

1	Contents	
2	A More Discussion on Related Works.	2
3	B Topological Properties of 0-order Pure Edges	2
4	C Verification of Purity Law on CVRP	3
5	D Detailed Description of Statistical Dataset	4
6	E Detailed Fitting Result of Purity Law	4
7	F Connection between Purity Law and k-Nearest Prior	5
8	G Proof of the supermodularity of Purity Availability ϕ	5
9	H The Whole PUPO Algorithm	6
10	I Further Discussion on Derivation Process and Explanation of PUPO	7
11	J Numerical Result on Two Classical Solvers	8
12	K Detailed Description of Dataset in Comparative Experiments	10
13	L Detailed Description of the Learning Rates	10
14	M Detailed Experimental Result	10

15 A More Discussion on Related Works.

16 With the rapid progress in deep learning, various neural approaches to combinatorial optimization
17 have emerged. Overviews of these methods can be found in Guo et al. [13] and Bengio et al. [4].

18 **Classical Neural Solvers.** According to the way the solutions are generated, neural approaches can
19 generally be divided into two classes: learning to directly construct and learning to iteratively improve
20 solutions. For constructive methods, Vinyals et al. [40] introduce the Pointer Network, which solves
21 routing problems end-to-end utilizing Recurrent Neural Networks to encode vertex embeddings,
22 trained via supervised learning. Kool et al. [23] first introduce the Transformer architecture [38]
23 to solve routing problems. Kwon et al. [25] further propose the policy optimization with multiple
24 optima (POMO), which improves the performance by exploiting solution symmetries. Meanwhile,
25 some works based on graph learning [20, 19] also show potential to solve routing problems. Pointer-
26 former [17] enhances memory efficiency by adopting a reversible residual network in the encoder
27 and a multi-pointer network in the decoder. For iterative methods, DRL-2opt [6] trains a DRL policy
28 to select appropriate 2-opt operators for refining solutions. Ma et al. [30] design dual-aspect collabora-
29 tive Transformer to learn embeddings for the node and positional features separately, iteratively
30 solving routing problems. GCN+MCTS [10] integrates graph decomposition and Monte Carlo Tree
31 Search to handle large-scale instances effectively but time-consuming.

32 Next, we will supplement more related work on improving generalization of neural routing problem
33 solvers. AMDKD [5] introduces knowledge distillation to tackle the cross-distribution generalization
34 concerns in the routing problem. Specifically, it leverages knowledge from teachers trained on
35 exemplar distributions to yield a generalist student model. Also along with knowledge distillation,
36 PDAM [43] adopts curriculum learning to train TSP samples in increasing order of their problem
37 size and progressively distilling high-level knowledge from small models to large models via a
38 distillation loss. SCA [22] designs a plugged network to mix scale information into the original
39 representation vector, which can help the pre-trained model adapt the policy to larger-scale tasks. By
40 studying adversarial robustness, Geisler [12] derives perturbation models for SAT and TSP to enhance
41 the expressiveness of the model with perturbations. Zhou [44] proposes a generic meta-learning
42 framework to enhance the ability of the initialized model to fast adapt to new tasks during inference
43 and conduct extensive experiments on VRPs. Sahil [31] Formalizes solving a CO problem over a
44 given instance distribution as a separate learning task and investigates meta-learning to optimize
45 the capacity of the model to adapt to new tasks. Reformulating the Markov Decision Process of
46 the solution construction in combinatorial optimization problems, BQ-NCO [7] propose a novel
47 Bisimulation Quotienting method for generalizable neural solver.

48 B Topological Properties of 0-order Pure Edges

49 As the class of edges with the lowest degree of redundancy, 0-order pure edges are demonstrated by
50 the following propositions to possess a series of favorable topological properties.

51 **Proposition B.1.** *Given an instance \mathcal{X} , for any vertex $x \in \mathcal{X}$, there exists at least one vertex that*
52 *can form a 0-order pure edge with x .*

53 *Proof.* We prove the proposition by contradiction.

Assume that for any $x \in \mathcal{X}$, we choose

$$y(x) := \arg \min \|x - y\|^2.$$

Since no vertex can form a 0-order pure edge with x , we have that

$$K_p(x, y(x)) > 0, \quad \forall x \in \mathcal{X}.$$

Therefore, there must exist a point $z(x)$ such that

$$(x - z(x))^T (y(x) - z(x)) < 0.$$

54 Then we can derive that

$$\begin{aligned}
& \|x - y(x)\|^2 \\
&= \|x - z(x) + z(x) - y(x)\|^2 \\
&= \|x - z(x)\|^2 + \|z(x) - y(x)\|^2 - 2(x - z(x))^T(y(x) - z(x)) \\
&\geq \|x - z(x)\|^2 + \|z(x) - y(x)\|^2 \\
&\geq \|x - z(x)\|^2,
\end{aligned}$$

55 which means that the distance between $z(x)$ and x is smaller than the distance between $y(x)$ and x .
56 Therefore, it contradicts the assumption that $y(x)$ is the nearest neighbor of x . \square

57 **Proposition B.2.** *Given an instance \mathcal{X} , the subgraph $G_0 = (\mathcal{X}, \mathcal{E}_0)$ is connected, where \mathcal{E}_0 is the*
58 *edge set of all 0-order pure edges.*

59 *Proof.* We prove the proposition by contradiction.

Suppose the subgraph G_0 is not connected. Without loss of generality, assume that G_0 has two connected components, denoted as F_1 and F_2 . Let $f_1 \in F_1$ and $f_2 \in F_2$ be such that

$$\text{dist}(f_1, f_2) = \text{dist}(F_1, F_2) = \min_{a \in F_1} \min_{b \in F_2} \text{dist}(a, b).$$

Since $e_{f_1 f_2} \notin G_0$, we have that

$$K_p(f_1, f_2) > 0.$$

Therefore, there must exist a point f_3 such that the following inequality holds:

$$(f_1 - f_3)^T(f_2 - f_3) < 0,$$

60 which means that the distance between f_3 and f_1 is smaller than the distance between f_2 and f_1 .
61 Regardless of whether $f_3 \in F_1$ or $f_3 \in F_2$, this leads to a contradiction with the assumption that the
62 distance between f_1 and f_2 is the shortest distance between F_1 and F_2 . \square

64 **Proposition B.3.** *Given an instance \mathcal{X} , for any vertex $x \in \mathcal{X}$, the polyhedron formed by its 0-order*
65 *pure neighbors D_0^x is convex.*

Proof. Consider any three adjacent 0-order pure neighbors A, B , and C of point X . We have that

$$\angle ABC = \angle ABX + \angle XBC.$$

Since A and C are 0-order pure neighbors of X , it follows that

$$\angle ABX, \angle XBC < \frac{\pi}{2},$$

66 and thus $\angle ABC < \pi$.

67 Due to the arbitrariness of A, B , and C , the polyhedron formed by its 0-order pure neighbors D_0^x is
68 convex. \square

69 Proposition B.1 establishes the existence of 0-order pure neighbors for any point, providing a
70 foundation for subsequent analysis. While proposition B.2 demonstrates that the subgraph formed by
71 0-order pure edges possesses overall graph structural properties. And proposition B.3 describes the
72 intrinsic topological features of the 0-order neighbor set for any given point. The three propositions
73 above characterize the mathematical properties of purity order theoretically.

