Probabilistic Greedy Behaviour of the Equivariant Quantum Circuit

Jonathan TEO[1], LEE Xin Wei[1], and LAU Hoong Chuin[1]

Singapore Management University, 80 Stamford Road, Singapore 178902
`{jrteo.2022, xwlee, hclau}@smu.edu.sg`

Keywords: Equivariant Quantum Circuit · Quantum Reinforcement Learning · Neural Combinatorial Optimization · Model Interpretability

1 Introduction

In this paper, we present experimental evidence that the *Equivariant Quantum Circuit* (EQC) [9] to solve the TSP performs *no better* than a classical *Probabilistic Nearest Neighbour* (PNN) Algorithm. The original EQC work reported near-optimal performance on TSP instances ≤ 20 nodes [9]. Despite strong interest since its 2023 publication, we find that the EQC's performance at Depth 1 is statistically indistinguishable from a PNN baseline in terms of optimality gaps on both TSP instances of uniform node locations, as well as TSPLIB instances between 5 and 55 nodes. On a set of handcrafted adversarial TSP instances designed to expose local decision making, the tours produced by the EQC at depths 1 to 4 are largely similar to the tours produced by the PNN baseline. Lastly, we evaluated a classical model *Structure2Vec* (S2V) [2], and found this model consistently performs better than the PNN baseline and the EQC. Scripts, result files and data to reproduce the figures are available in our GitHub repository archived at Zenodo [11]. To preserve novelty of ongoing work, the GitHub repository will be made public upon acceptance to QTML 2025. Training code will be released upon full publication.

Neural Combinatorial Optimization (NCO) uses *Deep Reinforcement Learning* (DRL) to approximate solutions to Combinatorial Optimization problems [1,12]. The EQC was the first application of Quantum RL to NCO. The design of the EQC creates a similar encoding for isomorphic graphs [9], which is highly similar to the design properties of classical NCO algorithms that respect graph structures through message passing and attention [2,5,6].

The EQC offered a significant parameter reduction from its classical counterparts. In [10], we showed that at depth 1, the γ parameter effectively controls the number of nearest neighbours explored by the agent. This has motivated us to analyze if the solutions produced by EQC is indeed intelligent. More precisely, we analyze whether the EQC is in fact any better than a PNN baseline, which constructs a solution based on exploring an unvisited node with probability relative to its distance from the current last node.

We conducted three experiments to evaluate our hypothesis, and the results are as follows. Firstly, on a dataset of TSP instances with uniformly generated node locations (of size 5 to 55), the best EQC parameter settings found at Depth 1 produces a similar optimality gaps to the PNN Baseline (max difference $\Delta_{\max} = 0.043$, average difference $\bar{\Delta} = 0.023$). Secondly, on a dataset of TSPLIB instances (of size 16 to 55), a paired T-test of the individual gaps of the 15 TSPLIB instances show that the difference in gaps of the EQC and PNN is statistically insignificant (T-test $p > 0.05$). Lastly, on a dataset of 10 Adversarial TSP instances of sizes 10 to 14, we show that the tours produced by the EQC at Depths 1 to 4 closely follow the PNN Baseline, and the difference in gaps between the EQC and the PNN is statistically insignificant at depths 1 to 4 ($p > 0.05$).

2 Methodology

Probabilistic Nearest Neighbour (PNN) Baseline Given the current node t and a set of unexplored nodes $\mathcal{A}$, an unexplored node $a \in \mathcal{A}$ is chosen with probability

$$\mathbb{P}(a|t) = \frac{\exp(-d_{ta}k)}{\sum_{v \in \mathcal{A}} \exp(-d_{tv}k)} \tag{1}$$

, where d_{ij} denotes the Euclidean Distance between nodes i and j, and we pick $k = 150$ as performance saturates beyond $k \geq 120$ (see Figure 1). We report the best tour after generating T tours in this fashion.

