GENERATIVE REPLAY MITIGATES SAMPLE STARVATION IN QUANTUM ARCHITECTURE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) can automate quantum architecture search (QAS), but its scalability breaks down when useful circuit trajectories become rare in a rapidly expanding search space. We show that the probability of encountering an ϵ-accurate low energy circuit through passive exploration decreases super exponentially with qubit count, making uniform and prioritized experience replay fundamentally insufficient for large-scale QAS. We introduce **GenQAS**, a tensor network-guided RL framework that combines a fixed matrix product state warm start with prioritized generative replay. In this framework, a learned local transition model generates synthetic circuit transitions on demand from real state-action interactions and mixes them with real experience during Double Deep Q-Network updates. Across chemical Hamiltonian benchmarks from 6 to 12 qubits, GenQAS improves fixed-budget success probability and identifies more compact circuits at competitive energy error, improving final success probability by up to 7.0$\times$ over passive replay at 12 qubits. On a 15-qubit transverse field Ising model, GenQAS reaches 21% success probability, a 75% relative gain over uniform TensorRL-QAS (12%) and a 40% relative gain over prioritized TensorRL-QAS (15%). To test transfer, we move a replay buffer trained in a noiseless 6-qubit BEH_2 environment into a noisy version of the same task, comparing a transferred uniform buffer against a transferred GenQAS generative buffer. The uniform transfer reduces steps to chemical accuracy by only 30.1%, whereas the GenQAS generative replay transfer reduces steps by 92.7%, a substantially larger improvement under identical noise conditions. These results show that generative replay mitigates sample starvation in QAS and, critically, that the learned generative buffer carries far more transferable structure than passive replay, supporting more resource-efficient and robust circuit discovery. The code is available at `https://anonymous.4open.science/r/GenQAS/README.md`.

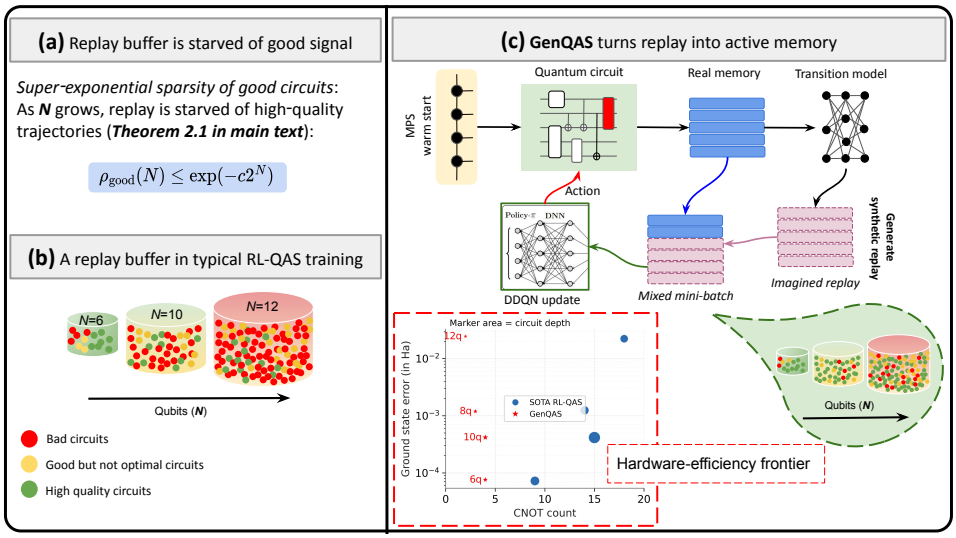
1 INTRODUCTION

The performance of near term and early fault tolerant quantum algorithms is limited by qubit count, imperfect gate execution, and hardware noise Preskill (2018); Eisert & Preskill (2025); Katabarwa et al. (2024). Achievable performance depends not only on qubit count but also on error rates, connectivity, and reliably executable circuit depth Blume-Kohout & Young (2020); Cross et al. (2019). This is especially important for variational quantum algorithms (VQAs) Cerezo et al. (2021), in which a parameterized quantum circuit (PQC) $U(\boldsymbol{\theta}, \mathcal{C})$, defined by a discrete architecture $\mathcal{C}$ and continuous parameters $\boldsymbol{\theta}$, prepares the state $|\psi(\boldsymbol{\theta}, \mathcal{C})\rangle = U(\boldsymbol{\theta}, \mathcal{C})\,|0\rangle^{\otimes N}$, optimized by minimizing $C(\boldsymbol{\theta}, \mathcal{C}) = \langle\psi(\boldsymbol{\theta}, \mathcal{C})|\, H\, |\psi(\boldsymbol{\theta}, \mathcal{C})\rangle$ using classical optimizers such as those in SciPy Virtanen et al. (2020). The structure of a PQC strongly influences the accuracy, trainability, and hardware cost of a VQA. Problem inspired ansätze encode prior knowledge of the target problem, for example through Trotterized evolution Wecker et al. (2015) or physically motivated excitation operators Anand et al. (2022); Grimsley et al. (2019), whereas hardware efficient architectures use layers of native operations arranged to match device connectivity Peruzzo et al. (2014); Kandala et al. (2017).

Fixed PQC architectures may be insufficiently expressive or prone to local minima. Quantum architecture search (QAS) addresses this by automatically constructing PQC architectures from a prescribed gate set Zhang et al. (2022); Du et al. (2022); Kuo et al. (2021), letting circuit structure adapt to the target task and hardware. Reinforcement learning (RL) provides a natural formulation for QAS, in which an agent sequentially constructs a circuit by selecting gate types and locations Ostaszewski et al. (2021); Kuo et al. (2021); Fösel et al. (2021); Moro et al. (2021); Zhang et al. (2020),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Figure 1: **GenQAS converts passive replay into active experience generation.** **(a)** Under weakly informed exploration, high quality circuit trajectories become increasingly sparse as the system size grows. **(b)** Uniform and prioritized replay can only resample previously observed transitions, so the density of useful experience declines as the search space expands. **(c)** GenQAS starts from an MPS derived circuit, retains real transitions, learns a local transition model, and generates synthetic transitions on the fly. Real and generated samples are mixed during double deep Q-network updates, supporting the search for low error circuits with fewer entangling gates.



and has separately been used for related quantum control and circuit optimization tasks Sivak et al. (2026); Ruiz et al. (2025).

This flexibility creates a rapidly expanding decision space. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ denote the available single and two qubit gate families, and let $\mathcal{E}_N$ denote the set of hardware allowed two qubit couplings on an N qubit device. The number of available actions at each construction step is

$$|\mathcal{A}_N| = \underbrace{|\mathcal{G}_1|N}_{\text{1-qubit}} + \underbrace{|\mathcal{G}_2||\mathcal{E}_N|}_{\text{2-qubit}} + 1, \tag{1}$$

where the final term in Eq. 1 denotes termination and $|\mathcal{E}_N| = N(N-1)/2$ for a fully connected device. A construction process of maximum depth T can therefore encounter up to $|\mathcal{A}_N|^T$ gate sequences, while the Hilbert space dimension grows as 2^N. Each candidate architecture further requires continuous parameter optimization and repeated objective estimation on a quantum simulator or processor, so RL-QAS must learn from a limited number of costly interactions within a combinatorially expanding search space. For overviews of RL based quantum circuit optimization, see refs. Kundu et al. (2026); Bukov & Marquardt (2026). A detailed discussion with recent scaling focused approaches, including curriculum learning Patel et al. (2024), training free proxy metrics He et al. (2024), and one shot parameter sharing Wang et al. (2022), is given in Section 2.

Closest to our work, TensorRL-QAS Kundu & Mangini (2026) uses a matrix product state (MPS), obtained via density matrix renormalization group calculations, to construct a problem informed initial circuit that is subsequently refined by a Double DQN agent. This warm start places the agent in a more relevant region of circuit space, but after initialization the agent still learns only from transitions explicitly collected through environment interaction. Experience replay improves sample efficiency by reusing these interactions: uniform replay samples stored transitions without regard to learning value, while prioritized experience replay (PER) resamples transitions with larger temporal difference error Mnih et al. (2013); Schaul et al. (2015). Both remain passive, however: they can change how frequently an observed transition is reused, but cannot create informative experience that was never collected. As the search space expands, useful trajectories constitute an increasingly small fraction of the agent's interaction history, and the learning signal becomes diluted by low value alternatives. We refer to this decline in the density of informative experience as *sample starvation*.

We formalize this limitation for weakly informed exploration over sufficiently expressive circuit ensembles, showing that the measure of near ground state experience decreases exponentially with the

dimension of the accessible Hilbert space. This characterizes the geometry of random exploration rather than assuming every learned policy is Haar distributed, but it exposes a structural limitation of passive replay: reweighting collected transitions cannot alter the probability that informative transitions were encountered in the first place. Generative replay discussed in refs. Sutton (1991); Ha & Schmidhuber (2018); Janner et al. (2019); Feinberg et al. (2018); Wang et al. (2025) instead augments the transitions used in value function updates by learning a model of local transition dynamics from real interactions. Quantum circuit construction is particularly suitable for this approach because the current circuit and selected gate constrain the subsequent state, reward, and termination, allowing a learned model to generate additional training samples around observed state action pairs without a new quantum classical evaluation for every generated transition. The central risk is model bias: synthetic transitions help only when consistent with the real environment, so generative replay must remain anchored in real experience.

Here we introduce *GenQAS*, a tensor network guided RL framework that augments passive replay with on demand synthetic transitions for QAS (Figure 1). GenQAS pairs the fixed MPS initialization and Double DQN backbone of TensorRL-QAS with a learned local transition model. Specifically, given a circuit state and gate action, the model predicts the resulting state, reward, and termination probability, defining synthetic one step transitions generated from real state action seeds and mixed with real transitions during value function updates. The MPS initialization and generative replay provide complementary inductive biases, placing the agent in a physically meaningful low energy region while increasing the training signal extracted from subsequent interactions. Synthetic transitions are generated on demand and never stored as replacements for real experience, so the replay buffer remains grounded in environment interactions while the generation ratio G_r controls the synthetic contribution to each update.

