Tensor and gadget reinforcement learning for improved, hardware-aware quantum architecture search

Akash Kundu

Delft University of Technology, The Netherlands, kunduaku339@gmail.com

Introduction

Variational quantum algorithms (VQAs) [1] are a leading paradigm for exploiting noisy intermediate-scale quantum (NISQ) [2, 3] devices and promising building blocks beyond NISQ [4], with applications in chemistry, combinatorial optimization, and many-body physics. Their performance hinges on parameterized quantum circuits (PQCs) that are both expressive and hardware-executable, yet this design is difficult. Problem-inspired ansätze [5, 6, 7] encode structure and symmetries but are rarely hardware friendly, while hardware-efficient ansätze (HEA) [8] align with native connectivity and gates but often become hard to train as system size and noise grow. Moreover, according to ref. [9], QPU noise can drive PQCs into noise-induced barren plateaus, causing gradients to vanish and training to effectively stall or fail to converge.

Table 1: Summary of selected quantum architecture search (QAS) methods.

Method	**Application**	**Maximum qubit**	**Noise inclusion**
RL-VQSD [10]	Quantum state diagonalization	4 qubits	Noiseless
Differential QAS [11]	Heisenberg model	5 qubits	Simulated
Vanilla RL-QAS [12]	Quantum chemistry	6 qubits	Simulated
Sim. annealing QAS [13]	Quantum circuit regeneration	7 qubits	Simulated
quantumDARTS [14], CRLQAS [15]	Quantum chemistry	8 qubits	Simulated
GRL [16]	TFIM & quantum chemistry	10 qubits	Simulated & **QPU**
TensorRL-QAS [17]	Quantum chemistry, Heisenberg & TFIM	**20 qubits**	Simulated

Reinforcement learning (RL) [12, 18, 19] has thus emerged as a natural tool for automating PQC design, also know as quantum architecture search (QAS), by casting circuit construction as a sequential decision problem and searching for architectures that balance expressivity and hardware efficiency. As summarized in Table 1, existing QAS methods still suffer from limited scalability and hardware awareness, typically stopping at modest qubit numbers and relying on abstract gate sets whose transpilation inflates depth and CNOT counts and hurts performance on noisy hardware. This work unifies two complementary frameworks aimed at closing these gaps: (i) **TensorRL-QAS** [17], which combines tensor-network warm starts with RL to alleviate classical-simulation and episode-length bottlenecks, and (ii) **Gadget reinforcement learning (GRL)** [16], which augments RL with learned composite gates ("gadgets") defined directly in a hardware-native gate set to mitigate transpilation-induced losses; together, they target both *scalability* and *hardware compatibility*, reaching 20-qubit simulations and 10-qubit demonstrations under realistic hardware constraints on quantum chemistry and many-body benchmarks.

Towards scalability with TensorRL-QAS

Linked paper: TensorRL-QAS (https://openreview.net/forum?id=0okFLZvtKs).
RL-based QAS typically starts from an empty circuit and lets an agent sequentially append gates, leading to long episodes and expensive statevector simulations whose cost grows sharply with qubit count and depth. In standard baselines (Figure 7 in ref. [17]), a 12-qubit circuit at depth ~ 400 can take on the order of 10 seconds per evaluation, so 10^4 training episodes imply $\gtrsim 10^5$ seconds of simulation, making scaling beyond ~ 8 qubits practically infeasible.

Figure 1: TensorRL-QAS performance on 8-qubit H_2O. On (left) is the number of function evaluations and (right) per-episode runtime versus RL baselines.

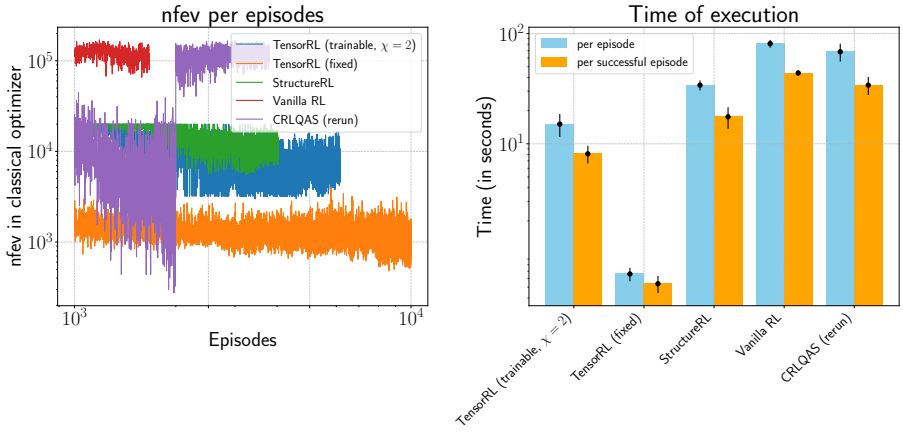


Table 2: TensorRL vs baselines for 12-qubit LiH ground-state preparation.