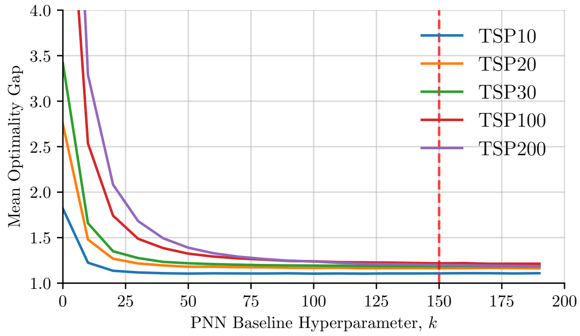


Fig. 1: Ablation on the value of k for the PNN baseline

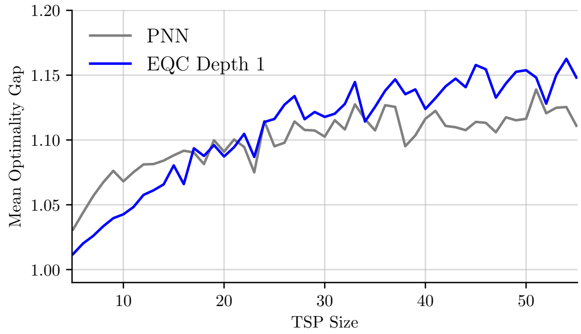


Fig. 2: Mean Optimality Gap with respect to TSP Size

Note. The plot for PNN shows the best gap obtained in $T = 30$ runs.

15 instances (sizes 16 to 55)	
EQC Gap	$\mu = 1.128,\ s = 0.049$
PNN Gap	$\mu = 1.105,\ s = 0.055$
$p = 0.25$	95 % CI $[-0.019,\ 0.065]$

(a) Comparison between Depth 1 EQC and PNN Baseline

38 instances (sizes 51 to 320)	
S2V Gap	$\mu = 1.045,\ \sigma = 0.032$
PNN Gap	$\mu = 1.149,\ \sigma = 0.057$
$p = 3.9 \times 10^{-14}$	95 % CI $[-0.122,\ -0.086]$

(b) Comparison between PNN Baseline and *Structure2Vec* (S2V)

Table 1: Experimental Results for TSPLIB instances

Note. The gap for PNN shows the best gap obtained in $T = 500$ runs. For EQC, we evaluate using γ^* found on the size of the TSPLIB instance. For S2V, we use the reported gaps on TSPLIB in [2]. p refers to a paired T-test of instance gaps. EQC and PNN tour lengths can be found in our supplementary repository [11].

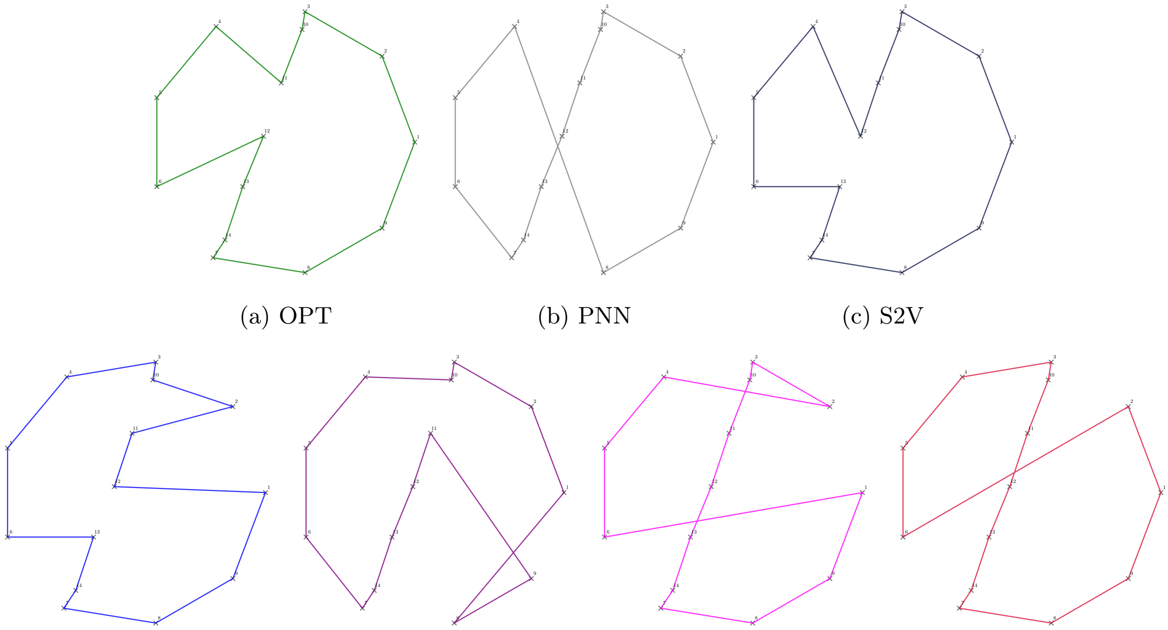
EQC Training We adopt different strategies for Depth 1 and Depths 2 to 4.