74 C Verification of Purity Law on CVRP

75 Although different routing problems face distinct constraints, the common goal of VRPs is to
76 minimize the sum of the total tour cost, which is typically characterized by the spatial relationships
77 between nodes. The Purity Law describes a universal local structure of spatial relationships, which
78 are highly relevant to the common goal of VRPs. We also verify the Purity Law on CVRP. Using
79 the dataset from the original paper of INVIT [9], we computed the proportion (%) of edges with
80 purity orders $0 \sim 10$ in the optima. The result is shown in the table1. It can be observed that the
81 negative exponential law still obviously holds. And lowest-order pure edges still keep dominance in
82 the optimal solutions, while the concentration is slightly reduced due to capacity constraints.

Table 1: The verification result of Purity Law on CVRP.

Purity Order	CVRP-50				CVRP-500			
	Uniform	Clustered	Explosion	Implosion	Uniform	Clustered	Explosion	Implosion
0	78.3	76.5	78.2	78.4	66.6	65.8	67.3	66.8
1	13.2	12.9	13.3	13.2	11.3	11.2	11.2	11.3
2	5.5	5.6	5.5	5.3	4.3	4.4	4.3	4.4
3	3.1	3.1	3.0	3.1	2.4	2.5	2.3	2.4
4	2.1	2.2	2.0	2.1	1.6	1.6	1.5	1.6
5	1.5	1.6	1.6	1.5	1.1	1.2	1.1	1.1
6	1.2	1.3	1.3	1.2	0.9	1.0	0.9	0.9
7	1.0	1.0	1.0	1.0	0.7	0.7	0.7	0.7
8	0.9	0.9	0.8	0.9	0.7	0.7	0.6	0.6
9	0.7	0.7	0.7	0.8	0.5	0.6	0.5	0.6
10	0.7	0.6	0.6	0.6	0.5	0.5	0.5	0.5

Table 2: Fitting results of the exponential function.

	FITTING ERROR	α	β
MEAN	2.23E-05	0.92	2.63
VARIANCE	8.45E-10	1.57E-04	1.49E-02

83 D Detailed Description of Statistical Dataset

84 To construct the dataset in statistical experiments, we first generate 21 different scales within the range
85 of 20 to 1000, with intervals of 50 except 20. For each scale, we consider four widely recognized
86 classical distributions, which is uniform, cluster, explosion and implosion, resulting in 84 different
87 instance types totally. For instance types with scales under 500, 256 instances are randomly sampled
88 to form the dataset, while for those with sizes of 500 or above, the number of instances is reduced
89 to 128, which is due to the fact that optimal solutions of large-scale instances require an excessive
90 amount of time. The optimal solutions for all instances are solved by LKH-3 [15, 16], which is the
91 SOTA heuristic capable of producing optimal solutions even for large-scale instances.

92 E Detailed Fitting Result of Purity Law

93 The mean and variance of fitted parameters α and β , along with fitting errors are shown in Table
94 2. And the detailed values of the fitting error, α and β for each instance type are listed in Table 3.
95 According to the table, the low mean and variance of the fitting errors demonstrate the reliability and
96 universality of the fitting results across different instance scales and distributions.

Table 3: The fitting errors, coefficients α and β .

	Fitting Error				α				β			
	Uniform	Clustered	Explosion	Implosion	Uniform	Clustered	Explosion	Implosion	Uniform	Clustered	Explosion	Implosion
20	6.58E-05	2.45E-04	1.01E-04	5.71E-05	0.8860	0.8554	0.8835	0.8872	2.2514	2.0800	2.2483	2.2630
50	5.42E-05	7.90E-05	3.96E-05	5.05E-05	0.9115	0.8919	0.9027	0.9002	2.5166	2.3295	2.4152	2.3923
100	2.41E-05	3.75E-05	1.85E-05	3.61E-05	0.9204	0.9124	0.9091	0.9137	2.5974	2.5133	2.4567	2.5286
150	2.54E-05	2.57E-05	2.96E-05	1.78E-05	0.9236	0.9188	0.9150	0.9184	2.6410	2.5782	2.5412	2.5681
200	2.18E-05	2.20E-05	2.62E-05	1.63E-05	0.9236	0.9221	0.9197	0.9202	2.6384	2.6192	2.6002	2.5887
250	1.18E-05	1.35E-05	2.38E-05	2.42E-05	0.9263	0.9245	0.9209	0.9227	2.6568	2.6366	2.6066	2.6337
300	1.26E-05	1.49E-05	2.06E-05	2.08E-05	0.9259	0.9258	0.9211	0.9224	2.6535	2.6578	2.6057	2.6201
350	1.39E-05	1.40E-05	1.61E-05	1.71E-05	0.9273	0.9263	0.9236	0.9224	2.6762	2.6621	2.6318	2.6166
400	1.33E-05	1.54E-05	1.80E-05	1.05E-05	0.9278	0.9280	0.9241	0.9235	2.6814	2.6918	2.6430	2.6206
450	1.04E-05	1.12E-05	1.61E-05	1.52E-05	0.9280	0.9288	0.9234	0.9260	2.6805	2.6947	2.6289	2.6633
500	8.38E-06	1.15E-05	1.18E-05	1.28E-05	0.9273	0.9288	0.9256	0.9255	2.6634	2.6936	2.6515	2.6536
550	1.04E-05	1.49E-05	1.94E-05	1.31E-05	0.9306	0.9298	0.9257	0.9267	2.7190	2.7186	2.6615	2.6673
600	7.55E-06	1.08E-05	1.48E-05	1.26E-05	0.9294	0.9306	0.9259	0.9275	2.6931	2.7207	2.6591	2.6803
650	1.13E-05	1.37E-05	9.63E-06	1.79E-05	0.9297	0.9310	0.9262	0.9263	2.7058	2.7293	2.6535	2.6689
700	1.34E-05	1.02E-05	1.40E-05	1.77E-05	0.9280	0.9306	0.9277	0.9256	2.6856	2.7175	2.6877	2.6563
750	1.22E-05	1.19E-05	1.50E-05	1.26E-05	0.9287	0.9308	0.9264	0.9282	2.6959	2.7251	2.6658	2.6930
800	1.22E-05	1.29E-05	9.97E-06	1.19E-05	0.9294	0.9311	0.9255	0.9273	2.7040	2.7323	2.6459	2.6750
850	1.30E-05	1.12E-05	1.80E-05	1.33E-05	0.9300	0.9305	0.9276	0.9271	2.7151	2.7184	2.6909	2.6707
900	1.33E-05	1.03E-05	1.08E-05	1.46E-05	0.9309	0.9316	0.9276	0.9281	2.7296	2.7357	2.6790	2.6906
950	1.33E-05	1.38E-05	1.44E-05	1.41E-05	0.9299	0.9314	0.9284	0.9277	2.7132	2.7376	2.6951	2.6855
1000	1.51E-05	1.21E-05	1.26E-05	1.24E-05	0.9310	0.9318	0.9282	0.9289	2.7340	2.7413	2.6890	2.6978

97 F Connection between Purity Law and k-Nearest Prior

98 The k-nearest prior in routing problems refers to the observation that in optimal solutions, the next
 99 city visited is frequently among the k nearest neighbors of the current city, where k typically a small
 100 value. Purity Law and the k-nearest prior are both local phenomena that are commonly observed
 101 in optimal solutions of routing problems. However, the Purity Law can reflect more topological
 102 information.