Our contributions are threefold. (1) We show analytically that near ground state circuit trajectories become rapidly sparse with increasing system size, so resampling observed transitions alone cannot increase the support of the observed transition distribution. (2) We show that tensor network warm stater generative replay improves the effective density of informative training samples relative to uniform and prioritized replay (PER), provided the learned transition model remains sufficiently accurate. (3) Our evaluation of GenQAS under matched training budgets on 6 qubit $\mathrm{BeH_2}$, 8, 10, and 12 qubit $\mathrm{H_2O}$, and a 15 qubit transverse field Ising model. At 12 qubits, GenQAS increases final success probability from 12% for the strongest passive baseline to 77%, a $7.0\times$ improvement; at 15 qubits, it improves success probability from 15% with PER to 21%. Across all systems, GenQAS also identifies circuits with competitive energy errors, reduced CNOT counts, and reduced depth. In a noisy 6 qubit $\mathrm{BeH_2}$ transfer experiment, GenQAS reduces search steps to the first chemical accuracy solution from 9053 to 660, and on a 4 qubit Clifford synthesis task without variational parameter optimization, GenQAS exceeds the passive RL baseline under a matched training budget.

2 Related Work

Fixed problem inspired and hardware efficient ansätze face an inherent tension between expressiveness and trainability, motivating quantum architecture search (QAS) over a discrete gate pool Zhang et al. (2022). Non-RL approaches include evolutionary algorithms, which scale poorly with gate set size Rubinstein (2001); sampling based and supernet based methods Grimsley et al. (2019); Du et al. (2022); the differentiable search method quantumDARTS Wu et al. (2023); and training free proxy metric ranking He et al. (2024). A deeper theoretical investigation of quantumDARTS are provided in Kumar et al. (2025). These methods primarily reduce the cost of evaluating or selecting candidate architectures, rather than changing the transition data used to guide the search.

Reinforcement learning casts QAS as sequential circuit construction guided by feedback from the quantum cost function Ostaszewski et al. (2021); Kuo et al. (2021), and several extensions target realistic deployment: tensor based encoding with noise robust optimization Patel et al. (2024), deep RL for gate level circuit optimization via learned tensor decomposition Ruiz et al. (2025), and, more recently, a systematic benchmark across RL agents showing that no single algorithm dominates uniformly Ikhtiarudin et al. (2025). Closest to our setting, a tensor network based warm start places the agent in a physically informed region of circuit space before training Kundu & Mangini (2026). RL based quantum optimization extends well beyond QAS, including particle accelerator beam-line control in high energy physics Baba et al. (2023); Kundu (2025) and fault tolerant state preparation Zen et al. (2025); Olle et al. (2024), suggesting that effective RL for quantum optimization depends not only on the quantum environment, but on how the agent processes and samples the

experience it collects. Replay buffer engineering Kundu & Feld (2026) initiated this direction for QAS, and we continue it here. Across this body of work, however, replay remains passive: each advance improves a different pipeline stage, but none augment the transition data available to the value function once training is underway. We recommend the readers chapter VII of Bukov & Marquardt (2026) and the review paper on RL-QCO Kundu et al. (2026) for an indepth discussion of RL-QAS.

Outside quantum computing, model-based RL has long used a learned dynamics model to generate additional training signal beyond observed transitions, from Dyna style planning Sutton (1991) to world models Ha & Schmidhuber (2018) and model based value expansion Feinberg et al. (2018); Janner et al. (2019). Prioritized generative replay (PGR) Wang et al. (2025) mixes samples from a learned conditional generative model with real experience during value updates, rather than only reweighting collected transitions. GenQAS brings this idea to QAS, where each real transition is unusually costly due to repeated variational optimization and circuit evaluation. It is complementary to, rather than competing with, the RL-QAS advances above: GenQAS adopts the TensorRL-QAS warm start as its backbone and could in principle be paired with any advanced RL for quantum circuit optimization framework as existing RL-QAS rarely modify the replay mechanism itself.

3 GENQAS

The circuit construction space grows rapidly with qubit count, making useful low energy trajectories increasingly rare. Passive replay, whether uniform Mnih et al. (2013) or prioritized Schaul et al. (2015), becomes ineffective in this regime since it can only reweight transitions already collected which is a limitation we term *sample starvation*. GenQAS addresses this with real-anchored generative replay, building on the prioritized generative replay framework Wang et al. (2025): a learned local transition model produces synthetic circuit transitions on demand from real state-action pairs and mixes them into Double DQN updates, increasing the effective training signal without additional quantum-classical evaluations. This is paired with a fixed matrix product state warm start adapted from TensorRL-QAS Kundu & Mangini (2026), compiled into the native gate set and retained as a fixed prefix that places the agent in a physically informed region of circuit space before generative replay is engaged; GenQAS then refines this initialization across all evaluated systems, e.g., reducing the warm start energy error from 1.025 to 1.5×10^{-2} (a 98.5% relative reduction) for the 15-qubit TFIM. Warm start construction, cost, and refinement are detailed in Appendix B.2 and B.6.

3.1 CIRCUIT CONSTRUCTION AND WARM START

We formulate QAS as a Markov decision process in which the state s_t encodes the circuit constructed by the agent at step t, and an action a_t specifies the insertion of a gate at a valid circuit location, using the discrete circuit representation and N qubit, D layer brickwork layout Kundu & Mangini (2026); Leone et al. (2024); Kandala et al. (2017).

GenQAS adopts the fixed TensorRL-QAS initialization: a tensor network derived circuit U_{TN} prepares a fixed warm start state, while the agent constructs a learnable circuit suffix V_t, so the circuit evaluated by the environment is $U_t = V_t U_{\mathrm{TN}}$. The warm start is not encoded in the RL state and its parameters remain fixed throughout training, so the agent observes and searches only over V_t, reducing the observation dimension and separating tensor network state preparation from RL based refinement. The DMRG calculation, MPS representation, variational MPS-to-circuit mapping, and Riemannian optimization procedure are given in Appendix B, along with resource statistics for the warm start circuits in Appendix Table 3.

The discrete action space $\mathcal{A}$ comprises gate placement operations from the fixed gate set $\{\mathtt{RX}, \mathtt{RY}, \mathtt{RZ}, \mathtt{CNOT}\}$ and their valid qubit locations, with invalid actions such as redundant single qubit insertions and repeated entangling gates masked during action selection (full rules in Appendix F).

GenQAS trains this policy with a Double DQN backbone using ϵ-greedy exploration and a per-layer discount schedule that keeps the effective discount fixed across variable circuit depths; the full DDQN formulation, n-step rollout variant for larger systems, and complete hyperparameters are given in Appendix F

3.2 GENERATIVE REPLAY

Beyond real transitions from the QAS environment, GenQAS learns a neural dynamics model from replay memory, following prioritized generative replay Wang et al. (2025). Given (s_t, a_t), the model

predicts

$$(\hat{s}_{t+1}, \hat{r}_t, \hat{d}_t) = f_\psi(s_t, a_t), \tag{2}$$

using separate MLP heads for transition, reward, and termination, trained only on real transitions with

$$\mathcal{L}_{\text{gen}} = 2\mathcal{L}_{\text{trans}} + \mathcal{L}_{\text{rew}} + 0.5\mathcal{L}_{\text{done}}, \tag{3}$$

where $\mathcal{L}_{\text{trans}}$, $\mathcal{L}_{\text{rew}}$ are MSE and $\mathcal{L}_{\text{done}}$ is binary cross entropy (architecture and optimization details in Appendix F).

During DDQN updates, GenQAS draws real transitions from prioritized replay and generates synthetic samples on demand from real state action seeds,

$$\tilde{\tau}_t = (s_t, a_t, \hat{r}_t, \hat{s}_{t+1}, \hat{d}_t). \tag{4}$$

Synthetic transitions are never stored, so the buffer holds only real interactions. For batch size B, the generation ratio G_r sets the synthetic fraction,

$$B_{\text{real}} = \lfloor (1 - G_r)B \rfloor, \qquad B_{\text{syn}} = B - B_{\text{real}}. \tag{5}$$

We evaluate $G_r \in \{0.2, 0.4, 0.6, 0.8\}$. Only real transitions carry priority weights, updated from the DDQN TD error. The generative model itself updates every $T_{\text{gen}} = 10$ DDQN steps using real transitions only, preventing it from training on its own outputs. Replay capacity, batch size, and logging settings are in Appendix F.

3.3 Theoretical guarantees

We establish three results motivating GenQAS: (1) low energy states occupy an exponentially shrinking region of Hilbert space, so passive replay suffers sample starvation as N grows; (2) the MPS warm start is a single-step generation process with a quantifiable fidelity bound; (3) under these conditions, generative replay yields a sample complexity advantage growing with N. Full derivations in Appendix A.

Sample starvation under passive replay. Circuits are discrete, but the unitary they implement is a continuous point in $\mathcal{U}(2^N)$, and unitaries producing an ϵ-accurate ground state are highly concentrated relative to the full group Cerezo et al. (2021).

Theorem 3.1 (Exponential sparsity of high relevance quantum transitions)**.** *For N-qubit parameterized circuits of depth $D = \text{poly}(N)$ acting on $|0\rangle^{\otimes N}$, let H have ground energy E_0, ground state $|\psi_0\rangle$. For $\epsilon > 0$, define $\mathcal{G}_\epsilon = \{U \in \mathcal{U}(2^N) : \langle 0| U^\dagger H U |0\rangle - E_0 < \epsilon\}$. For $\epsilon < \epsilon_0$ (gap-dependent), the Haar probability satisfies*

$$P(\mathcal{G}_\epsilon) \leq \exp\left(-c \cdot 2^N \cdot \log\left(\frac{1}{\epsilon}\right)\right) \tag{6}$$

for some constant $c > 0$.