- **Depth 1**: For each TSP size between 5 and 55, we fix $\beta = 1.1$ and grid-search $\gamma \in [0, 1.7]$. We keep γ giving the lowest mean optimality gap. This is due to the size-invariant properties at Depth 1 of the EQC, discussed in our companion submission [10], where we showed that the maximum difference in optimality gaps achieved by RL and this approach was $\Delta_{\max} = 0.018$, and average difference $\bar{\Delta} = 0.008$. For sizes larger than 17, the quantum simulation becomes memory intensive. As such, we use the analytical expression for the Q-value provided by the authors in [9].
- **Depth 2-4**: We train the EQC under the *Deep Q Learning* (DQN) RL framework [7,13]. Our train set consists of 500 TSP instances with uniformly sampled locations, and validating every 10 episodes on 100 TSP instances. We keep the parameters with minimal gap. We used the discount factor we used is $g = 1$ (no discounting).

The TSP instances we generate have coordinates in a unit square ($x_i \in [0, 1]^2$ for all i). For TSP instances of size ≤ 14, we take gaps with respect to the Held-Karp optima [4], otherwise we take the best of 30 simulated annealing runs [3]. For TSPLIB instances, we use the optimal solutions found in [8]. *Mean Optimality Gap* (Mean Gap) is computed as the average of (Length of tour)/(Optimal Tour length).

3 Results

Uniform Random TSP Instances (EQC Depth 1 vs PNN): Figure 2 shows that the Mean Gaps of the Depth 1 EQC and PNN Baseline are highly similar ($\bar{\Delta} = 0.0208, \Delta_{\max} = 0.0439$). The best γ for Depth 1 EQC can be interpreted as the best tradeoff between making locally optimal decisions (i.e., selecting the nearest unvisited node) and occasional sub-optimal exploration [10]. As such, this experiment tests whether the EQC is able to make more globally informed decisions than a simple greedy heuristic. Given the small differences in mean gap, we conclude a Depth 1 EQC exhibits behavior is indistinguishable from a PNN Baseline.



(a) OPT (b) PNN (c) S2V

(d) EQC Depth 1 (e) EQC Depth 2 (f) EQC Depth 3 (g) EQC Depth 4

Fig. 3: Adversarial TSP Instance `circle14`

	EQC D1	**EQC D2**	**EQC D3**	**EQC D4**	**S2V**	**PNN**
Mean Gap	1.081	1.088	1.094	1.086	1.005	1.080
Paired T test with PNN, p	0.816	0.0699	0.114	0.0953	0.00423	-
95% CI	[-0.0107, 0.0133]	$[-8.21 \times 10^{-4}$, 0.0173]	$[-4.04 \times 10^{-3}$, 0.0313]	$[1.49 \times 10^{-3}$, 0.0154]	[-0.113, -0.0302]	-

Table 2: Results on Adversarial TSP Instances, designed to expose local decision making

Note. For Figure 3 and Table 2, the PNN gap and tour is the best tour produced in $T = 500$ runs. For EQC (all depths), we use optimal parameters identified for that TSP size. [11]. This dataset consists of 10 (handcrafted) adversarial TSP instances (size between 10 and 14). Figure 3 is one adversarial TSP instance. For more visualizations and S2V Baseline Details, please refer to our Supplementary Repository [11].