103 First, the k-nearest neighbor with smaller values of that exhibit good structural quality tend to
 104 have lower purity orders for the edges they form. Second, the purity order reflects local structural
 105 information that the k-nearest prior can not capture. For example, in Fig. 1, both x_1 and x_2 are
 106 the 5-NN of z , and they are equivalent under the k-NN metric. However, under the purity order
 107 metric, they are distinguished. The neighbor x_1 with a lower purity order is a more ideal candidate
 108 for connecting to z compared to x_2 , due to its more coherent local structure. Thus, Purity Law
 109 not only contains distance information but also incorporates the structural information of the node
 110 distribution around the edges. This richer information carried by Purity Law allows it to contribute
 111 more effectively to generalization.

112 G Proof of the supermodularity of Purity Availability ϕ

113 **Proposition G.1.** *The set function $\phi : 2^X \rightarrow \mathbb{R}$ defined on the subsets of the finite set X is*
 114 *supermodular, that is, for any subset $A \subseteq B \subseteq X$ and any $x \in X \setminus B$, the following inequality*
 115 *holds:*

$$\phi(A \cup \{x\}) - \phi(A) \leq \phi(B \cup \{x\}) - \phi(B).$$

116 *Proof.* Given subsets $U \subseteq X$ and any $v_1, v_2 \in X \setminus U$, we prove the equivalent definition of
 117 supermodular functions:

$$\phi(U \cup \{v_1\}) + \phi(U \cup \{v_2\}) \leq \phi(U \cup \{v_1, v_2\}) + \phi(U). \quad (1)$$

118 First, we have

$$\begin{aligned} \phi(U) &= \frac{\sum_{x_i \in U} \min_{\substack{x_j \in U \\ j \neq i}} K_p(x_i, x_j)}{|U|}. \\ \phi(U \cup \{v_1\}) &= \frac{\sum_{x_i \in U} \min_{\substack{x_j \in U \cup \{v_1\} \\ j \neq i}} K_p(x_i, x_j) + \min_{x_j \in U} K_p(v_1, x_j)}{|U| + 1} \\ \phi(U \cup \{v_2\}) &= \frac{\sum_{x_i \in U} \min_{\substack{x_j \in U \cup \{v_2\} \\ j \neq i}} K_p(x_i, x_j) + \min_{x_j \in U} K_p(v_2, x_j)}{|U| + 1} \\ \phi(U \cup \{v_1, v_2\}) &= \frac{\sum_{x_i \in U} \min_{\substack{x_j \in U \cup \{v_1, v_2\} \\ j \neq i}} K_p(x_i, x_j) + \min_{x_j \in U \cup \{v_2\}} K_p(v_1, x_j) + \min_{x_j \in U \cup \{v_1\}} K_p(v_2, x_j)}{|U| + 2}. \end{aligned}$$

119 By the following relations, we partition U into three parts, denoted as U_0, U_1, U_2 , respectively.

$$U_0 = \{x_i \mid \arg \min_y K_p(x_i, y) \in U\}$$

$$U_1 = \{x_i \mid \arg \min_y K_p(x_i, y) = v_1\}$$

$$U_2 = \{x_i \mid \arg \min_y K_p(x_i, y) = v_2\}$$

120 Then, we prove that the following expression is non-positive from three parts:

$$(\phi(U \cup \{v_1\}) + \phi(U \cup \{v_2\})) - (\phi(U \cup \{v_1, v_2\}) + \phi(U)).$$

121 **Part 1**

$$\begin{aligned} & \sum_{x_i \in U_0} \min_{x_j \in U} K_p(x_i, x_j) \left[\frac{2}{|U|+1} - \left(\frac{1}{|U|} + \frac{1}{|U|+2} \right) \right] \\ & \leq \sum_{x_i \in U_0} \min_{x_j \in U} K_p(x_i, x_j) \left(\frac{-1}{|U|(|U|+1)(|U|+2)} \right) \leq 0 \end{aligned}$$

122 **Part 2**

$$\begin{aligned} & \sum_{x_i \in U_1} \left(\left(\frac{\min_{x_j \in U} K_p(x_i, x_j)}{|U|+1} - \frac{\min_{x_j \in U} K_p(x_i, x_j)}{|U|} \right) + \left(\frac{K_p(x_i, v_1)}{|U|+1} - \frac{K_p(x_i, v_1)}{|U|+2} \right) \right) \\ & + \left(\frac{\min_{x_j \in U} K_p(x_j, v_1)}{|U|+1} - \frac{\min_{x_j \in U \cup \{v_2\}} K_p(x_j, v_1)}{|U|+2} \right) \\ & = \sum_{x_i \in U_1} \left(-\frac{\min_{x_j \in U} K_p(x_i, x_j)}{|U|(|U|+1)} + \frac{K_p(x_i, v_1)}{(|U|+1)(|U|+2)} \right) + \frac{\min_{x_j \in U} K_p(x_j, v_1)}{(|U|+1)(|U|+2)} \\ & \leq \sum_{x_i \in U_1} \left(-\frac{K_p(x_i, v_1) + 1}{|U|(|U|+1)} + \frac{K_p(x_i, v_1)}{(|U|+1)(|U|+2)} \right) + \frac{\min_{x_j \in U} K_p(x_j, v_1)}{(|U|+1)(|U|+2)} \\ & = \frac{1}{|U|(|U|+1)(|U|+2)} \left(-2 \sum_{x_j \in U_1} K_p(x_i, v_1) + |U| \min_{x_j \in U} K_p(x_j, v_1) - |U_1|(|U|+2) \right) \\ & \leq \frac{1}{|U|(|U|+1)(|U|+2)} \left(-2|U_1| \min_{x_j \in U} K_p(x_j, v_1) + |U| \min_{x_j \in U} K_p(x_j, v_1) - |U_1|(|U|+2) \right) \\ & \leq \frac{1}{|U|(|U|+1)(|U|+2)} (-2|U_1|^2 - 2|U_1|) \leq 0 \end{aligned}$$

123 **Part 3**

$$\begin{aligned} & \sum_{x_i \in U_2} \left(\left(\frac{\min_{x_j \in U} K_p(x_i, x_j)}{|U|+1} - \frac{\min_{x_j \in U} K_p(x_i, x_j)}{|U|} \right) + \left(\frac{K_p(x_i, v_2)}{|U|+1} - \frac{K_p(x_i, v_2)}{|U|+2} \right) \right) \\ & + \left(\frac{\min_{x_j \in U} K_p(x_j, v_2)}{|U|+1} - \frac{\min_{x_j \in U \cup \{v_1\}} K_p(x_j, v_2)}{|U|+2} \right) \\ & = \sum_{x_i \in U_2} \left(-\frac{\min_{x_j \in U} K_p(x_i, x_j)}{|U|(|U|+1)} + \frac{K_p(x_i, v_2)}{(|U|+1)(|U|+2)} \right) + \frac{\min_{x_j \in U} K_p(x_j, v_2)}{(|U|+1)(|U|+2)} \\ & \leq \sum_{x_i \in U_2} \left(-\frac{K_p(x_i, v_2) + 1}{|U|(|U|+1)} + \frac{K_p(x_i, v_2)}{(|U|+1)(|U|+2)} \right) + \frac{\min_{x_j \in U} K_p(x_j, v_2)}{(|U|+1)(|U|+2)} \\ & = \frac{1}{|U|(|U|+1)(|U|+2)} \left(-2 \sum_{x_j \in U_2} K_p(x_i, v_2) + |U| \min_{x_j \in U} K_p(x_j, v_2) - |U_2|(|U|+2) \right) \\ & \leq \frac{1}{|U|(|U|+1)(|U|+2)} \left(-2|U_2| \min_{x_j \in U} K_p(x_j, v_2) + |U| \min_{x_j \in U} K_p(x_j, v_2) - |U_2|(|U|+2) \right) \\ & \leq \frac{1}{|U|(|U|+1)(|U|+2)} (-2|U_2|^2 - 2|U_2|) \leq 0 \end{aligned}$$

124 The gain consists of three parts, each of which is non-positive. Therefore, the overall gain is
125 non-positive, and the supermodular property is thus proven.