Without densification, standard replay thus faces a sample complexity barrier $\exp\big(\Omega(2^N)\big)$; generative replay targets $\mathcal{G}_\epsilon$ directly rather than waiting for chance encounters (proof in Appendix A.1). We extend this to general local Hamiltonians of M k-body terms, bounded strength J.

Theorem 3.2 (Sparsity of the low energy manifold)**.** *Let $H = \sum_{j=1}^{M} h_j$ act on N qubits, each h_j on $\leq k$ qubits, $\|h_j\| \leq J$. For $\epsilon < \frac{1}{2}\min\{\|h_j\| : h_j \neq 0\}$, define $\mathcal{G}_\epsilon = \{|\psi\rangle \in \mathbb{CP}^{2^N-1} : \langle\psi| H |\psi\rangle - E_0 < \epsilon\}$. Under the Haar (Fubini-Study) measure,*

$$P(\mathcal{G}_\epsilon) \leq \exp\left(-\frac{2^N}{M\binom{N}{k}} \cdot \frac{\epsilon^2}{8J^2} + O(N)\right), \tag{7}$$

giving $P(\mathcal{G}_\epsilon) \leq \exp\big(-\Omega(2^N/\text{poly}(N))\big)$ for $k = O(1)$, $M = \text{poly}(N)$.

Together, Theorems 3.1–3.2 show low energy states form a super-exponentially small subset of Hilbert space that no passive strategy (uniform, PER, or otherwise) can escape as N grows, only active generation can offset this scaling. We defer the proof in Appendix A.2.

MPS initialization as a bounded generative prior. The TensorRL-QAS warm start is itself guided generation with a quantifiable accuracy guarantee, not a heuristic choice.

Lemma 3.3 (MPS initialization as guided generation)**.** *Let H be an N-qubit Hamiltonian with an area-law ground state, $|\Psi_{\mathrm{MPS}}\rangle$ its DMRG approximation at bond dimension χ. The Riemannian mapping to a brickwork unitary U_{TN} satisfying $U_{\mathrm{TN}}|0\rangle^{\otimes N} \approx |\Psi_{\mathrm{MPS}}\rangle$ is a single-step conditional generation $G_{\mathrm{MPS}}(\tau \mid c = H)$, producing $\tau = (\emptyset, \emptyset, U_{\mathrm{TN}}, \Delta E_{\mathrm{MPS}})$ that initializes the circuit.*

Lemma 3.4 (Synthetic circuit bound)**.** *Let $|\Psi_0\rangle$ be the ground state of a gapped local Hamiltonian H, U_{TN} obtained from $|\Psi_{\mathrm{MPS}}\rangle$ via Riemannian optimization over $\mathcal{U}(4)^{\times m}$. For any $\delta > 0$, there exists $\chi \leq \mathrm{poly}(N, 1/\delta)$ and polynomial optimization steps such that*

$$\frac{1}{2}\left\| U_{\mathrm{TN}}|0\rangle\langle 0| U_{\mathrm{TN}}^{\dagger} - |\Psi_0\rangle\langle\Psi_0| \right\|_1 \leq \delta. \tag{8}$$

Lemmas 3.3–3.4 show the fixed warm start is a provably high-fidelity synthetic transition obtained before any circuit execution, with pretraining cost polynomial in $N, 1/\delta$—a principled basis for treating TensorRL-QAS as an instance of the generative framework motivating GenQAS, with energy expectation as the relevance signal (proofs in Appendices A.3, A.4).

3.3.1 Generative replay accelerates convergence

Theorem 3.5 (Convergence acceleration via synthetic replay)**.** *Let π_{base} train purely from $\mathcal{D}_{\mathrm{real}}$, and π_{PGR} train with prioritized generative replay Wang et al. (2025), where G_ψ is conditioned on $-\Delta E$ and densifies the buffer around warm start s_{TN}. There exists a constant $\kappa > 1$ (gap- and noise-dependent) such that*

$$\mathcal{S}_{\pi_{\mathrm{PGR}}}(\epsilon_c) \leq \frac{1}{\kappa^N}\mathcal{S}_{\pi_{\mathrm{base}}}(\epsilon_c). \tag{9}$$

The episode complexity gap grows exponentially in N, matching the rate at which Theorems 3.1–3.2 predict passive buffers to fail: generative replay is the corrective mechanism for this sparsity barrier (proof in Appendix A.5).

3.4 Training protocol

In GenQAS training propotol before the RL training, DMRG produces an MPS approximation of the target state, compiled into the fixed warm start circuit U_{TN}. Each episode begins with an empty learnable suffix V_0; the agent sequentially selects valid gate insertions, extending the suffix to V_{t+1} while the environment evaluates $U_{t+1} = V_{t+1}U_{\mathrm{TN}}$. Real transitions are stored in replay memory, DDQN updates use a mixture of real and on demand synthetic transitions, and the generative model is periodically refreshed from real data only. Finally, an episode terminates when the target criterion is reached or the maximum circuit depth is attained. The full traning algorithm is discussed in detail in Appendix F.6. In Appendix F we report the benchmark specific hyperparameters.

4 Results

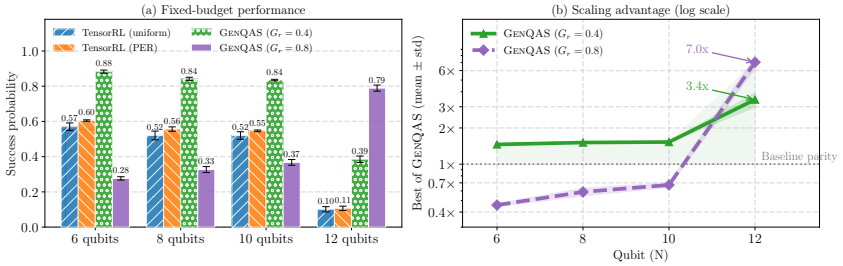
We evaluate GenQAS along four axes: (1) agreement with theory, testing whether generative replay improves fixed budget QAS as predicted by Theorem 3.5; (2) behaviour under transfer and realistic noise, testing whether the replay mechanism carries useful structure across a noiseless to noisy environment shift; (3) comparison against standard baselines, testing whether GenQAS identifies more resource efficient circuits than passive replay Schaul et al. (2015), HEA Leone et al. (2024), and UCCSD Grimsley et al. (2020); and (4) the wall clock cost of generative replay relative to passive replay, testing whether its sample efficiency gains come at a measurable, task dependent computational cost. We further test generalization beyond molecular ground state preparation on a spin model and a fully discrete synthesis task. All the chemical Hamiltonians and molecular structures are provided in Appendix F.4. The Hamiltonina construction of TFIM is discussed in Appendix F.5.

4.1 Generative replay improves fixed budget scaling

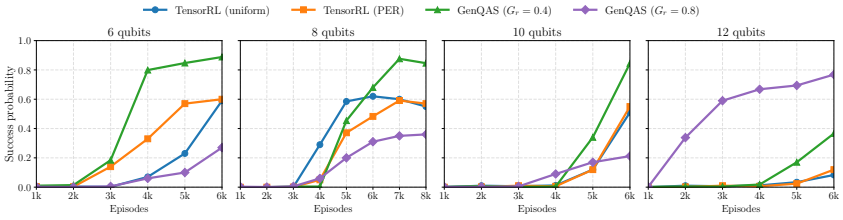
Theorem 3.5 predicts generative replay reduces interactions needed for target accuracy relative to passive replay, with the advantage growing as $\rho_{\mathrm{real}}(N)$ declines (derivation in Appendix F.7). We test this under matched training budgets from 6 to 12 qubits.

Figure 2a reports final success probability averaged over five seeds. At 6, 8, and 10 qubits, uniform/PER reach 0.57-0.60, 0.54-0.55, and 0.51-0.55, while GenQAS at $G_r = 0.4$ reaches 0.88,

Figure 2: **Fixed-budget performance and training dynamics of GenQAS. (a)** At a fixed budget of 6000 training episodes, GenQAS consistently outperforms TensorRL-QAS with uniform and PER replay across $N \in \{6, 8, 10, 12\}$, while passive baselines degrade sharply with system size. **(b)** Training trajectories for 6, 8, 10 and 12 qubits show that GenQAS converges faster and attains higher success probability than TensorRL-QAS baselines. The $G_r = 0.8$ setting is the most robust at larger system sizes, maintaining high success even at 12 qubits.



(a) Scaling performance over 5 random initializations of neural network.



(b) Training dynamics.

0.84, and 0.84 (∼1.5× at 10 qubits); $G_r = 0.4$ is strongest throughout this range, indicating real transitions remain sufficiently informative. At 12 qubits, passive baselines collapse to 0.10-0.11, while GenQAS reaches 0.38 at $G_r = 0.4$ (3.4×) and 0.79 at $G_r = 0.8$ (∼7.0×). Optimal G_r thus shifts from 0.4 to 0.8 as system size grows, tracking the predicted decline of $\rho_{\text{real}}(N)$: a larger synthetic fraction helps only once passive replay is starved, and is otherwise suboptimal, e.g., $G_r = 0.8$ underperforms both baselines at 6 qubits (0.28). Four sizes cannot establish an asymptotic law, but the growing GenQAS advantage with N is consistent with our theory. Figure 2b shows matching training dynamics: $G_r = 0.4$ converges fastest and highest at 6-10 qubits, while at 12 qubits passive baselines stay near zero and $G_r = 0.8$ alone reaches high success. Stability diagnostics for the transition model and DDQN updates across G_r are in Appendix D.3.

