] Yiheng Zhu, Jialu Wu, Chaowen Hu, Jiahuan Yan, Tingjun Hou, Jian Wu, et al. Sample-efficient multi-objective molecular optimization with gflownets. *Advances in Neural Information Processing Systems*, 36:79667–79684, 2023.

A Methodology Details

A.1 Graph Neural Network

Our GNN architecture follows the same design as those used in AGFN and GFACS to ensure a fair comparison, which uses a custom GNN architecture designed specifically for the VRP task, and employs a custom message-passing GNN that jointly updates node and edge representations over multiple layers. At each layer l, node embedding h_i^l of the i-th node and edge embedding e_{ij}^l between the i-th and j-th nodes are updated via the following formulations:

$$h_i^{l+1} = h_i^l + \text{ACT}\Big(\text{BN}\big(W_1^l h_i^l + A(\sigma(e_{ij}^l) \odot W_2^l h_j^l)\big)\Big), \tag{13}$$

$$e_{ij}^{l+1} = e_{ij}^l + \text{ACT}\Big(\text{BN}\big(W_3^l e_{ij}^l + W_4^l h_i^l + W_5^l h_j^l\big)\Big), \tag{14}$$

where $W_1^l, W_2^l, W_3^l, W_4^l, W_5^l$ are learnable parameters, $A(\cdot)$ denotes the aggregation function (mean pooling in our case), $\sigma(\cdot)$ is the sigmoid activation that modulates attention over neighbors, and ACT denotes the SiLU activation. Batch normalization (BN) is applied at each step for stability. The number of layers is set to 12 for HBG-GFACS and 16 for HBG-AGFN, with hidden dimensions of 32 and 64, respectively.

A.2 Proof Process of Hybrid-Balance

To solve Eq. 4, We begin by establishing the boundary conditions. When either $a = 0$ or $j = 0$, the recurrence simplifies accordingly. For instance, when $a = 0$, we obtain:

$$B(\mathcal{A}_0, \mathcal{J}_j) = j \cdot B(\mathcal{A}_0, \mathcal{J}_{j-1}). \tag{15}$$

When $a = 0$ and $j = 0$, there are no sub-trajectories in τ_i, and τ_i contains only the depot node. Therefore, the base case becomes $B(\mathcal{A}_0, \mathcal{J}_0) = 1$, and it is equivalent to:

$$B(\mathcal{A}_0, \mathcal{J}_0) = 1 = \frac{(0+0)!}{0! \cdot 0!}, \quad \text{for } a = j = 0. \tag{16}$$

Using Eq. 15 recursively and applying the base case $B(\mathcal{A}_0, \mathcal{J}_0) = 1$, we derive:

$$B(\mathcal{A}_0, \mathcal{J}_j) = j!. \tag{17}$$

Similarly, when $j = 0$, we can deduce that:

$$B(\mathcal{A}_a, \mathcal{J}_0) = 2^a \cdot a!. \tag{18}$$

We now normalize Eq. 4 by dividing both sides by $2^a \cdot a! \cdot j!$, resulting in:

$$\frac{B(\mathcal{A}_a, \mathcal{J}_j)}{2^a \cdot a! \cdot j!} = \frac{B(\mathcal{A}_{a-1}, \mathcal{J}_j)}{2^{a-1} \cdot (a-1)! \cdot j!} + \frac{B(\mathcal{A}_a, \mathcal{J}_{j-1})}{2^a \cdot a! \cdot (j-1)!}. \tag{19}$$

Based on this, we define a normalized function:

$$c(\mathcal{A}_a, \mathcal{J}_j) \triangleq \frac{B(\mathcal{A}_a, \mathcal{J}_j)}{2^a \cdot a! \cdot j!}. \tag{20}$$

Substituting Eq. 20 into Eq. 19, we obtain the recurrence:

$$c(\mathcal{A}_a, \mathcal{J}_j) = c(\mathcal{A}_{a-1}, \mathcal{J}_j) + c(\mathcal{A}_a, \mathcal{J}_{j-1}). \tag{21}$$

We analyze the recurrence in Eq. 21 under the boundary conditions, and derive results from Eqs. 17, 18, and 20:

$$c(\mathcal{A}_0, \mathcal{J}_j) = 1, \quad c(\mathcal{A}_a, \mathcal{J}_0) = 1, \quad \text{for all } a, j \geq 0.$$

These boundary conditions are equivalent to:

$$c(\mathcal{A}_0, \mathcal{J}_j) = 1 = \frac{(0+j)!}{0! \cdot j!}, \quad \text{for } a = 0, j > 0, \tag{22}$$

$$c(\mathcal{A}_a, \mathcal{J}_0) = 1 = \frac{(a+0)!}{a! \cdot 0!}, \quad \text{for } a > 0, j = 0. \tag{23}$$

Table 6: Test on the CVRPLIB-XXL benchmark. Gap(%) is computed with respect to the optimal solution.