126

□

127 H The Whole PUPO Algorithm

128 The whole algorithm of PUPO is presented in Algorithm 1. PUPO is established upon the REIN-
129 FORCE with baseline, incorporating Purity Law information into the modified policy gradient. At

Algorithm 1 PUPO Training

Input: number of epochs E , steps per epoch M , batch size B , scale of training instances N , discount factor γ , learning rate δ
Init $\theta, \theta^{BL} \leftarrow \theta$
for epoch = 1, ..., E **do**
 for step = 1, ..., M **do**
 $\forall i \in \{1, 2, \dots, B\}$
 $s_i \leftarrow \text{RandomInstance}()$
 $\tau^i \leftarrow \text{SampleRollout}(s_i, p_\theta)$
 $\tau^{i,BL} \leftarrow \text{GreedyRollout}(s_i, p_{\theta^{BL}})$
 $U_0^i \leftarrow s_i$
 for time = 1, ..., $N - 1$ **do**
 $U_t^i \leftarrow U_{t-1}^i \setminus \tau_t^i$
 $\phi(U_t^i) \leftarrow \sum_{a \in U_t^i} \min_{\substack{b \in U_t^i \\ b \neq a}} K_p(a, b, s_i)$
 $C(U_t^i, \tau_{t+1}^i) \leftarrow K_p(\tau_t^i, \tau_{t+1}^i, s_i) + \frac{\phi(U_{t+1}^i)}{|U_{t+1}^i|} - \frac{\phi(U_t^i)}{|U_t^i|}$
 $W_{t+1}^i \leftarrow 1 + \sum_{j=t}^N \gamma^{j-t} C(U_j^i, \tau_{j+1}^i)$
 end for
 $\nabla \hat{\mathcal{L}} \leftarrow \frac{1}{B} \sum_{i=1}^B (L(\tau^i) - L(\tau^{i,BL})) \cdot \left(\sum_{t=2}^N W_t^i \nabla_\theta \log p_\theta(\tau_t^i | \tau_{1:t-1}^i, s_i) \right)$
 $\theta \leftarrow \theta + \delta \nabla \hat{\mathcal{L}}$
 end for
 $\theta^{BL} \leftarrow \text{UpdateBaseline}(\theta, \theta^{BL})$
end for

different stages of solution construction, PUPO introduces discounted cost information that reflects the purity potential of the partial solution and the unvisited vertex set, which encourages the model to take actions with lower purity orders at each step during the training process. This approach helps the model to learn consistent patterns with the Purity Law prior, an information that is independent of specific instances, thereby enhancing its generalization ability. Notably, PUPO can be easily integrated with arbitrary existing constructive neural solvers, without any alterations to network architecture.

I Further Discussion on Derivation Process and Explanation of PUPO

In this section, We will discuss the derivation process and explanation of PUPO in more detail. For convenience, W_{t+1} denotes the shorthand for $W(\mathcal{U}_t, \tau_{t+1})$.

Enhancing the purity potential of the unvisited vertex set at each step is essential for learning a policy with strong generalization capability. For intuitive understanding, we plot a specific example shown in Fig. 1. In this figure, policy (a) selects myopic local optima in the earlier decision stages. In the later stages, constrained by the requirement to select vertices only from the unvisited vertex set, it is forced to connect some "bad" edges with higher purity orders (see red edges). In contrast, strategy (b) constructs a tour with a higher overall purity level. For solving instances of the same type during training, the tour lengths obtained by strategies (a) and (b) may not differ significantly. However, when generalized to other instances, myopic strategies like (a), which result in poor purity in the later stages, are more likely to lead to tours of lower quality.

Inspired by supermodularity, we design purity weights to assess the potential purity of states at each time step for future decision under different policies. We constructed a Purity Availability metric with supermodular properties to measure the purity potential of the unvisited vertex set. On one hand, since Purity Availability represents the average minimum available purity order, higher values indicate that the state will not be able to generate highly pure structures in the future. For example, in the figure above, the Purity Availability of strategies (a) and (b) at the current state

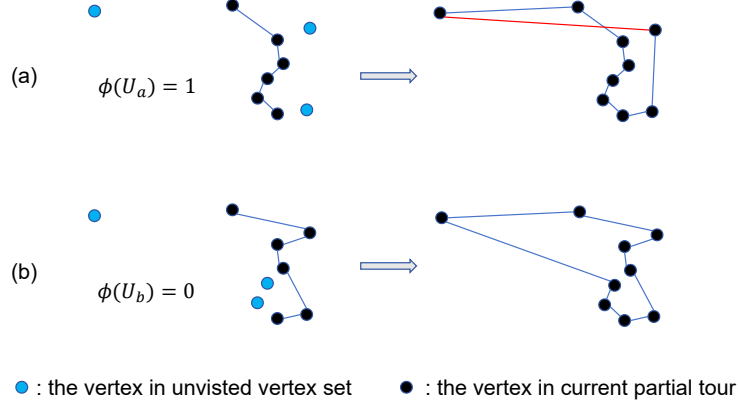


Figure 1: Example of two purity potential.

are 1 and 0, respectively. It means that policy (b) possess ability to construct purer structures in the future. On the other hand, constrained by the necessity to select nodes only from the unvisited vertex set, later decision stages often lead to states that are unfavorable for generalization and difficult to distinguish under tour length metrics. Due to supermodularity, the marginal benefit of Purity Availability, $\phi(\mathcal{U}_t) - \phi(\mathcal{U}_{t-1})$, is greater at later stages, making the purity measure in the later decision phases more significant. Inspired by supermodularity, we designed the purity cost based on the marginal benefit of state and the purity order of the actions. Furthermore, we designed purity weights based on the concept of future discounting to comprehensively assess the future purity potential of intermediate states during policy transitions.

Through analysis of upper and lower bounds and experimental analysis, PUPO encourages the exploration of higher-quality states under the purity measure. First, we present a pair of upper and lower bounds for Eq. (12). Since $W_t \geq 1$, the following inequality holds:

$$\begin{aligned}
 & E_{p_\theta(\tau|\mathcal{X})} \left[L(\tau) \sum_{t=2}^N \nabla \log p_\theta(\tau_t | \tau_{1:t-1}, \mathcal{X}) \right] \\
 & \leq E_{p_\theta(\tau|\mathcal{X})} \left[L(\tau) \sum_{t=2}^N W_t \nabla \log p_\theta(\tau_t | \tau_{1:t-1}, \mathcal{X}) \right] \\
 & \leq E_{p_\theta(\tau|\mathcal{X})} \left[L(\tau) \left(\sum_{t=2}^N W_t \right) \sum_{t=2}^N \nabla \log p_\theta(\tau_t | \tau_{1:t-1}, \mathcal{X}) \right].
 \end{aligned}$$

The left-hand side of the inequality represents the original policy gradient with tour length as the reward, while the right-hand side represents the policy gradient with $\left(\sum_{t=2}^N W_t \right) L(\tau)$ as the reward.