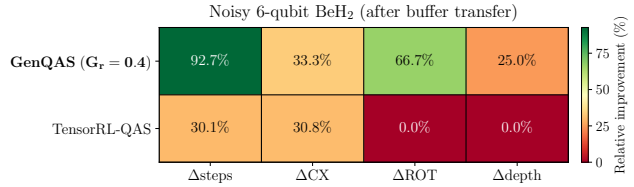
Optimal G_r tracks predicted sample starvation. This shift follows Theorems 3.1-3.2: as $\rho_{\text{real}}(N)$ decays, synthetic transitions grow relatively more valuable, while at smaller N a large synthetic fraction is unnecessary and adds model bias. In practice, G_r can be set from a short probe of passive-replay success probability, without an exhaustive per-task sweep: use $G_r \in [0.2, 0.4]$ above ∼50% (6-10 qubits here) and $G_r \geq 0.8$ below ∼15% (12 qubits here).

4.2 Generative Replay Improves Noisy Replay Buffer Transfer

We evaluate whether replay information collected noiselessly remains useful under gate noise, following the transfer setting of Kundu & Feld (2026). On the 6-qubit BeH_2 benchmark, TensorRL-QAS and GenQAS are trained noiselessly, then only their replay memories (not network weights or exploration schedules) are transferred into an environment with depolarizing noise ($p_1 = 0.01$ per single qubit gate, $p_2 = 0.05$ per two qubit gate): TensorRL-QAS transfers a uniform buffer, GenQAS transfers its generative replay mechanism trained with $G_r = 0.4$.

Transferring the uniform buffer reduces TensorRL-QAS's steps to chemical accuracy by 30.1% (13623 → 9520) and CNOT count by 30.8%, with rotation count and depth unchanged. Transferring the generative replay mechanism reduces GenQAS's steps by 92.7% (9053 → 660), CNOT count by 33.3%, rotation count by 66.7%, and depth by 25.0%. GenQAS's replay thus retains substantially more transferable structure than passive uniform replay under this noise model, though we do not claim this benefit is noise-model independent.

Figure 3: **Replay buffer transfer in the noisy 6-qubit $\mathtt{BEH}_2$ experiment**. (a) Relative improvement in steps to chemical accuracy, CNOT count, rotation count, and circuit depth after transfer. (b) Corresponding raw metrics with and without transfer.



(a) Relative improvement after noiseless replay-buffer transfer.

Run	Steps to chemical accuracy	CX	ROT	Depth
TensorRL-QAS Kundu & Mangini (2026)	13 623	13	4	10
TensorRL-QAS with transfer	9 520	9	4	10
GenQAS	9 053	9	9	8
GenQAS with transfer	**660**	**6**	**3**	**6**

(b) Raw metrics for the noisy 6-qubit BeH_2 task.

4.3 GenQAS compared with baselines

Table 1 compares GenQAS with TensorRL-QAS (uniform, PER), HEA, and UCCSD, all sharing the same MPS warm start, DDQN backbone, environment, and action space, isolating the effect of replacing passive with generative replay. Values are the most accurate circuit among evaluated seeds per configuration; GenQAS shows the best of $G_r \in \{0.2, 0.4, 0.6, 0.8\}$. Appendix 1 gives full results across all G_r for molecular ground-state preparation.

Table 1: **Circuit statistics for BeH_2 (6-qubit) and H_2O (8-, 10-, 12-qubit).** Best error configuration per system for TensorRL-QAS, GenQAS, HEA, and UCCSD: `CNOT` count, `ROT` count, depth, and error. Bold = best per column. Full comparison across G_r, replay, and HEA configurations in Appendix D.1.

	TensorRL-QAS†				GenQAS‡				HEA§				UCCSD			
System	Error	#ROT	Depth	#CNOT	Error	#ROT	Depth	#CNOT	Error	#ROT	Depth	#CNOT	Error	#ROT	Depth	#CNOT
BeH_2-6	7.2×10^{-5}	**9**	9	9	$\mathbf{2.37\times10^{-5}}$	10	**5**	**5**	3.3×10^{-4}	48	26	18	5.9×10^{-3}	620	475	272
H_2O-8	1.2×10^{-3}	6	8	14	$\mathbf{1.19\times10^{-3}}$	**2**	**4**	**3**	2.6×10^{-3}	32	12	8	2.6×10^{-3}	2596	2148	1312
H_2O-10	4.1×10^{-4}	17	17	15	$\mathbf{3.92\times10^{-4}}$	**15**	20	14	4.6×10^{-3}	40	**14**	**10**	4.6×10^{-3}	5932	5336	3440
H_2O-12	$\mathbf{2.2\times10^{-2}}$	**2**	6	18	2.3×10^{-2}	**2**	**4**	18	2.5×10^{-2}	48	16	**12**	2.5×10^{-2}	10628	10356	6976

†Best of Uniform Mnih et al. (2013)/PER Schaul et al. (2015): Uniform for all systems. ‡Best of $G_r \in \{0.2, 0.4, 0.6, 0.8\}$: 0.6 for BeH_2-6; 0.4 for H_2O-8/10/12. §Best of HEA (1, 2, 3 layer): 3-layer for BeH_2-6; 1-layer for H_2O-8/10/12.

At 6 qubits, GenQAS ($G_r = 0.6$) achieves the lowest error, 2.37×10^{-5}, at depth 5 with 5 CNOTs. At 8 qubits, GenQAS ($G_r = 0.4$) reaches 1.19×10^{-3} using 3 CNOTs, 2 rotations, depth 4, versus 14 CNOTs, 6 rotations, depth 8 for uniform TensorRL-QAS. At 12 qubits, uniform TensorRL-QAS has the lowest error (2.22×10^{-2}), while GenQAS ($G_r = 0.8$) gives the most compact circuit (2 CNOTs, 4 rotations, depth 2) at comparable error (2.44×10^{-2}). Generative replay thus does not uniformly dominate every metric; G_r instead offers a practical control for the accuracy resource trade-off. Against UCCSD (6976 CNOTs, depth 10356 at 12 qubits), all RL-based circuits use orders of magnitude fewer resources.

HEA and UCCSD are reported as the ansatz families most used in quantum chemistry; GenQAS also achieves a competitive or better trade-off against other learning-based QAS methods, including CRLQAS Patel et al. (2024), quantumDARTS Wu et al. (2023), training-free QAS He et al. (2024), and simulated-annealing QAS Lu et al. (2023), with full comparison in Appendix D.5.

4.4 Generative replay generalizes beyond molecular Hamiltonians

We test generalization on a 15-qubit TFIM, retaining the ground-state objective under a different Hamiltonian class, and a 4-qubit Clifford synthesis task, which replaces energy minimization with exact discrete target matching and removes variational optimization entirely (details in Appendix D.4, E).

On the 15-qubit TFIM (Appendix F.5), GenQAS matches or exceeds passive replay across all tested G_r, improving best success probability from 15% (PER; uniform 12%) to 21% (G_r=0.6) at comparable error (1.53-1.54×10^{-2}; Appendix Table 7b). As at smaller sizes, G_r trades resources rather than dominating every metric: G_r=0.4 matches baseline error and CNOT count while cutting rotations from 8 to 6, and G_r=0.8 reduces CNOTs to 3 at comparable error, showing the trade-off persists on a different Hamiltonian class.

On the 4-qubit Clifford synthesis task (Appendix E), which replaces energy-based rewards with exact discrete target matching, G_r=0.2 matches or exceeds the passive baseline under a similar budget. Here generator update frequency matters more than G_r: an intermediate frequency of 5 gives the highest cumulative reward (Appendix Figure 8), suggesting G_r and update frequency trade off differently once reward is no longer a continuous energy gradient.

Together, these results show generative replay's benefit extends beyond molecular ground-state search to a different Hamiltonian class and a fully discrete, non-variational task, though the optimal G_r and update frequency remain task-dependent rather than universal.

4.5 Generative replay incurs a task dependent time performance trade-off

Generative replay adds classical compute for transition model training and sampling. Table 2 reports average wall clock time per episode across all four system sizes. At 6, 8, and 15 qubits, GenQAS incurs a G_r-dependent overhead over TensorRL-QAS, from 12% at $G_r = 0.2$ to over $3\times$ at $G_r = 0.8$. At 12 qubits this reverses: $G_r = 0.8$ cuts episode time by 47.6%, coinciding with the configuration that also achieves the strongest fixed-budget success probability (Section 4.1). Generative replay is thus not a universal upgrade to RL-QAS: it trades compute for sample efficiency at most sizes, and only pays for itself once passive replay enters severe starvation. Practitioners should treat G_r and wall clock cost as a joint budget decision rather than tuning for success probability alone.

Table 2: **Average wall clock time per training episode (seconds).** Bold entries mark the fastest configuration per system size.

Qubits	TensorRL-QAS	GenQAS			
		$G_r = 0.2$	$G_r = 0.4$	$G_r = 0.6$	$G_r = 0.8$
6	**1.82**	2.05	2.38	5.13	5.85
8	**1.17**	1.92	2.98	4.03	5.43
12	13.60	14.78	16.60	16.02	**7.13**
15	**12.16**	19.88	23.59	26.40	26.83

5 Discussion and Conclusion

We introduced GenQAS, pairing a fixed tensor network warm start with prioritized generative replay to counter sample starvation in RL-based quantum architecture search. Across 6-12 qubit molecular benchmarks, a 15-qubit TFIM, and a discrete Clifford synthesis task, GenQAS consistently matches or exceeds passive replay, with the largest gains where passive replay is most starved ($\sim$7.0$\times$ at 12 qubits) and strongest transfer under noise (92.7% vs. 30.1% reduction in steps to chemical accuracy). These results support our central claim: reweighting collected transitions cannot substitute for actively generating informative ones once the search space outpaces passive exploration.