Method	Error	Depth	CNOT	ROT
TensorRL (train.)	$\mathbf{1.0 \times 10^{-2}}$	31	37	203
TensorRL (fixed)	2.4×10^{-2}	**15**	**30**	**9**
CRLQAS [15]	2.4×10^{-2}	32	68	23
Vanilla RL [12]	2.2×10^{-2}	140	321	94
RA-QAS [10]	2.3×10^{-2}	165	364	86
SA-QAS [13]	2.5×10^{-2}	178	390	88
TN-VQE[a]	8.6×10^{-2}	33	81	29

[a]Fixed tensor-network initialization followed by a standard hardware-efficient ansatz.

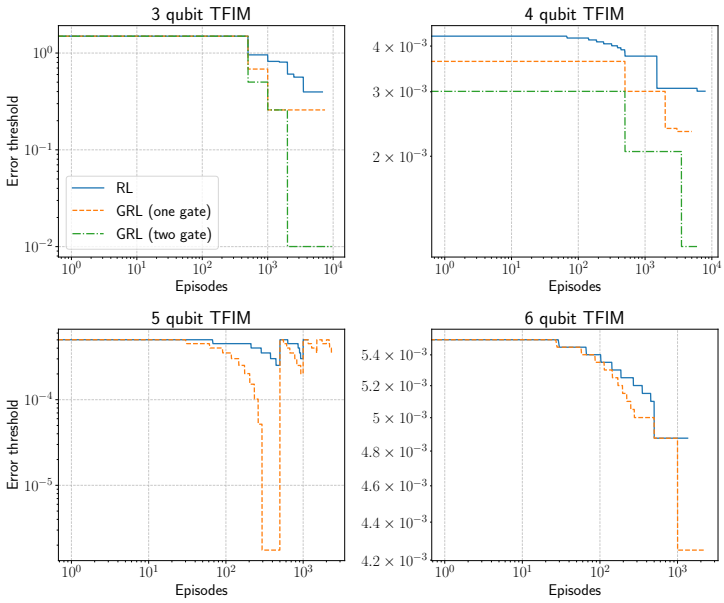
TensorRL-QAS mitigates this bottleneck by warm-starting the RL agent from a tensor-network approximation of the target ground state. Given a Hamiltonian H, DMRG first computes an MPS approximation $|\psi_{\mathrm{MPS}}\rangle \approx \arg\min_{|\psi\rangle \in \mathcal{M}_\chi} \langle\psi| H |\psi\rangle$, which is then compiled into a shallow brickwork PQC $U(\boldsymbol{\theta})$ by minimizing $\mathcal{L}(\boldsymbol{\theta}) = 1 - \left|\langle \psi_{\mathrm{MPS}} | U(\boldsymbol{\theta}) 0^{\otimes N} \rangle\right|^2$ using Riemannian optimization [20, 21] on the Stiefel manifold. RL-based QAS then learns an additional unitary $V(\boldsymbol{\phi})$ on top of the warm start, refining the combined ansatz $V(\boldsymbol{\phi})U(\boldsymbol{\theta})$ to minimize the variational energy $E(\boldsymbol{\theta}, \boldsymbol{\phi}) = \langle 0| U^\dagger(\boldsymbol{\theta}) V^\dagger(\boldsymbol{\phi}) H V(\boldsymbol{\phi}) U(\boldsymbol{\theta}) |0\rangle$, rather than exploring from an empty circuit. We study two variants: *TensorRL (trainable TN-init)*, where $U(\boldsymbol{\theta})$ is trainable by the agent, and *TensorRL (fixed TN-init)*, where the warm start is fixed and only $V(\boldsymbol{\phi})$ is optimized.