In this reward, the purity weights act as multiplicative factors on the tour length. It means that the overall purity of the tour is also incorporated into the reward information. The policy gradient of PUPO lies between the two, applying each W_t to the logarithm of action probabilities at each time step. This more subtle approach encourages the exploration of strategies that can transition to states with higher quality under the purity measure.

J Numerical Result on Two Classical Solvers

We also conduct experiments on two classical neural solvers POMO and PF. Table 4 show the results on random dataset of TSP. It can be observed that PUPO training enhances the performance of four classical models across nearly all instance types. In particular, AM-100 and PF-100 demonstrate superior performance across all distributions and scales compared to those trained with the vanilla. Table 5 also shows the results on TSPLIB, and Table 6 shows the result on random dataset of CVRP.

180 It can be observed that for CVRP, PUPO training enhances the performance more significantly.
181 Although PUPO also enhanced the generalizability of AM and PF, their global perception and
182 decision modules remain vulnerable, resulting in larger performance gaps as the scale increases. This
183 observation abstracts us to focus on developing network architectures that incorporate Purity Law in
184 future work.

Table 4: The experimental results of average gap (%) on the randomly generated dataset with different distributions and scales of TSP after training two classical model using both the vanilla and PUPO methods, where bold formatting represents superior results.

Instances	POMO-50		POMO-100		PF-50		PF-100	
	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO
U-100	5.21	5.35	5.07	4.61	4.16	4.31	3.64	3.39
U-1000	35.51	34.01	31.23	30.16	41.00	39.10	30.57	29.93
U-5000	67.99	65.12	80.64	68.42	113.55	104.09	80.59	77.53
C-100	7.75	8.03	7.66	6.07	7.67	7.71	7.24	6.90
C-1000	41.14	39.82	37.99	33.84	51.04	49.58	47.50	44.11
C-5000	77.78	75.93	76.37	70.03	137.51	131.61	137.15	121.79
E-100	5.01	5.18	5.04	4.71	4.98	5.12	4.74	4.65
E-1000	35.57	35.07	32.97	30.20	46.04	45.34	41.60	37.73
E-5000	71.95	66.66	74.55	66.33	133.99	129.02	120.96	110.16
I-100	5.02	5.17	5.12	4.64	4.71	4.91	4.49	4.35
I-1000	37.38	37.12	32.44	30.94	46.54	44.83	36.11	34.47
I-5000	78.57	74.05	72.87	66.58	151.37	138.71	120.67	103.74

Table 5: Performance of average gap (%) on TSPLIB after training two classical model using both the vanilla and PUPO methods.

	POMO-50		POMO-100		PF-50		PF-100	
	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO
1~100	5.82	5.17	4.98	4.48	7.42	8.37	7.82	7.16
101~1000	17.39	16.14	14.12	12.17	25.96	24.77	23.37	22.68
1001~5000	48.41	46.48	43.33	40.46	75.57	72.58	66.05	60.86

Table 6: The experimental results of average gap (%) on the randomly generated dataset with different distributions and scales of CVRP after training POMO model using both the vanilla and PUPO methods, where bold formatting represents superior results.

Instance	POMO-50		POMO-100	
	Vanilla	PUPO	Vanilla	PUPO
U-50	4.6	5.704	6.146	6.769
U-500	16.379	14.311	14.843	13.383
U-5000	22.846	16.122	20.599	18.067
C-50	5.204	6.346	6.388	6.384
C-500	14.997	13.21	12.837	12.279
C-5000	15.803	11.694	13.172	13.376
E-50	4.624	5.851	6.042	6.704
E-500	16.227	14.622	14.249	13.525
E-5000	22.761	17.148	21.638	17.673
I-50	4.57	5.794	6.117	6.911
I-500	15.744	14.239	14.033	13.16
I-5000	19.703	14.64	16.513	16.214

K Detailed Description of Dataset in Comparative Experiments

To ensure the adequacy of the experiments, we validate the model’s performance on two categories of datasets. The randomly generated dataset used in this paper is the same as that in INVIT [9], which is widely adopted to testify existing DRL approach. For TSP, it contains 16 subsets and corresponding (near-)optimal solutions for TSP, including 4 distributions (uniform, clustered, explosion, and implosion, denoted as U, C, E, I) and 4 scales (100, 1000, 5000 and 10000). For CVRP, it contains 12 subsets and corresponding (near-)optimal solutions for CVRP, including 4 distributions (uniform, clustered, explosion, and implosion, denoted as U, C, E, I) and 3 scales (50, 500 and 5000).

The real-world dataset we used is TSPLIB and CVRPLIB. TSPLIB is a well-known TSP library [36] that contains 100 instances with various nodes distributions and their optimal solutions. These instances come from practical applications with scale ranging from 14 to 85,900. In our experiment, we consider all instances with no more than 10000 nodes. For CVRP, we include all instances in CVRPLIB Set-X [37], containing 100 instances varying in scale from 100 to 1000.

L Detailed Description of the Learning Rates

Specifically, for TSP, the learning rates are set to 0.0001, 0.00015, 0.0001, 0.00017, 0.0001, 0.00012, 0.00011, 0.00012 for POMO-50, POMO-100, PF-50, PF-100, ELG-50, ELG-100, INVIT-50, and INVIT-100, respectively. For CVRP, the learning rates are set to 0.0001, 0.00006, 0.0001, 0.0001, 0.00005, 0.0001 for POMO-50, POMO-100, ELG-50, ELG-100, INVIT-50, and INVIT-100.

M Detailed Experimental Result

Table 7 shows the training time per epoch with different methods. We can see that the training time of PUPO does not increase significantly owing to the tensorizable computation.

Table 7: The numerical results of training time (minutes) per epoch during different training.

	POMO-50	POMO-100	INVIT-50	INVIT-100
Vanilla	3.05	4.49	7.72	15.75
PUPO	5.95	14.20	11.598	21.40

The experimental results of the specific tour length is presented in Table 8. Table 9 illustrates the solving time of Vanilla-trained and PUPO-trained models on randomly generated dataset. Following the table, it can be observed that there is almost no difference between Vanilla and PUPO. The numerical results of purity metrics for all models are presented in Table 10. Following the table, PUPO-trained models possess the ability to generate solutions with more outstanding purity. Detailed results of ELG and INVIT are provided in Table 11 to Table 22.

Table 8: The length of tours generated from each model after different training on randomly generated dataset of TSP.