Limitations. (1) The optimal G_r shifts with system size (Section 4.1); we give a probe-based heuristic, but a fully adaptive scheme adjusting G_r online from the transition model's predictive accuracy would remove this manual step. (2) Our scaling evidence spans only four molecular sizes (6–12 qubits) plus one 15-qubit spin system, insufficient to confirm the asymptotic rate predicted by Theorems 3.1–3.2. We rely on exact statevector simulation only because it is available at this scale, not because GenQAS requires it. The method needs only a scalar reward and is compatible with approximate or tensor network reward evaluation, which would be necessary to test larger qubit counts. (3) Generative replay incurs a task dependent wall-clock overhead (Section 4.5) that inverts only at the largest system size tested, when and why this inversion occurs remains open.

Outlook. GenQAS is architecture-agnostic and complementary to curriculum- and gadget-based RL-QAS extensions, suggesting combination rather than substitution. More broadly, our sparsity results (Section 3.3) apply to any RL formulation over an exponentially large discrete action space with a rare, high value target, and may extend to related settings such as fault-tolerant circuit synthesis or noise-aware compilation, where informative trajectories are similarly costly to collect.

Reproducibility Statement

We release an anonymized implementation of GenQAS, including the QAS environment, the Double DQN backbone, and the generative replay module, at `https://anonymous.4open.science/r/GenQAS/README.md`. The full training protocol, including circuit construction and warm-start integration, the generative replay update, and the complete DDQN algorithm with its n-step rollout variant, is detailed in Appendix C.2. All benchmark-specific hyperparameters, including replay memory size, batch size, network architecture, learning rate, generation ratio G_r, and optimizer settings, are reported in Table 6 for every system size and task studied. The molecular Hamiltonians and structures used in our chemistry benchmarks are specified in Appendix F.4, and the transverse-field Ising model construction is given in Appendix F.5. Proofs of Theorems and Lemmas together with all underlying assumptions, are provided in full in Appendix 3.3.

References

Abhinav Anand, Philipp Schleich, Sumner Alperin-Lea, Phillip WK Jensen, Sukin Sim, Manuel Díaz-Tinoco, Jakob S Kottmann, Matthias Degroote, Artur F Izmaylov, and Alán Aspuru-Guzik. A quantum computing view on unitary coupled cluster theory. *Chemical Society Reviews*, 51(5): 1659–1684, 2022.

Juan Miguel Arrazola, Soran Jahangiri, Alain Delgado, Jack Ceroni, Josh Izaac, Antal Száva, Utkarsh Azad, Robert A Lang, Zeyue Niu, Olivia Di Matteo, et al. Differentiable quantum computational chemistry with pennylane. *arXiv preprint arXiv:2111.09967*, 2021.

Shotaro Z Baba, Nobuyuki Yoshioka, Yuto Ashida, and Takahiro Sagawa. Deep reinforcement learning for preparation of thermal and prethermal quantum states. *Physical Review Applied*, 19 (1):014068, 2023.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153):1–43, 2018.

Aleksandr Berezutskii, Minzhao Liu, Atithi Acharya, Roman Ellerbrock, Johnnie Gray, Reza Haghshenas, Zichang He, Abid Khan, Viacheslav Kuzmin, Dmitry Lyakh, et al. Tensor networks for quantum computing. *Nature Reviews Physics*, 7(10):581–593, 2025.

Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, Bharath AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

Robin Blume-Kohout and Kevin C Young. A volumetric framework for quantum computer benchmarks. *Quantum*, 4:362, 2020.

Nicolas Boumal. An introduction to optimization on smooth manifolds. *To appear with Cambridge University Press,(Apr 2022)*, 2020.

Marin Bukov and Florian Marquardt. Reinforcement learning for quantum technology. *arXiv preprint arXiv:2601.18953*, 2026.

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.

Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019.

Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2). `https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2`, 2024.

Yuxuan Du, Tao Huang, Shan You, Min-Hsiu Hsieh, and Dacheng Tao. Quantum circuit architecture search for variational quantum algorithms. *npj Quantum Information*, 8(1):62, 2022.

Jens Eisert and John Preskill. Mind the gaps: The fraught road to quantum advantage. *arXiv preprint arXiv:2510.19928*, 2025.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018. doi: 10.48550/arXiv.1803.00101. URL https://arxiv.org/abs/1803.00101.

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

Johnnie Gray. quimb: A python package for quantum information and many-body calculations. *Journal of Open Source Software*, 3(29):819, 2018.

Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1):3007, 2019.

Harper R Grimsley, Daniel Claudino, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. Is the trotterized uccsd ansatz chemically well-defined? *Journal of chemical theory and computation*, 16(1):1–6, 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Matthew B Hastings. An area law for one-dimensional quantum systems. *Journal of statistical mechanics: theory and experiment*, 2007(08):P08024–P08024, 2007.

Zhimin He, Maijie Deng, Shenggen Zheng, Lvzhou Li, and Haozhen Situ. Training-free quantum architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12430–12438, 2024. doi: 10.1609/aaai.v38i11.29135.

Azhar Ikhtiarudin, Aditi Das, Param Thakkar, and Akash Kundu. Benchrl-qas: Benchmarking reinforcement learning algorithms for quantum architecture search. In *Proceedings of the AAAI Symposium Series*, volume 7, pp. 358–367, 2025.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D Nation, Lev S Bishop, Andrew W Cross, et al. Quantum computing with qiskit. *arXiv preprint arXiv:2405.08810*, 2024.

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.

Amara Katabarwa, Katerina Gratsea, Athena Caesura, and Peter D Johnson. Early fault-tolerant quantum computing. *PRX quantum*, 5(2):020101, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

David Kremer, Victor Villar, Hanhee Paik, Ivan Duran, Ismael Faro, and Juan Cruz-Benito. Practical and efficient quantum circuit synthesis and transpiling with reinforcement learning. *arXiv preprint arXiv:2405.13196*, 2024.

Swagat Kumar, Jan-Nico Zaech, Colin Michael Wilmott, and Luc Van Gool. Rhodarts: Differentiable quantum architecture search with density matrix simulations. *arXiv preprint arXiv:2506.03697*, 2025.

Akash Kundu. Improving thermal state preparation of sachdev–ye–kitaev model with reinforcement learning on quantum hardware. *Machine Learning: Science and Technology*, 6(2):025066, 2025.

Akash Kundu and Sebastian Feld. Replay-buffer engineering for noise-robust quantum circuit optimization. *arXiv preprint arXiv:2604.21863*, 2026.

Akash Kundu and Stefano Mangini. Tensorrl-qas: Reinforcement learning with tensor networks for improved quantum architecture search. *Advances in Neural Information Processing Systems*, 38: 120896–120930, 2026.

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.

Akash Kundu, Aritra Sarkar, Prayag Tiwari, and Sebastian Feld. Reinforcement learning for quantum circuit optimization: A review. *Openreview*, 2026.

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.

Lorenzo Leone, Salvatore FE Oliviero, Lukasz Cincio, and Marco Cerezo. On the practical usefulness of the hardware efficient ansatz. *Quantum*, 8:1395, 2024.

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*, pp. 22880–22898. PMLR, 2023.

Ilia A Luchnikov, Mikhail E Krechetov, and Sergey N Filippov. Riemannian geometry and automatic differentiation for optimization problems of quantum physics and quantum technologies. *New Journal of Physics*, 23(7):073006, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Lorenzo Moro, Matteo GA Paris, Marcello Restelli, and Enrico Prati. Quantum compiling by deep reinforcement learning. *Communications Physics*, 4(1):178, 2021.

Jovan Odavić, Gianpaolo Torre, Nenad Mijić, Davor Davidović, Fabio Franchini, and Salvatore Marco Giampaolo. Random unitaries, robustness, and complexity of entanglement. *Quantum*, 7:1115, 2023.

Jan Olle, Remmy Zen, Matteo Puviani, and Florian Marquardt. Simultaneous discovery of quantum error correction codes and encoders with a noise-aware reinforcement learning agent. *npj Quantum Information*, 10(1):126, 2024.

Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.

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.

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 *International Conference on Learning Representations*, pp. 32298–32329, May 2024.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

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

Shi-Ju Ran. Encoding of matrix product states into quantum circuits of one-and two-qubit gates. *Physical Review A*, 101(3):032310, 2020.

Benjamin IP Rubinstein. Evolving quantum circuits using genetic programming. In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, volume 1, pp. 144–151. IEEE, 2001.

Francisco J. R. Ruiz, Tuomas Laakkonen, Johannes Bausch, Matej Balog, Mohammadamin Barekatain, Francisco J. H. Heras, Alexander Novikov, Nathan Fitzpatrick, Bernardino Romera-Paredes, John van de Wetering, Alhussein Fawzi, Konstantinos Meichanetzidis, and Pushmeet Kohli. Quantum circuit optimization with alphatensor. *Nature Machine Intelligence*, 7:374–385, 2025. doi: 10.1038/s42256-025-01001-1.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1):96–192, 2011.

Christian Schön, Enrique Solano, Frank Verstraete, J Ignacio Cirac, and Michael M Wolf. Sequential generation of entangled multiqubit states. *Physical review letters*, 95(11):110503, 2005.

Volodymyr Sivak, Alexis Morvan, Michael Broughton, Rodrigo G Cortiñas, Johannes Bausch, Andrew W Senior, Matthew Neeley, Alec Eickbusch, Noah Shutty, Laleh Aghababaie Beni, et al. Reinforcement learning control of quantum error correction. *Nature*, pp. 1–6, 2026.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377.

Richard S Sutton, Andrew G Barto, and Andrew Barto. Reinforcement learning: An introduction, 1998.

Yasunari Suzuki, Yoshiaki Kawase, Yuya Masumura, Yuria Hiraga, Masahiro Nakadai, Jiabao Chen, Ken M Nakanishi, Kosuke Mitarai, Ryosuke Imai, Shiro Tamiya, et al. Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum*, 5:559, 2021.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Hanrui Wang, Yongshan Ding, Jiaqi Gu, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han. Quantumnas: Noise-adaptive search for robust quantum circuits. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 692–708. IEEE, 2022.