On molecular ground-state problems (6–12 qubits), TensorRL-QAS yields more compact circuits than strong QAS baselines while keeping energies in the 10^{-2} range or better (Table 2). For 12-qubit LiH, TensorRL (trainable TN-init) achieves the best error 1.0×10^{-2}, and TensorRL (fixed TN-init) attains a much smaller circuit (depth 15, 30 CNOTs, 9 single-qubit rotations) than CRLQAS, vanilla RL, RA-QAS, SA-QAS, and TN-VQE. Figure 1 further shows that TensorRL (fixed) reduces classical function evaluations by up to 100× and per-episode runtime by about 98%, and transverse-field Ising experiments extend this advantage up to 17–20 qubits, indicating a practical route to larger-scale RL-QAS via tensor-network initialization.

Table 3: Average and minimum TFIM ground-state energies for GRL and RL on IBM Heron fake backends.

2-qubit TFIM					**3-qubit TFIM**				
Backend name	**GRL**		**RL**		Backend name	**GRL**		**RL**	
	Avg.	Min.	Avg.	Min.		Avg.	Min.	Avg.	Min.
fake_torino	−2.188	**−2.213**	−2.164	−2.1992	fake_torino	−3.309	**−3.366**	−3.287	−3.351
fake_kawasaki	−2.162	**−2.196**	−2.123	−2.1592	fake_kawasaki	−3.370	**−3.397**	−3.235	−3.319
fake_quebec	−2.118	**−2.145**	−2.084	−2.1597	fake_quebec	−3.318	**−3.379**	−3.266	−3.318

Figure 2: Error threshold versus episodes for RL and GRL on 3-6 qubit TFIM instances.



QPU-friendly PQC construction by GRL

Linked paper GRL (`https://www.nature.com/articles/s42005-025-02475-6`).

Most existing RL-based QAS methods operate in an abstract universal gate set such as $\{R_X, R_Y, R_Z, \mathrm{CX}\}$ and only later transpile to hardware-native gates, which inflates depth and CX count and degrades fidelity on real devices. Gadget reinforcement learning (GRL) [16] addresses this by combining RL with program synthesis to expand the action space with learned composite gates ("gadgets") that are defined directly in a hardware-native gate set.

For a Hamiltonian H and native gate set $\mathcal{G}_{\mathrm{nat}} = \{R_Z, \mathrm{SX}, X, \mathrm{CZ}\}$, the agent learns PQCs $U(\boldsymbol{\theta}) \in \langle \mathcal{G}_{\mathrm{nat}} \rangle$ that minimize the variational energy $E(\boldsymbol{\theta}) = \langle 0 | U^\dagger(\boldsymbol{\theta}) H U(\boldsymbol{\theta}) | 0 \rangle$, with rewards whenever $E(\boldsymbol{\theta})$ falls below a threshold ε. A program-synthesis module then analyzes the top-k circuits $\{U_i\}$ and extracts recurrent subsequences $G_j = g_{j,1} g_{j,2} \cdots g_{j,m_j}$, where $g_{j,\ell} \in \mathcal{G}_{\mathrm{nat}}$, which are promoted to new actions, yielding an augmented gate set $\mathcal{G}_{\mathrm{aug}} = \mathcal{G}_{\mathrm{nat}} \cup \{G_j\}$. Subsequent RL stages operate directly in $\mathcal{G}_{\mathrm{aug}}$, so the agent reasons in terms of higher-level, hardware-native building blocks rather than only elementary gates.

Figure 2 shows that gadgets such as $\mathrm{RZ}_i\mathrm{CZ}_{ij}$ and $X_i\sqrt{X}_j$, learned on 2-qubit TFIM, transfer effectively up to 6-qubit, with GRL reaching lower error thresholds in fewer episodes than standard RL on 3–6-qubit TFIM. This advantage carries over to IBM Heron-like backends (Table 3), where GRL attains more negative ground-state energies than RL for 2- and 3-qubit TFIM while keeping circuits compact, and for the 3-qubit case matches the RL energy at machine precision using roughly half as many single-qubit gates after transpilation, highlighting improved accuracy, lower transpilation overhead, and better hardware compatibility. Moreover, Figure 1 of Ref. [16] shows that on 3-qubit TFIM, GRL consistently outperforms non-RL baselines (QNEAT [22], hybrid evolutionary algorithms [23], constant-depth TFIM ansätze [24], and HEA [8]) in both error and gate counts.