		POMO-50		POMO-100		PF-50		PF-100		INVIT-50		INVIT-100	
		Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO	Vanilla	PUPO
Uniform	100	8.28	8.29	8.27	8.23	8.19	8.21	8.15	8.13	8.03	8.04	8.07	8.05
	1000	31.47	31.12	30.47	30.22	32.74	32.30	30.32	30.17	24.91	24.88	24.94	24.67
	5000	85.75	84.28	92.20	85.97	109.00	104.18	92.18	90.62	55.85	55.69	55.84	55.15
	10000	-	-	-	-	-	-	-	-	79.07	78.88	78.79	77.87
Clustered	100	5.72	5.73	5.71	5.63	5.71	5.71	5.68	5.67	5.46	5.46	5.50	5.48
	1000	19.82	19.65	19.39	18.80	21.26	21.05	20.78	20.30	15.25	15.19	15.24	15.07
	5000	53.06	52.50	52.64	50.77	70.86	69.10	70.71	66.13	32.79	32.66	32.73	32.39
	10000	-	-	-	-	-	-	-	-	44.88	44.58	44.72	44.17
Explosion	100	6.86	6.87	6.86	6.83	6.86	6.87	6.84	6.83	6.67	6.66	6.70	6.68
	1000	21.90	21.84	21.47	21.03	23.50	23.36	22.72	22.10	17.67	17.60	17.67	17.53
	5000	57.30	55.81	58.14	55.62	77.35	75.30	72.57	69.02	37.52	37.28	37.23	36.88
	10000	-	-	-	-	-	-	-	-	43.28	43.05	43.18	42.66
Implosion	100	7.48	7.49	7.48	7.45	7.45	7.47	7.43	7.42	7.29	7.29	7.32	7.30
	1000	27.61	27.57	26.63	26.33	29.39	29.04	27.28	26.95	21.64	21.63	21.69	21.51
	5000	72.59	71.24	71.28	68.46	100.18	95.14	86.94	80.27	44.96	44.83	45.05	44.78
	10000	-	-	-	-	-	-	-	-	72.24	71.79	72.17	71.18

Table 9: The solving time for each model with different training methods on randomly generated dataset of TSP.

	POMO-50		POMO-100		PF-50		PF-100		INVt-50		INVt-100	
	Vanilla	PUPU	Vanilla	PUPU	Vanilla	PUPU	Vanilla	PUPU	Vanilla	PUPU	Vanilla	PUPU
U-100	0.20	0.22	0.21	0.22	2.02	1.97	1.00	1.02	0.85	0.80	1.07	0.91
U-1000	2.35	2.34	2.41	2.40	23.67	24.08	13.57	14.18	13.24	12.88	17.83	17.34
U-5000	19.68	19.10	20.00	19.57	246.65	246.12	205.14	204.15	91.20	89.44	120.07	119.87
U-10000	-	-	-	-	-	-	-	-	224.99	223.69	288.44	287.11
C-100	0.21	0.21	0.21	0.21	2.08	2.13	0.97	0.98	0.84	0.80	1.04	1.17
C-1000	2.35	2.35	2.44	2.43	25.28	26.11	14.05	15.28	13.00	12.77	17.57	16.56
C-5000	19.16	19.38	19.99	19.35	275.68	274.49	212.36	211.68	89.48	88.23	118.63	117.31
C-10000	-	-	-	-	-	-	-	-	222.47	222.32	288.67	288.90
E-100	0.21	0.21	0.20	0.22	2.18	2.17	0.97	1.09	0.82	0.83	1.05	0.94
E-1000	2.38	2.36	2.46	2.40	25.44	24.53	13.94	13.82	12.94	13.05	17.70	17.98
E-5000	19.30	18.67	19.98	19.57	411.08	411.22	213.01	212.39	89.58	89.93	118.93	118.34
E-10000	-	-	-	-	-	-	-	-	223.24	223.03	288.86	288.32
I-100	0.21	0.20	0.21	0.21	3.02	2.89	1.00	0.97	0.83	0.79	1.05	1.01
I-1000	2.37	2.45	2.45	2.45	20.84	21.01	14.08	13.99	12.91	12.73	17.60	17.31
I-5000	19.95	19.28	19.84	19.30	445.75	444.87	186.14	186.05	89.92	88.30	118.87	118.69
I-10000	-	-	-	-	-	-	-	-	222.56	221.72	289.39	290.99

Table 10: The numerical results of purity evaluation for three models on TSP after different training.