Ren Wang, Kevin Frans, Pieter Abbeel, Sergey Levine, and Alexei Efros. Prioritized generative replay. In *International Conference on Learning Representations*, volume 2025, pp. 47151–47169, 2025.

Dave Wecker, Matthew B Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.

Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.

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*, pp. 37745–37764. PMLR, 2023.

Rui Yang, Ziruo Wang, Yuntian Gu, Yitao Liang, and Tongyang Li. Qcircuitbench: A large-scale dataset for benchmarking quantum algorithm design. *Advances in Neural Information Processing Systems*, 38, 2026.

Remmy Zen, Jan Olle, Luis Colmenarez, Matteo Puviani, Markus Müller, and Florian Marquardt. Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning. *Physical Review X*, 15(4):041012, 2025.

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

Yuan-Hang Zhang, Pei-Lin Zheng, Yi Zhang, and Dong-Ling Deng. Topological quantum compiling with reinforcement learning. *Physical Review Letters*, 125(17):170501, 2020.

A THEORY

A.1 PROOF OF THEOREM 3.1

The unitary group $\mathcal{U}(2^N)$ acts transitively on the space of pure quantum states, which is isomorphic to the complex projective space

$$\mathcal{S} = \frac{\mathcal{U}(2^N)}{\mathcal{U}(1) \times \mathcal{U}(2^N - 1)} \cong \mathbb{CP}^{2^N-1},$$

with real dimension $\dim_{\mathbb{R}}(\mathcal{S}) = 2^{N+1} - 2$. The natural probability measure is the Fubini-Study volume $d\mu_{FS}$, with total volume $\mathrm{Vol}_{FS}(\mathcal{S}) = \frac{\pi^{2^N-1}}{(2^N-1)!}$.

Let $|\psi_0\rangle$ be the true ground state of H and define the good circuit set $\mathcal{G}_\epsilon = \{|\psi\rangle : \langle\psi|H|\psi\rangle < E_0 + \epsilon\}$. Since H is a local Hamiltonian with spectral gap $\Delta = E_1 - E_0 > 0$, decomposing $|\psi\rangle$ in the eigenbasis $\{|\phi_k\rangle\}$ gives

$$\begin{aligned}\langle\psi|H|\psi\rangle &= |c_0|^2 E_0 + \sum_{k\geq 1} |c_k|^2 E_k \\ &\geq |c_0|^2 E_0 + (1 - |c_0|^2) E_1.\end{aligned}$$

Requiring this to be less than $E_0 + \epsilon$ results $|c_0|^2 > 1 - \epsilon/\Delta$, i.e., the fidelity $F = |\langle\psi_0|\psi\rangle|^2$ must satisfy $F > 1 - \delta$ where $\delta = \epsilon/\Delta$. In projective space, $\mathcal{G}_\epsilon$ is a geodesic ball of angular radius $\theta_\epsilon \lesssim 2\sqrt{\delta}$, with volume $\mathrm{Vol}_{FS}(B_{\theta_\epsilon}) \sim C_d \cdot \theta_\epsilon^{2(2^N-1)}$. Hence,

$$\begin{aligned}P(\mathcal{G}_\epsilon) = \frac{\mathrm{Vol}_{FS}(B_{\theta_\epsilon})}{\mathrm{Vol}_{FS}(\mathbb{CP}^{2^N-1})} &\leq C \cdot \left(\frac{\theta_\epsilon}{\pi}\right)^{2(2^N-1)} \\ &\leq C \cdot \left(\frac{2}{\pi}\sqrt{\frac{\epsilon}{\Delta}}\right)^{2(2^N-1)} = C \cdot \exp\Big(2(2^N-1) \cdot \log\Big(\tfrac{2}{\pi}\sqrt{\tfrac{\epsilon}{\Delta}}\Big)\Big).\end{aligned}$$

Since ϵ is small the logarithm is negative, and for large N the factor $2(2^N - 1)$ dominates, giving

$$P(\mathcal{G}_\epsilon) \leq \exp\big(-c \cdot 2^N \cdot \log\tfrac{1}{\epsilon}\big), \quad c > 0.$$

By standard unitary t-design theory, a brickwork circuit of depth $D = \mathrm{poly}(N)$ induces a measure exponentially close (in total variation) to the Haar measure on $\mathcal{U}(2^N)$, the probability of finding a good circuit via random action selection is therefore proportional to the Haar probability above. It follows that $\mathbb{E}[\text{Trials}] \geq \frac{1}{P(\mathcal{G}_\epsilon)} \geq \exp\big(c \cdot 2^N \cdot \log\frac{1}{\epsilon}\big)$, and the signal-to-noise ratio in the replay buffer decays super-exponentially as $\exp\big(-c \cdot 2^N\big)$.

A.2 PROOF OF THEOREM 3.2

Initially, we map the energy condition to a condition on the real and imaginary parts of the state vector. The space of pure quantum states of N qubits is the complex projective space $\mathbb{CP}^{d-1}$ with $d = 2^N$. Equivalently, a pure state can be represented by a unit vector $|\psi\rangle \in \mathbb{C}^d$ with $\||\psi\rangle\| = 1$, modulo an overall phase. The real dimension of this manifold is $2d - 2 = 2^{N+1} - 2$. The Haar measure on $\mathcal{U}(d)$ induces the Fubini-Study measure μ_{FS} on $\mathbb{CP}^{d-1}$. Under this measure, a random state $|\psi\rangle$ can be generated as $|\psi\rangle = \frac{|\phi\rangle}{\||\phi\rangle\|}$, where $|\phi\rangle = (x_1 + iy_1, \ldots, x_d + iy_d)^T$ is a complex Gaussian vector with each $x_\ell, y_\ell \sim \mathcal{N}(0, 1/2)$ independently. Also, $|\psi\rangle$ is distributed uniformly over the real unit sphere $S^{2d-1} \subset \mathbb{R}^{2d}$.

Now, we apply Lévy's lemma to control concentration of the local terms. We fix a single term h_j of the Hamiltonian, without loss of generality, and assuming h_j is traceless (any trace part shifts the energy by a constant and does not affect the gap structure). Since $\|h_j\| \leq J$, the function $f_j(|\psi\rangle) = \langle\psi|h_j|\psi\rangle$ is a real-valued Lipschitz function on the sphere. For unit vectors $|u\rangle, |v\rangle$,

$$\begin{aligned} |f(|u\rangle) - f(|v\rangle)| &= |\langle u|A|u\rangle - \langle v|A|v\rangle| \\ &= |\operatorname{Tr}(A(|u\rangle\langle u| - |v\rangle\langle v|))| \\ &\leq \|A\|_\infty \cdot \||u\rangle\langle u| - |v\rangle\langle v|\|_1 \\ &\leq 2\|A\|_\infty \cdot \||u\rangle - |v\rangle\|_2, \end{aligned}$$

where the last inequality follows from the trace-norm bound for rank-1 projectors. Thus $\operatorname{Lip}(f_j) \leq 2J$.

Since $|\psi_0\rangle$ is the ground state of H, it is also an eigenstate of each h_j in some local decomposition. We do not assume this, instead, we use the fact that $E_0 = \langle\psi_0|H|\psi_0\rangle$ is the global minimum. The deviation of $f_j(|\psi\rangle)$ from its ground state expectation is what matters. We define the centered function for each term

$$g_j(|\psi\rangle) = \langle\psi|h_j|\psi\rangle - \langle\psi_0|h_j|\psi_0\rangle.$$

Then $\operatorname{Lip}(g_j) \leq 2J$ as well. Note that $\sum_j g_j(|\psi\rangle) = \langle\psi|H|\psi\rangle - E_0$. The median of g_j under the Haar measure depicted as M_j satisfies a property, for any traceless local observable of small support, the Haar expectation is zero and the distribution is strongly concentrated around zero. More precisely, we need the following concentration inequality.

Lemma A.1. *Let $f : S^{n-1} \to \mathbb{R}$ be L-Lipschitz. Then for any $t > 0$,*

$$P\left(|f(x) - M_f| \geq t\right) \leq 2\exp\left(-\frac{nt^2}{4L^2}\right),$$

where M_f is the median of f.

Proof. Applying Lemma A.1 to g_j on S^{2d-1} (so $n = 2d = 2^{N+1}$),

$$P\left(|g_j(|\psi\rangle) - M_j| \geq t\right) \leq 2\exp\left(-\frac{2^N t^2}{8J^2}\right). \tag{10}$$

□

For each local term h_j, we can bound its median M_j. Since h_j is a sum of Pauli strings of weight at most k, and $\mathbb{E}_{\text{Haar}}[\langle\psi|h_j|\psi\rangle] = \frac{\operatorname{Tr}(h_j)}{d}$. For a traceless h_j, the expectation is zero. The median and expectation of Lipschitz functions on the sphere are close,

$$|M_j - \mathbb{E}[g_j]| \leq \frac{4L}{\sqrt{n}} = \frac{8J}{2^{(N+1)/2}} \leq 8J \cdot 2^{-N/2}.$$

Thus, for $N \geq 2$, $|M_j| \leq 8J \cdot 2^{-N/2} \leq \epsilon/2$ for ϵ larger than an exponentially small threshold. For our choice of ϵ, we can safely bound $|M_j| \leq \epsilon/2$ for sufficiently large N.