Gap(%) ↓	**L1 (3k)**	**L2 (4k)**	**A1 (6k)**	**A2 (7k)**	**G1 (10k)**	**G2 (11k)**	**B1 (15k)**	**B2 (16k)**
AGFN	1145.18	27.87	29.62	24.59	142.62	27.24	**20.65**	120.69
HBG-AGFN	**97.15**	**25.09**	**25.36**	**20.96**	**113.89**	**20.47**	109.80	**46.97**
GFACS	119.24	1788.57	112.08	236.79	281.34	2207.71	352.69	2537.29
HBG-GFACS	**52.25**	**33.96**	**21.58**	**39.19**	**78.88**	**31.60**	**179.70**	**37.43**

Moreover, motivated by Pascal's rule [17], which states:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \quad \text{for all integers } n \geq 1 \text{ and } 1 \leq k \leq n,$$

and by setting $n = a + j$, $k = a$, we obtain:

$$\binom{a+j}{a} = \binom{a+j-1}{a-1} + \binom{a+j-1}{a}, \quad \text{for } a, j \geq 1,$$

which is equivalent to:

$$\binom{a+j}{a} = \binom{(a-1)+j}{a-1} + \binom{a+(j-1)}{a}.$$

This is similar with the function of $c(\mathcal{A}_a, \mathcal{J}_j)$ in Eq. 21. Therefore, the closed-form expression of $c(\mathcal{A}_a, \mathcal{J}_j)$ is:

$$c(\mathcal{A}_a, \mathcal{J}_j) = \binom{a+j}{a} = \frac{(a+j)!}{a! \cdot j!}, \quad \text{for } a, j \geq 1. \tag{24}$$

By extending Eqs. 16, 22, 23, and 24, the generalized closed-form solution for all $a, j \geq 0$ is given as:

$$c(\mathcal{A}_a, \mathcal{J}_j) = \binom{a+j}{a} = \frac{(a+j)!}{a! \cdot j!}. \tag{25}$$

Finally, substituting Eq. 25 into Eq. 20, we obtain the closed-form expression for the total number of distinct sub-trajectory orderings resulting in the same complete trajectory:

$$B(\mathcal{A}_a, \mathcal{J}_j) = (a+j)! \cdot 2^a, \quad \text{for } a, j \geq 0. \tag{26}$$

B More Experiments

B.1 Test on Real world dataset

To evaluate our model's performance on real-world data, we conduct experiments on the CVRPLIB-XXL [1], which is designed to test model's performance on large-scale real-world instances. As shown in Table 6, HBG-AGFN achieves up to a 91.51% reduction in gap compared to AGFN, while HBG-GFACS achieves up to 98.52% improvement over GFACS. These results demonstrate that our framework significantly enhances the performance of GFlowNet-based solvers on real-world datasets.

Table 7: Comparison of Hyperparameter Settings on CVRP

(a) Sparsity Parameter

CVRP (Gap%)	200	500	1000
HBG_GFACS_2	19.26	16.27	12.69
HBG_GFACS_5(origin)	**16.48**	13.59	10.61
HBG_GFACS_8	17.90	13.83	11.02
HBG_GFACS_10	20.43	17.88	14.9
HBG_GFACS(local)_5(origin)	1.96	**2.81**	2.75
HBG_GFACS(local)_10	2.07	3.43	4.57
HBG_AGFN_2	12.62	13.48	17.32
HBG_AGFN_5(origin)	9.50	**10.44**	9.34
HBG_AGFN_8	11.95	13.57	12.69
HBG_AGFN_10	9.10	14.56	14.04

(b) Learning Rate

CVRP (Gap%)	200	500	1000
HBG_GFACS_5×10^{-3}	17.15	14.48	11.77
HBG_GFACS_1×10^{-3}	17.43	14.57	11.50
HBG_GFACS_5×10^{-4}(origin)	**16.48**	**13.59**	**10.61**
HBG_GFACS_1×10^{-4}	18.22	14.19	11.30
HBG_GFACS(local)_5×10^{-3}	1.71	2.77	4.33
HBG_GFACS(local)_1×10^{-3}	**1.70**	**2.65**	**2.51**
HBG_GFACS(local)_5×10^{-4}(origin)	1.93	4.42	5.75
HBG_GFACS(local)_1×10^{-4}	4.53	6.49	6.50
HBG_AGFN_5×10^{-3}	11.41	14.62	11.00
HBG_AGFN_1×10^{-3}	10.14	12.77	10.62
HBG_AGFN_5×10^{-4}(origin)	**9.95**	**10.44**	**9.34**
HBG_AGFN_1×10^{-4}	9.37	10.27	9.59