TSPLIB Instances (EQC Depth 1 vs PNN): Table 1a summarizes the comparison of the Depth 1 EQC and PNN on TSPLIB instances. Table 1 shows that PNN attains a lower mean gap than EQC, but this difference is statistically insignificant ($p = 0.24$). This further supports our observation that the behaviour of a depth 1 EQC is similar to a PNN. In contrast, Table 1b shows that the classical learning-based method *Structure2Vec* (S2V) significantly outperforms the PNN baseline ($p < 0.01$). This performance gap suggests that, despite its circuit-based inductive bias, the EQC still lags behind classical NCO models in capturing problem structure - leaving substantial room for improvement.

Adversarial TSP Instances (EQC Depths 1 to 4, PNN, S2V): Table 2 shows that the EQC at all depths produces tours that are statistically indistinguishable from the PNN Baseline (at 5% significance level), while S2V produces a near-optimal tour. Figure 3 shows the EQC (all depths) and PNN produces tours with detour-prone behaviour, as a result of its bias toward its nearer (unexplored) neighbors. S2V, in contrast, demonstrates stronger global decision making in its tour construction. Upon further investigation, we observe that the best EQC models from (Depth 2 to 4) gives shorter edges generally larger Q values than longer edges, similar to a Depth 1 EQC that we analyzed in [10]. This partially explains the tours produced at Depths 2 to 4. These findings raise the question if EQC's inductive bias limits its ability to generalize beyond local heuristics, or if optimal EQC parameters is highly sensitive to instance generation procedure. We leave a more thorough explanation of these possibilities to a future work.

References

1. Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. CoRR **abs/1611.09940** (2016), http://arxiv.org/abs/1611.09940
2. Dai, H., Khalil, E.B., Zhang, Y., Dilkina, B., Song, L.: Learning combinatorial optimization algorithms over graphs (2018), https://arxiv.org/abs/1704.01665
3. Dreo, J., Siarry, P., Petrowski, A., Taillard, E.: Metaheuristics for hard optimization : methods and case studies. Springer-Verlag (2006), https://hal.science/hal-01341683
4. Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied Mathematics **10**(1), 196–210 (1962), http://www.jstor.org/stable/2098806
5. Kaempfer, Y., Wolf, L.: Learning the multiple traveling salesmen problem with permutation invariant pooling networks (2019), https://arxiv.org/abs/1803.09621
6. Kool, W., van Hoof, H., Welling, M.: Attention, learn to solve routing problems! (2019), https://arxiv.org/abs/1803.08475
7. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (Feb 2015). https://doi.org/10.1038/nature14236
8. Reinelt, G.: TSPLIB—a traveling salesman problem library. ORSA Journal on Computing **3**(4), 376–384 (1991). https://doi.org/10.1287/ijoc.3.4.376
9. Skolik, A., Cattelan, M., Yarkoni, S., Bäck, T., Dunjko, V.: Equivariant quantum circuits for learning on weighted graphs. npj Quantum Information **9**, 1–15 (2022). https://doi.org/10.1038/s41534-023-00710-y
10. Teo, J., Lee, X., Lau, H.C.: Size-invariant properties at depth 1 of the equivariant quantum circuit (2025), submitted to QTML 2025 (Extended Abstract)
11. Teo, J., Lee, X., Lau, H.C.: Supplementary repository snapshot for review purposes: probabilistic-greedy-eqc (Jun 2025). https://doi.org/10.5281/zenodo.15616831
12. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks (2017), https://arxiv.org/abs/1506.03134
13. Watkins, C.: Learning from Delayed Rewards. Phd thesis, University of Cambridge, England (1989)