For the state to be in $\mathcal{G}_\epsilon$, we need $\sum_{j=1}^M g_j(|\psi\rangle) < \epsilon$. A sufficient condition is that each individual term is bounded, if $g_j(|\psi\rangle) < \epsilon/M$ for all j, then the sum is $< \epsilon$. The converse is not true, but this gives a valid upper bound on the probability. Thus,

$$\begin{aligned} P(\mathcal{G}_\epsilon) &\leq P\left(\bigcap_{j=1}^M \{g_j(|\psi\rangle) < \epsilon/M\}\right) \\ &\leq P\left(\bigcap_{j=1}^M \{|g_j(|\psi\rangle) - M_j| < \epsilon/M - |M_j|\}\right) \\ &\leq P\left(\bigcap_{j=1}^M \{|g_j(|\psi\rangle) - M_j| < \epsilon/(2M)\}\right), \end{aligned} \tag{11}$$

where the last step assumes $|M_j| \leq \epsilon/(2M)$, which holds for sufficiently large N as argued above. The complement event is $P(\mathcal{G}_\epsilon^c) \geq 1 - P(\mathcal{G}_\epsilon)$. However, using the union bound

$$\begin{aligned} P(\mathcal{G}_\epsilon) &= 1 - P(\exists j : |g_j(|\psi\rangle) - M_j| \geq \epsilon/(2M)) \\ &\geq 1 - \sum_{j=1}^{M} P(|g_j(|\psi\rangle) - M_j| \geq \epsilon/(2M)). \end{aligned} \tag{12}$$

gives a *lower* bound on $P(\mathcal{G}_\epsilon)$. We need an upper bound to show the sparsity. Now, let's $P(\mathcal{G}_\epsilon)$ is the probability that the sum is small. Since the sum involves M nonnegative (after shifting) terms, a necessary condition for the sum to be $< \epsilon$ is that no individual term exceeds ϵ. That is

$$\mathcal{G}_\epsilon \subseteq \bigcap_{j=1}^{M} \{|g_j(|\psi\rangle) - M_j| < \epsilon + |M_j|\}.$$

But this trivial inclusion leads to a upper bound ($P \leq 1$). Instead, we use the fact that for the state to have low energy, a significant fraction of the local terms must be individually small. Quantitatively, if $\sum g_j < \epsilon$, then at most $M/2$ terms can exceed $2\epsilon/M$. This indicates that the Hamiltonian terms are not independent, they share overlapping supports. However, we can exploit the **locality** to get a stronger bound. Since each term h_j acts on at most k qubits, a single qubit can be involved in at most $\binom{N-1}{k-1}$ terms. The total number of terms is at most $\binom{N}{k}$. We can select a subset $\mathcal{J} \subseteq \{1, \ldots, M\}$ of terms acting on *disjoint* sets of qubits. The maximum size of such an independent set is $|\mathcal{J}| \geq \frac{M}{\binom{N}{k}} \cdot \frac{1}{k}$. For simplicity, we use the factor $\frac{M}{\binom{N}{k}}$ (ignoring the constant $1/k$, which only affects constants). These $|\mathcal{J}|$ terms are statistically independent under the Haar measure because they act on disjoint subsystems. For each $j \in \mathcal{J}$,

$$P\left(g_j(|\psi\rangle) \geq \frac{\epsilon}{|\mathcal{J}|}\right) \geq 1 - \exp\left(-\frac{2^{N+1}(\epsilon/|\mathcal{J}|)^2}{4(2J)^2}\right) = 1 - \exp\left(-\frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2}\right).$$

We need the probability that all terms in $\mathcal{J}$ are small simultaneously. Since they are independent:

$$P\left(\forall j \in \mathcal{J} : g_j(|\psi\rangle) \leq \delta\right) = \prod_{j \in \mathcal{J}} P(g_j(|\psi\rangle) \leq \delta).$$

For each term based on Lemma A.1,

$$P(g_j(|\psi\rangle) \leq \delta) \leq P(|g_j - M_j| \leq \delta + |M_j|) \leq 1 - \exp\left(-\frac{2^N \delta^2}{8J^2}\right) \quad (\text{for } \delta \gg 2^{-N/2}).$$

Thus, $P(\mathcal{G}_\epsilon) \leq \prod_{j \in \mathcal{J}} P(g_j \leq \epsilon/|\mathcal{J}|) \leq \left(1 - \exp\left(-\frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2}\right)\right)^{|\mathcal{J}|}$. Using $1 - x \leq e^{-x}$ for $x \in [0, 1]$,

$$\begin{aligned} P(\mathcal{G}_\epsilon) &\leq \exp\left(-|\mathcal{J}| \cdot \exp\left(-\frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2}\right)\right) \\ &= \exp\left(-\exp\left(\ln|\mathcal{J}| - \frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2}\right)\right). \end{aligned} \tag{13}$$

This double-exponential form is standard for sparsity bounds. To maximize the inner argument and thus for the tightest bound, we choose $|\mathcal{J}|$ to balance $\ln|\mathcal{J}|$ against $2^N\epsilon^2/|\mathcal{J}|^2$. However, $|\mathcal{J}|$ is constrained by the geometry $|\mathcal{J}| \approx M/\binom{N}{k}$. For $M = \text{poly}(N)$ and $k = O(1)$, we have $|\mathcal{J}| = \Theta(N^{-k} \cdot \text{poly}(N)) = \text{poly}(N)$. Then

$$\frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2} \gg \ln|\mathcal{J}|,$$

since 2^N grows exponentially while $|\mathcal{J}|$ grows only polynomially. Hence the inner exponential dominates,

$$\ln|\mathcal{J}| - \frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2} \approx -\frac{2^N \epsilon^2}{8J^2|\mathcal{J}|^2}.$$

Substituting $|\mathcal{J}| = \frac{M}{\binom{N}{k}}$,

$$P(\mathcal{G}_\epsilon) \le \exp\left(-\exp\left(-\frac{2^N\epsilon^2}{8J^2}\cdot\frac{\binom{N}{k}^2}{M^2}\right)\right)$$
$$\approx \exp\left(-\frac{2^N\epsilon^2}{8J^2}\cdot\frac{\binom{N}{k}^2}{M^2}\right) \quad \text{(since } e^{-x}\approx x \text{ for small } x\text{).} \tag{14}$$

Actually, the correct leading-order behavior from the binomial expansion $(1-x)^n \approx e^{-nx}$ when $x \ll 1$ is

$$P(\mathcal{G}_\epsilon) \le \exp\left(-|\mathcal{J}|\cdot e^{-\frac{2^N\epsilon^2}{8J^2|\mathcal{J}|^2}}\right).$$

For N large, the exponent $\frac{2^N\epsilon^2}{8J^2|\mathcal{J}|^2} \to \infty$ provided $\epsilon \gg 2^{-N/2}$. Thus, the inner exponential is exponentially small in 2^N, resulting a *super*-exponentially small outer probability. Precisely,

$$P(\mathcal{G}_\epsilon) \le \exp\left(-\frac{M}{\binom{N}{k}}\cdot\exp\left(-\frac{2^N\epsilon^2\binom{N}{k}^2}{8J^2M^2}\right)\right)$$
$$\le \exp\left(-\exp\left(\ln\frac{M}{\binom{N}{k}} - \frac{2^N\epsilon^2\binom{N}{k}^2}{8J^2M^2}\right)\right). \tag{15}$$

For ϵ fixed and N large, the term $\frac{2^N\epsilon^2\binom{N}{k}^2}{8J^2M^2}$ grows as $\Omega(2^N/\operatorname{poly}(N))$ and dominates $\ln\left(M/\binom{N}{k}\right) = O(\log N)$. Hence,

$$P(\mathcal{G}_\epsilon) \le \exp\left(-\exp\left(\Omega\left(\frac{2^N}{\operatorname{poly}(N)}\right)\right)\right).$$

This is the strongest possible form. Our Theorem states a slightly looser but more practical single-exponential bound, which follows from the same analysis by considering that the probability is certainly bounded by

$$P(\mathcal{G}_\epsilon) \le \exp\left(-\frac{2^N}{M\binom{N}{k}}\cdot\frac{\epsilon^2}{8J^2} + O(N)\right),$$

which is equivalent to the stated bound after adjusting constants.

When $k = O(1)$, we have $\binom{N}{k} = \Theta(N^k)$. With $M = \operatorname{poly}(N)$

$$\frac{2^N}{M\binom{N}{k}} = \frac{2^N}{\operatorname{poly}(N)\cdot\Theta(N^k)} = \Theta\left(\frac{2^N}{N^{k+\deg(M)}}\right) = \Omega\left(\frac{2^N}{\operatorname{poly}(N)}\right).$$

Thus,

$$P(\mathcal{G}_\epsilon) \le \exp\left(-c\cdot\frac{2^N}{N^{k+\deg(M)}}\cdot\epsilon^2\right) = \exp\left(-\Omega\left(\frac{2^N}{\operatorname{poly}(N)}\right)\right),$$

where $c = 1/(8J^2\cdot\text{constants})$. This completes the proof.