(c) Optimizer Type

CVRP (Gap%)	200	500	1000
HBG_GFACS_SGD	17.80	15.19	12.45
HBG_GFACS_Adam	17.33	14.28	11.11
HBG_GFACS_AdamW(origin)	**16.48**	**13.59**	**10.61**
HBG_GFACS(local)_SGD	4.23	4.26	4.55
HBG_GFACS(local)_Adam	**2.03**	2.70	2.73
HBG_GFACS(local)_AdamW(origin)	2.30	3.43	4.57
HBG_AGFN_SGD	14.66	13.00	12.47
HBG_AGFN_Adam	10.16	11.88	11.77
HBG_AGFN_AdamW(origin)	**9.95**	**10.44**	**9.34**

B.2 Hyperparameter Sensitivity Analysis on CVRP

The introduction of a new loss component (L_{DB}) could potentially alter the optimization landscape, and that the original hyperparameter settings used in AGFN and GFACS may not be optimal for the proposed HBG variants. We further conduct additional experiments in which we re-tuned key hyperparameters for the HBG models, including the sparsity parameter, learning rate, and optimizer settings. The updated results (Table 7a–7c) will be incorporated into the Appendix of the revised version. HBG-GFACS_2 refers to the HBG-GFACS model evaluated with a sparsity parameter $k = |V|/2$; the interpretation is analogous for the other entries. The term *origin* denotes the original hyperparameter configuration used in our main experiments. HBG-GFACS_5$\times 10^{-3}$ refers to the HBG-GFACS model evaluated with a learning rate of 5×10^{-3}; the interpretation is analogous for the other entries. HBG-GFACS_SGD refers to the HBG-GFACS model evaluated using the SGD optimizer; the interpretation is analogous for the other entries.

As the sparsity parameter results shown in Table 7a, the optimal sparsity setting yields the best performance for both HBG-GFACS and HBG-AGFN. Regarding the learning rate comparisons in Table 7b, for HBG-GFACS, the original value of 5×10^{-4} performs best without local search, while a value of 1×10^{-3} achieves the best results when local search is enabled. For HBG-AGFN, a learning rate of 1×10^{-4} shows superior performance on the 200- and 500-node instances, whereas the original value performs best on the 1000-node instances. As for the optimizer comparison in Table 7c, the original AdamW setting provides the best performance in most cases. An exception is observed on the 500- and 1000-node instance with local search, where Adam slightly outperforms AdamW for HBG-GFACS.

B.3 Weight λ of DB and TB

Table 8: Weights Analysis of HBG-AGFN on CVRP

Size (Gap%)	0.5→0	1→0	2→0	0.5→0.5	1→0.5	2→0.5	0.5→1	1→1 (origin)	2→2
200	11.09	10.80	10.94	11.41	11.98	11.20	9.98	**9.95**	10.44
500	11.27	12.04	11.18	12.15	11.66	11.80	11.63	**10.44**	11.98
1000	10.89	10.61	9.89	11.03	10.84	10.98	10.87	**9.34**	10.08

Table 9: Weights Analysis of HBG-AGFN on TSP

Size (Gap%)	0.5→0	1→0	2→0	0.5→0.5	1→0.5	2→0.5	0.5→1	1→1 (origin)	2→2
200	10.26	10.42	10.49	11.48	11.26	10.66	**10.08**	10.45	10.36
500	15.15	15.40	16.13	16.87	16.09	15.79	14.91	**14.05**	15.64
1000	18.80	18.55	19.00	19.26	21.87	22.00	18.52	**18.47**	18.81

Table 10: Weights Analysis of HBG-GFACS on CVRP

Size / Gap(%) / λ	0.5→0	1→0	2→0	0→0.5	0→1	0→2	0.5→0.5	1→1 (origin)	2→2
200	**14.05**	19.29	17.37	17.72	18.47	19.08	16.42	16.48	18.87
500	**11.78**	17.66	15.38	15.28	16.35	15.71	13.57	13.53	15.49
1000	11.29	15.80	12.71	12.88	13.47	12.80	10.91	**10.61**	12.80
200 (local search)	2.39	**1.78**	2.03	2.07	2.14	1.96	1.96	1.96	1.78
500 (local search)	3.02	**2.51**	2.91	3.41	3.44	2.99	3.17	2.81	2.62
1000 (local search)	2.82	**2.27**	2.82	3.31	3.53	3.22	2.75	2.75	2.40