	POMO-50						POMO-100					
	Prop-0 (%)	Vanilla APO (all)	APO (non-0)	Prop-0 (%)	PUPU APO (all)	APO (non-0)	Prop-0 (%)	Vanilla APO (all)	APO (non-0)	Prop-0 (%)	PUPU APO (all)	APO (non-0)
U-100	75.65%	0.15	1.11	75.29%	0.15	1.11	76.81%	0.15	1.12	79.48%	0.14	1.17
U-1000	62.51%	0.64	1.98	64.32%	0.62	1.92	65.67%	0.52	1.75	67.70%	0.52	1.76
U-5000	24.91%	1.36	2.97	26.39%	1.33	3.05	24.54%	1.50	3.03	26.64%	1.23	2.74
U-10000	-	-	-	-	-	-	-	-	-	-	-	-
C-100	70.66%	0.23	1.26	70.13%	0.24	1.28	72.34%	0.24	1.30	75.35%	0.20	1.28
C-1000	59.13%	0.85	2.30	58.29%	0.80	2.19	61.18%	0.77	2.17	62.52%	0.66	1.97
C-5000	23.65%	1.93	3.88	23%	1.64	3.33	23.58%	1.64	3.31	24.76%	1.48	3.05
C-10000	-	-	-	-	-	-	-	-	-	-	-	-
E-100	73.72%	0.17	1.15	74.35%	0.17	1.14	75.24%	0.17	1.17	77.77%	0.17	1.22
E-1000	62.92%	0.69	2.03	61.35%	0.67	2.03	63.32%	0.62	1.89	65.58%	0.57	1.84
E-5000	25.09%	1.53	3.27	23.99%	1.41	3.06	24.20%	1.55	3.13	26.04%	1.30	2.82
E-10000	-	-	-	-	-	-	-	-	-	-	-	-
I-100	72.97%	0.19	1.20	73.50%	0.19	1.19	74.80%	0.19	1.23	77.27%	0.18	1.26
I-1000	61.52%	0.77	2.22	60.27%	0.77	2.20	64.14%	0.65	2.01	65.01%	0.62	1.95
I-5000	24.05%	1.93	3.83	23.70%	1.69	3.41	24.12%	1.65	3.33	25.30%	1.46	3.07
I-10000	-	-	-	-	-	-	-	-	-	-	-	-
	PF-50						PF-100					
	Prop-0 (%)	Vanilla APO (all)	APO (non-0)	Prop-0 (%)	PUPU APO (all)	APO (non-0)	Prop-0 (%)	Vanilla APO (all)	APO (non-0)	Prop-0 (%)	PUPU APO (all)	APO (non-0)
U-100	77.46%	0.14	1.09	77.57%	0.15	1.10	80.07%	0.12	1.08	80.31%	0.12	1.08
U-1000	59.68%	0.74	2.00	60.73%	0.70	1.94	65.66%	0.56	1.78	66.06%	0.54	1.75
U-5000	18.66%	2.36	3.92	20.04%	2.14	3.65	23.09%	1.56	2.94	23.60%	1.49	2.92
U-10000	-	-	-	-	-	-	-	-	-	-	-	-
C-100	71.41%	0.24	1.27	71.53%	0.25	1.29	72.49%	0.24	1.29	73.64%	0.23	1.28
C-1000	52.39%	1.21	2.67	52.53%	1.17	2.60	54.38%	1.09	2.54	56.51%	0.97	2.37
C-5000	15.08%	4.52	6.54	15.42%	4.06	5.95	16.08%	4.20	6.31	18%	3.13	4.93
C-10000	-	-	-	-	-	-	-	-	-	-	-	-
E-100	74.92%	0.19	1.16	74.82%	0.18	1.17	76.63%	0.17	1.17	77.34%	0.16	1.16
E-1000	55.35%	0.98	2.33	55.95%	0.98	2.34	57.53%	0.89	2.21	59.66%	0.81	2.12
E-5000	16.05%	3.52	5.18	16.64%	3.39	5.10	17.60%	3.01	4.62	18.77%	2.73	4.40
E-10000	-	-	-	-	-	-	-	-	-	-	-	-
I-100	73.76%	0.24	1.30	73.68%	0.24	1.28	75.67%	0.22	1.29	76.24%	0.22	1.29
I-1000	54.17%	1.40	3.09	55.04%	1.42	3.15	59.29%	1.14	2.79	60.07%	1.13	2.81
I-5000	15.18%	8.14	10.81	15.90%	8.46	11.59	17.79%	7.46	10.38	18.88%	6.02	8.88
I-10000	-	-	-	-	-	-	-	-	-	-	-	-
	INVt-50						INVt-100					
	Prop-0 (%)	Vanilla APO (all)	APO (non-0)	Prop-0 (%)	PUPU APO (all)	APO (non-0)	Prop-0 (%)	Vanilla APO (all)	APO (non-0)	Prop-0 (%)	PUPU APO (all)	APO (non-0)
U-100	81.52%	0.10	1.05	0.82	0.10	1.04	80.95%	0.11	1.07	0.82	0.10	1.05
U-1000	86%	0.16	1.37	0.86	0.15	1.33	85.80%	0.17	1.43	0.87	0.14	1.33
U-5000	43.52%	0.18	1.56	0.44	0.17	1.05	43.38%	0.20	1.61	0.44	0.17	1.48
U-10000	87.39%	0.18	1.55	0.88	0.17	1.50	87.32%	0.20	1.65	0.88	0.16	1.47
C-100	79.78%	0.14	1.18	0.80	0.13	1.17	79.26%	0.16	1.26	0.80	0.14	1.19
C-1000	86.02%	0.22	1.88	0.86	0.19	1.63	85.56%	0.23	1.83	0.87	0.18	1.61
C-5000	43.72%	0.20	1.61	0.44	0.18	1.55	43.32%	0.27	2.20	0.44	0.17	1.52
C-10000	87.40%	0.84	6.91	0.88	0.88	7.34	87.15%	0.71	5.74	0.88	0.87	7.77
E-100	80.56%	0.12	1.11	0.81	0.12	1.11	80.09%	0.13	1.15	0.81	0.12	1.11
E-1000	85.42%	0.30	2.34	0.86	0.28	2.23	85.09%	0.30	2.33	0.86	0.26	2.16
E-5000	43.54%	0.42	3.47	0.44	0.48	4.07	43.40%	0.53	4.33	0.44	0.43	3.73
E-10000	87%	0.95	7.68	0.87	1.03	8.38	87%	0.97	7.58	0.88	0.88	7.57
I-100	79.98%	0.12	1.12	0.80	0.12	1.12	79.77%	0.14	1.17	0.80	0.13	1.13
I-1000	85.69%	0.19	1.54	0.86	0.18	1.54	85.39%	0.21	1.64	0.86	0.18	1.55
I-5000	43.39%	0.23	1.88	0.44	0.22	1.81	43.33%	0.24	1.92	0.44	0.19	1.66
I-10000	87.34%	0.30	2.47	0.88	0.29	2.39	87.16%	0.27	2.18	0.88	0.30	2.62

Table 11: Detailed results of ELG-TSP-50 [Vanilla] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.167	8.385	6.599	1.095	13.377	0.019
U-500	1.638	28.051	20.805	16.915	25.198	0.017
U-5000	21.006	67.121	31.496	30.52	32.456	0.009
C-50	0.17	5.838	10.047	3.552	20.179	0.03
C-500	2.027	17.867	27.142	19.898	34.527	0.034
C-5000	19.721	41.468	38.888	35.129	42.994	0.03
E-50	0.169	6.97	6.848	1.456	19.511	0.024
E-500	2.079	20.064	24.554	17.191	32.079	0.036
E-5000	19.915	45.129	35.672	32.112	42.508	0.045
I-50	0.167	7.591	6.655	1.143	14.219	0.021
I-500	1.96	24.401	21.303	17.254	25.035	0.02
I-5000	19.832	53.693	30.403	28.01	32.592	0.018

Table 12: Detailed results of ELG-TSP-50 [PUPO] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.163	8.425	7.102	0.73	13.637	0.018
U-500	1.667	27.742	19.474	15.497	22.476	0.015
U-5000	21.34	64.744	26.84	24.991	28.767	0.015
C-50	0.169	5.783	8.982	2.337	17.056	0.026
C-500	2.012	17.461	24.132	18.047	29.298	0.024
C-5000	19.84	40.475	35.625	34.174	38.202	0.016
E-50	0.171	6.996	7.234	0.849	15.83	0.022
E-500	2.139	19.941	23.767	18.862	30.275	0.032
E-5000	19.73	42.989	28.882	26.586	32.44	0.022
I-50	0.172	7.629	7.233	1.79	19.087	0.023
I-500	2	24.238	20.544	16.96	25.203	0.019
I-5000	19.781	52.342	27.644	25.039	30.885	0.021

Table 13: Detailed results of ELG-TSP-100 [Vanilla] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.17	8.327	5.859	1.414	10.455	0.017
U-500	1.714	27.389	17.954	15.157	21.448	0.016
U-5000	21.802	65.96	29.223	27.378	30.284	0.011
C-50	0.18	5.912	11.495	3.5	33.416	0.041
C-500	2.135	17.842	27.057	17.086	53.627	0.059
C-5000	20.303	41.904	40.44	35.008	53.044	0.073
E-50	0.177	6.991	7.277	1.466	32.814	0.037
E-500	2.164	19.952	24.007	15.306	34.444	0.046
E-5000	20.288	44.711	34.771	28.244	53.191	0.104
I-50	0.181	7.582	6.639	1.27	53.637	0.037
I-500	2.129	23.869	18.621	13.569	23.293	0.021
I-5000	20.454	53.499	30.701	28.03	37.713	0.04

Table 14: Detailed results of ELG-TSP-100 [PUPO] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.169	8.416	6.394	1.544	11.726	0.018
U-500	1.642	27.513	18.491	15.788	21.281	0.013
U-5000	21.291	63.176	23.769	22.484	24.722	0.008
C-50	0.168	5.803	9.374	3.302	19.165	0.027
C-500	2.028	17.033	21.131	15.879	26.933	0.022
C-5000	19.813	38.576	29.285	27.614	30.616	0.011
E-50	0.17	6.993	7.188	1.913	15.156	0.022
E-500	2.094	19.599	21.568	16.944	27.233	0.025
E-5000	19.835	42.083	26.262	23.14	29.791	0.025
I-50	0.171	7.657	7.602	1.684	16.868	0.023
I-500	1.978	24.035	19.515	14.306	23.205	0.017
I-5000	19.735	51.01	24.636	21.293	28.75	0.031