Remarks

TensorRL-QAS and GRL jointly address the two main bottlenecks of RL-QAS. Scalability and hardware-awareness, enabling compact circuits up to 20-qubit in simulation and 10-qubit on hardware-inspired backends. Tensor-network warm starts drastically reduce the search space cost, while gadget-based action spaces preserve performance after transpilation with hardware-native gates. These ingredients provide a practical path toward scalable, hardware-compatible RL-driven circuit design for NISQ and beyond.

References

[1] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[2] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[3] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[4] Zoltán Zimborás, Bálint Koczor, Zoë Holmes, Elsi-Mari Borrelli, András Gilyén, Hsin-Yuan Huang, Zhenyu Cai, Antonio Acín, Leandro Aolita, Leonardo Banchi, et al. Myths around quantum computation before full fault tolerance: What no-go theorems rule out and what they don't. *arXiv preprint arXiv:2501.05694*, 2025.

[5] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology*, 4(1):014008, 2018.

[6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[7] Dmitry A Fedorov, Yuri Alexeev, Stephen K Gray, and Matthew Otten. Unitary selective coupled-cluster method. *Quantum*, 6:703, 2022.

[8] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.

[9] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):6961, 2021.

[10] Akash Kundu, Przemysław Bedełek, Mateusz Ostaszewski, Onur Danaci, Yash J Patel, Vedran Dunjko, and Jarosław A Miszczak. Enhancing variational quantum state diagonalization using reinforcement learning techniques. *New Journal of Physics*, 26(1):013034, 2024.

[11] Shi-Xin Zhang, Chang-Yu Hsieh, Shengyu Zhang, and Hong Yao. Differentiable quantum architecture search. *Quantum Science and Technology*, 7(4):045023, 2022.

[12] Mateusz Ostaszewski, Lea M Trenkwalder, Wojciech Masarczyk, Eleanor Scerri, and Vedran Dunjko. Reinforcement learning for optimization of variational quantum circuit architectures. *Advances in neural information processing systems*, 34:18182–18194, 2021.

[13] Xudong Lu, Kaisen Pan, Ge Yan, Jiaming Shan, Wenjie Wu, and Junchi Yan. Qas-bench: rethinking quantum architecture search and a benchmark. In *International conference on machine learning*, pages 22880–22898. PMLR, 2023.

[14] Wenjie Wu, Ge Yan, Xudong Lu, Kaisen Pan, and Junchi Yan. Quantumdarts: differentiable quantum architecture search for variational quantum algorithms. In *International conference on machine learning*, pages 37745–37764. PMLR, 2023.

[15] Yash J. Patel, Akash Kundu, Mateusz Ostaszewski, Xavier Bonet-Monroig, Vedran Dunjko, and Onur Danaci. Curriculum reinforcement learning for quantum architecture search under hardware errors. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] Akash Kundu and Leopoldo Sarra. Reinforcement learning with learned gadgets to tackle hard quantum problems on real hardware. *Communications Physics*, 2026.

[17] Akash Kundu and Stefano Mangini. Tensorrl-qas: Reinforcement learning with tensor networks for improved quantum architecture search. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[18] Thomas Fösel, Murphy Yuezhen Niu, Florian Marquardt, and Li Li. Quantum circuit optimization with deep reinforcement learning. *arXiv preprint arXiv:2103.07585*, 2021.

[19] En-Jui Kuo, Yao-Lung L Fang, and Samuel Yen-Chi Chen. Quantum architecture search via deep reinforcement learning. *arXiv preprint arXiv:2104.07715*, 2021.

[20] Hiroyuki Sato. *Riemannian optimization and its applications*, volume 670. Springer, 2021.

[21] Isabel Nha Minh Le, Shuo Sun, and Christian B Mendl. Riemannian quantum circuit optimization based on matrix product operators. *arXiv preprint arXiv:2501.08872*, 2025.

[22] Alessandro Giovagnoli, Volker Tresp, Yunpu Ma, and Matthias Schubert. Qneat: Natural evolution of variational quantum circuit architecture. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 647–650, 2023.

[23] Leo Sünkel, Philipp Altmann, Michael Kölle, Gerhard Stenzel, Thomas Gabor, and Claudia Linnhoff-Popien. Quantum circuit construction and optimization through hybrid evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 934–942, 2025.

[24] Daan Camps, Efekan Kokcu, Lindsay Bassman Oftelie, Wibe A De Jong, Alexander F Kemper, and Roel Van Beeumen. An algebraic quantum circuit compression algorithm for hamiltonian simulation. *SIAM Journal on Matrix Analysis and Applications*, 43(3):1084–1108, 2022.