We have conducted additional experiments on the weighted combination (i.e., $\mathcal{L}_{\mathrm{HB}} = \mathcal{L}_{\mathrm{TB}} + \lambda\mathcal{L}_{\mathrm{DB}}$) to assess the sensitivity of our method to different values of λ, which are gathered in Tables 8–11. The results include both adaptive weights (i.e., λ: 0.5→0, 1→0, 2→0 and so on) and fixed weights (i.e., 0.5→0.5, 1→1, 2→2), where 0.5→0 denotes changing the TB:DB loss weight ratio from 1:0.5 to 1:0 during training, with TB weight fixed at 1. The interpretation for other entries follows analogously. The term *origin* denotes the original hyperparameter setting used in our main experiments. We find that, for HBG-AGFN, a fixed 1:1 weight between TB and DB consistently yields the best performance. Similarly, for HBG-GFACS, this ratio offers the most favorable trade-off between performance and stability with and without local search.

Table 11: Weights Analysis of HBG-GFACS on TSP

Size / Gap(%) / λ	0.5→0	1→0	2→0	0→0.5	0→1	0→2	0.5→0.5	1→1 (origin)	2→2
200	**19.13**	20.61	23.06	22.21	20.56	24.02	21.64	19.40	22.82
500	49.51	49.61	50.80	49.48	48.47	56.13	49.62	**48.41**	50.37
1000	**72.30**	80.19	90.50	79.37	80.39	96.36	77.45	75.93	87.23
200 (local search)	1.57	**1.49**	1.59	1.59	1.62	1.82	1.54	1.50	1.62
500 (local search)	4.81	**4.65**	5.02	4.88	4.73	5.18	4.73	4.68	5.12
1000 (local search)	7.73	7.68	7.86	7.83	8.02	8.20	7.70	**7.67**	7.99

B.4 Statistical Significance of Experiment

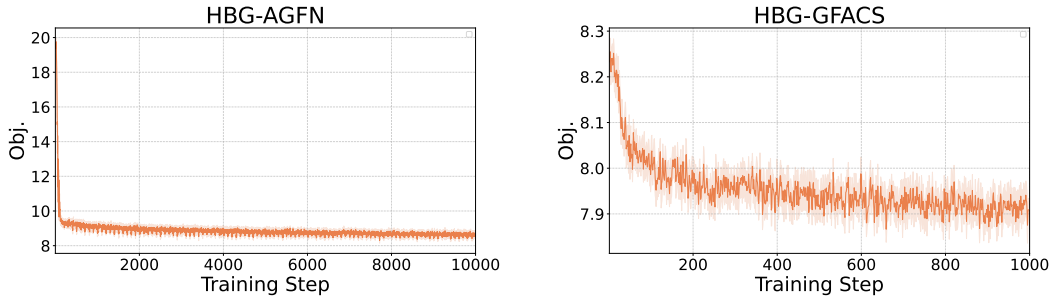

Figure 3: Training Process of HBG-AGFN and HBG-GFACS.

As shown in Fig. 3, both HBG-AGFN and HBG-GFACS exhibit a clear and steady decline in objective values throughout the training process, indicating stable convergence. The narrow shaded areas—representing standard deviation across five random seeds—suggest low variance among runs. These results collectively highlight the effectiveness of the training process and the statistical reliability of the observed performance gains.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and

write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope, and are supported by the methodology and results presented in the main body.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work in the Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, the paper provides the full set of assumptions and corresponding proofs for each theoretical result in the Methodology section and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, the paper provides all necessary implementation and experimental details required to reproduce the main results, as described in the Method and Experiment sections. The code will be made publicly available upon publication.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will also release the code upon publication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test details are shown in experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Description about the statistical significance of the experiments are provided in Appendix B.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report compute resources in experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: This paper first time unify Detailed Balance and Trajectory Balance for solving vehicle routing problems, and the Hybrid-balance handle well with global-local optimization. This work has potential to have good impact in both GFlowNet and VRPs research community.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: For all models and datasets used, we cite its original papers.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not introduce any new assets at this stage. Upon publication, all code and datasets used in this work will be made publicly available. The methodology and experimental settings are thoroughly documented in paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.