Table 15: Detailed results of ELG-VRP-50 [Vanilla] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.182	10.597	7.446	2.438	15.163	0.024
U-500	1.702	79.532	15.478	10.783	20.443	0.027
U-5000	41.938	729.196	22.753	12.2	31.278	0.069
C-50	0.192	9.228	7.451	1.417	19.272	0.032
C-500	1.788	64.274	16.468	10.468	25.644	0.035
C-5000	31.429	681.954	20.448	14.562	29.403	0.063
E-50	0.189	9.461	7.577	1.767	16.394	0.026
E-500	1.783	60.506	15.855	11.473	26.178	0.032
E-5000	31.684	437.246	27.103	15.508	48.303	0.126
I-50	0.191	10.086	7.766	1.961	19.897	0.028
I-500	1.797	73.173	15.762	10.151	22.652	0.03
I-5000	31.711	700.698	19.427	15.743	28.109	0.051

Table 16: Detailed results of ELG-VRP-50 [PUPO] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.186	10.665	8.15	2.483	21.006	0.027
U-500	1.694	76.488	10.968	8.122	15.112	0.016
U-5000	31.128	647.568	8.312	7.469	10.66	0.013
C-50	0.189	9.21	7.275	1.289	20.119	0.029
C-500	1.759	60.866	10.214	6.542	15.321	0.02
C-5000	30.684	615.111	8.72	4.76	16.811	0.048
E-50	0.189	9.513	8.132	2.725	17.49	0.027
E-500	1.748	58.13	11.117	8.009	17.598	0.02
E-5000	30.737	373.572	8.516	6.827	11.048	0.018
I-50	0.187	10.127	8.22	2.307	17.804	0.028
I-500	1.767	70.051	10.83	7.665	14.91	0.016
I-5000	30.623	629.346	7.046	4.803	9.885	0.021

Table 17: Detailed results of ELG-VRP-100 [Vanilla] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.195	10.685	8.315	1.049	19.544	0.028
U-500	1.843	75.079	8.907	6.067	13.065	0.015
U-5000	33.265	631.606	5.969	4.003	8.587	0.018
C-50	0.199	9.22	7.395	1.519	21.493	0.032
C-500	1.856	60.128	8.818	4.944	13.067	0.018
C-5000	31.53	609.493	7.103	5.912	8.039	0.01
E-50	0.2	9.547	8.566	2.14	23.925	0.031
E-500	1.865	57.156	9.322	7.064	15.291	0.019
E-5000	31.831	372.201	8.259	6.408	9.693	0.013
I-50	0.201	10.144	8.344	1.643	20.51	0.031
I-500	1.888	68.729	8.97	5.076	14.439	0.017
I-5000	31.839	622.795	6.013	5.351	6.99	0.007

Table 18: Detailed results of ELG-VRP-100 [PUPO] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.178	10.811	9.093	2.543	22.455	0.031
U-500	1.673	75.034	8.842	6.207	11.874	0.015
U-5000	31.197	626.848	5.142	3.314	7.476	0.016
C-50	0.189	9.359	8.047	1.617	23.159	0.038
C-500	1.757	59.604	7.92	5.273	13.105	0.015
C-5000	30.857	597.049	5.434	2.846	6.627	0.016
E-50	0.19	9.647	9.148	1.991	23.308	0.034
E-500	1.757	56.838	8.676	6.645	12.068	0.015
E-5000	30.796	365.526	6.33	4.875	7.202	0.009
I-50	0.189	10.262	8.673	1.927	23.732	0.036
I-500	1.758	68.808	8.833	6.697	14.897	0.014
I-5000	30.911	617.372	5.237	3.594	6.488	0.011

Table 19: Detailed results of INVIT-VRP-50 [Vanilla] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.836	10.286	4.251	0.544	8.944	0.016
U-500	8.903	76.366	10.747	9.049	13.794	0.012
U-5000	137.071	654.376	9.672	8.218	10.969	0.01
C-50	0.74	8.978	4.5	0.718	15.507	0.024
C-500	8.035	60.676	9.786	7.865	11.631	0.008
C-5000	131.165	615.721	8.492	7.421	9.728	0.011
E-50	0.769	9.199	4.555	0.296	11.336	0.018
E-500	8.044	57.801	10.475	7.92	15.046	0.015
E-5000	131.003	376.968	9.641	8.73	10.735	0.009
I-50	0.743	9.775	4.408	0.427	12.283	0.019
I-500	7.756	69.659	10.148	7.872	12.083	0.01
I-5000	128.097	635.588	8.229	7.275	9.21	0.008

Table 20: Detailed results of INVIT-VRP-50 [PUPO] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.797	10.329	4.584	0.572	9.606	0.016
U-500	7.909	75.738	9.828	7.848	12.912	0.011
U-5000	128.458	639.748	7.256	6.153	9.235	0.012
C-50	0.789	9.008	4.841	0.553	15.779	0.025
C-500	7.947	60.189	8.926	7.654	11.551	0.009
C-5000	127.466	608.262	7.072	6.281	7.985	0.007
E-50	0.795	9.233	4.943	0.531	10.928	0.019
E-500	8.012	57.44	9.819	7.607	14.179	0.014
E-5000	132.614	370.181	7.705	6.471	9.107	0.012
I-50	0.846	9.819	4.872	0.481	14.009	0.02
I-500	8.1	69.219	9.49	7.007	12.117	0.012
I-5000	130.251	627.491	6.873	6.444	7.233	0.003

Table 21: Detailed results of INVIT-VRP-100 [Vanilla] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.778	10.319	4.75	0.235	10.179	0.017
U-500	7.906	75.658	9.672	8.112	11.936	0.009
U-5000	130.972	643.717	7.896	6.829	9.479	0.01
C-50	0.77	8.989	4.639	1.182	15.259	0.024
C-500	7.903	60.362	9.223	7.537	11.981	0.009
C-5000	126.556	611.67	7.655	7.229	7.876	0.003
E-50	0.777	9.227	4.878	0.587	11.05	0.019
E-500	7.915	57.486	9.82	7.829	12.352	0.01
E-5000	127.763	372.908	8.472	7.441	9.643	0.01
I-50	0.779	9.81	4.782	0.707	13.409	0.021
I-500	7.941	69.15	9.336	7.609	11.371	0.009
I-5000	127.558	630.358	7.353	6.908	7.836	0.004

Table 22: Detailed results of INVIT-VRP-100 [PUPO] on random dataset.

Instance	Mean Time (s)	Mean Length	Mean Gap (%)	Min Gap (%)	Max Gap (%)	Std
U-50	0.776	10.365	5.055	0.369	9.866	0.017
U-500	7.796	74.755	8.366	7.137	10.577	0.008
U-5000	127.661	625.141	4.723	4.172	5.306	0.005
C-50	0.775	9.017	4.973	0.651	17.35	0.026
C-500	7.754	59.417	7.511	5.883	9.331	0.007
C-5000	127.345	593.798	4.511	4.173	4.826	0.003
E-50	0.783	9.27	5.365	0.804	12.195	0.021
E-500	7.903	56.784	8.517	7.271	12.073	0.01
E-5000	127.504	361.641	5.165	4.653	5.629	0.004
I-50	0.793	9.856	5.283	0.473	13.671	0.021
I-500	7.891	68.487	8.276	6.14	10.157	0.008
I-5000	126.872	614.135	4.584	4.223	4.953	0.003