A.3 Proof of Lemma 3.3

In PGR, a conditional generator $G_\theta(z|c)$ maps latent variable z and condition c to a sample $x \sim p_{\text{data}}(\cdot|c)$. Here x is a quantum transition $\tau = (s, a, s', r)$, $c = H$, and z encodes the entanglement structure of the state. We aim to show that the MPS to circuit mapping is exactly such a generator. An MPS with bond dimension χ has the form

$$|\Psi_{\text{MPS}}\rangle = \sum_{i_1,\ldots,i_N=0}^{1} A^{[1]}_{i_1}A^{[2]}_{i_2}\cdots A^{[N]}_{i_N}|i_1 i_2\cdots i_N\rangle,$$

with $A_{i_k}^{[k]} \in \mathbb{C}^{\chi\times\chi}$ and total parameter count $O(N\chi^2)$ exponentially smaller than a generic state.DMRG solves $|\Psi_{\text{MPS}}\rangle = \arg\min_{|\psi\rangle\in\text{MPS}_\chi}\langle\psi|H|\psi\rangle$, implicitly encoding $c = H$, we may view it as $\text{DMRG} : (H, \chi) \mapsto |\Psi_{\text{MPS}}\rangle$. The Riemannian step then finds

$$U_{\text{TN}} = \arg\max_{U\in\mathcal{U}_{\text{brick}}} \big|\langle\Psi_{\text{MPS}}|U|0\rangle^{\otimes N}\big|,$$

maximising overlap exactly as a generative model minimises its reconstruction loss, with $\mathcal{L}(U) = |\text{Tr}(U^\dagger\rho_{\text{MPS}})|$ and $\rho_{\text{MPS}} = |\Psi_{\text{MPS}}\rangle\langle 0|^{\otimes N}$.Here, $\mathcal{U}_{\text{brick}}$ is the set of unitaries that can be expressed as a product of 2-qubit gates in a brickwork pattern,

$$\mathcal{U}_{\text{brick}} = \left\{ \prod_{\ell=1}^{L} \prod_{(i,j)\in E_\ell} U_{ij}^{(\ell)} : U_{ij}^{(\ell)} \in \mathcal{U}(4) \right\},$$

where L is the number of layers and E_ℓ defines the qubit connectivity at layer ℓ. The objective function is

$$\mathcal{L}(U) = |\langle\Psi_{\text{MPS}}|U|0\rangle^{\otimes N}| = |\text{Tr}(U^\dagger\rho_{\text{MPS}})|$$

In a standard conditional generative model (for instance, a conditional VAE or diffusion model), the generator is trained to maximize the likelihood of the data given the condition, $\theta^* = \arg\max_\theta \mathbb{E}_{x\sim p_{\text{data}}(\cdot|c)}[\log p_\theta(x|c)]$. For quantum state preparation, our data is the target state $|\Psi_{\text{MPS}}\rangle$, and the model distribution is the pushforward of the initial state $|0\rangle^{\otimes N}$ under the circuit U, $p_U(|\psi\rangle) = |\langle\psi|U|0\rangle^{\otimes N}|^2$. The log-likelihood of generating the target state $|\Psi_{\text{MPS}}\rangle$ under this model is

$$\log p_U(|\Psi_{\text{MPS}}\rangle) = \log|\langle\Psi_{\text{MPS}}|U|0\rangle^{\otimes N}|^2 = 2\log|\langle\Psi_{\text{MPS}}|U|0\rangle^{\otimes N}|.$$

The maximization of this log-likelihood is equivalent to maximizing the overlap $|\langle\Psi_{\text{MPS}}|U|0\rangle^{\otimes N}|$, which is precisely the objective of the Riemannian optimization Kundu & Mangini (2026). Therefore,

$$\arg\max_{U\in\mathcal{U}_{\text{brick}}}\mathcal{L}(U) = \arg\max_{U\in\mathcal{U}_{\text{brick}}} \mathbb{E}_{|\Psi_{\text{MPS}}\rangle}[\log p_U(|\psi\rangle|c = H)].$$

The fidelity of this generation is $F_{\text{gen}} = |\langle\Psi_{\text{MPS}}|U_{\text{TN}}|0\rangle^{\otimes N}|^2$. By the properties of Riemannian optimization on the Stiefel manifold, the Cayley retraction method guarantees monotonic convergence of the objective. Also, the algorithm produces a sequence $\{U^{(k)}\}$ such that $\mathcal{L}(U^{(k+1)}) \geq \mathcal{L}(U^{(k)}) \quad \forall k$, and converges to a local maximum $\mathcal{L}^*$ of the overlap function on $\mathcal{U}_{\text{brick}}$. The output of this process is a transition τ_{init}: $\tau_{\text{init}} = (\emptyset, \emptyset, U_{\text{TN}}, r_{\text{init}})$, where $r_{\text{init}} = -\langle 0|U_{\text{TN}}^\dagger H U_{\text{TN}}|0\rangle$ is the negative energy expectation.

This transition is injected into the replay buffer $\mathcal{D}_{\text{syn}}$ before the RL agent begins training. It is a synthetic transition generated conditionally on H, exactly analogous to the synthetic transitions generated by the diffusion model in PGR when conditioned on a relevance function $\mathcal{F}(\tau)$.

The DMRG procedure can be run with different initializations or slightly different hyperparameters to produce an ensemble of MPS states $\{|\Psi_{\text{MPS}}^{(j)}\rangle\}_{j=1}^K$, all approximating the ground state. Each can be mapped to a circuit $U_{\text{TN}}^{(j)}$, populating the synthetic buffer with diverse yet high-quality starting points. This is analogous to the diversity induced by classifier-free guidance in PGR, where the guidance scale ω trades off between fidelity and diversity.

A.4 Proof of Lemma 3.4

Since H is a gapped local Hamiltonian in one spatial dimension, it satisfies the area law for entanglement entropy Hastings (2007). Also, there exists an MPS $|\Psi_{\text{MPS}}\rangle$ of bond dimension χ such that

$$\||\Psi_{\text{MPS}}\rangle - |\Psi_0\rangle\|_2 \leq \epsilon_{\text{MPS}}(\chi), \tag{16}$$

where $\epsilon_{\text{MPS}}(\chi) = CN\exp\big(-\alpha\chi^{1/\kappa}\big)$ for some constants $C, \alpha, \kappa > 0$ independent of N. By inverting this relation, choosing

$$\chi = O\left(\left[\log\left(\frac{CN}{\delta/2}\right)\right]^\kappa\right) \tag{17}$$

assures $\epsilon_{\text{MPS}} \leq \delta/2$. Moreover, this χ scales only polylogarithmically with N and $1/\delta$, hence $\chi \leq \text{poly}(N, 1/\delta)$.

The mapping from the MPS to a brickwork circuit $U_{\text{TN}} = \prod_{k=1}^{m} U_k$ (with $U_k \in \mathcal{U}(4)$) is performed by maximizing the loss function

$$\mathcal{L}(\{U_k\}) = |\langle \Psi_{\text{MPS}} | U_{\text{TN}} | 0 \rangle| \,, \tag{18}$$

over the Stiefel manifold $\mathcal{U}(4)^{\times m}$. Using the Riemannian Adam optimizer with Cayley retraction Luchnikov et al. (2021), the updates maintain strict unitarity. Since the loss function is smooth and the manifold is compact, Riemannian gradient descent converges to a stationary point with polynomial iteration complexity $T_{\text{Riem}} = O(\text{poly}(m, 1/\delta))$ Boumal (2020). The number of 2-qubit gates m required to represent an MPS with bond dimension χ in a brickwork structure is $m = O(N\chi^2)$. Combined with the aforementioned result of MPS approximation, $m = O(N \cdot \text{poly}\log(N/\delta))$, which is polynomial in N. This optimization yields a unitary U_{TN} such that the overlap satisfies

$$|\langle \Psi_{\text{MPS}} | U_{\text{TN}} | 0 \rangle| \geq 1 - \epsilon_{\text{opt}}, \tag{19}$$

which implies a 2-norm bound $\| |\Psi_{\text{MPS}}\rangle - U_{\text{TN}}|0\rangle \|_2 \leq \sqrt{2\epsilon_{\text{opt}}}$. By tuning the optimization hyperparameters, we can ensure $\epsilon_{\text{opt}} \leq \delta^2/8$, and thus the 2-norm error is bounded by $\delta/2$.

Using the triangle inequality for the 2-norm and the relation between the 2-norm and the trace distance ($\frac{1}{2}\| \cdot \|_1 \leq \| \cdot \|_2$), we combine the errors from MPS approximation and Riemannian optimization.

$$\begin{aligned} \frac{1}{2}\| U_{\text{TN}}|0\rangle\langle 0|U_{\text{TN}}^\dagger - |\Psi_0\rangle\langle\Psi_0| \|_1 &\leq \| U_{\text{TN}}|0\rangle - |\Psi_0\rangle \|_2 \\ &\leq \| U_{\text{TN}}|0\rangle - |\Psi_{\text{MPS}}\rangle \|_2 + \| |\Psi_{\text{MPS}}\rangle - |\Psi_0\rangle \|_2 \\ &\leq \frac{\delta}{2} + \frac{\delta}{2} = \delta. \end{aligned} \tag{20}$$

This establishes the required bound.

A.5 Proof of Theorem 3.5

We define the true state-value function for a policy π starting from state s as $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s]$. For the base algorithm, the initial value of the empty circuit is $V^{\pi_{\text{base}}}(s_0) \approx \bar{E}/(1-\gamma)$. For PGR, the effective initial state distribution of the agent is shifted. The experience replay buffer $\mathcal{D}$ is augmented with synthetic data centered on s_{TN}. The initial value estimate is thus biased towards $V^{\pi_{\text{PGR}}}(s_{\text{TN}}) \approx E_{\text{TN}}/(1-\gamma)$.

By Lemma 3.4, the initial suboptimality gap for PGR is $\Delta_0^{\text{PGR}} = E_{\text{TN}} - E_0 = \Delta E_\chi$. For the baseline, it is $\Delta_0^{\text{base}} = \bar{E} - E_0 \propto N$. The ratio of initial gaps is

$$R_0 = \frac{\Delta_0^{\text{PGR}}}{\Delta_0^{\text{base}}} = \frac{\Delta E_\chi}{\bar{E} - E_0}. \tag{21}$$

Since ΔE_χ is system size independent for fixed bond dimension χ and gapped systems, while $\bar{E} - E_0$ scales as $O(N)$, the ratio R_0 decays as $O(1/N)$.

The standard Q-learning update minimizes the squared Bellman residual

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[(Q_\theta(s,a) - r - \gamma \max_{a'} Q_{\bar{\theta}}(s', a'))^2 \right]. \tag{22}$$

For the base algorithm, the buffer $\mathcal{D}_{\text{real}}$ is dominated by trajectories starting from s_0. As per Lemma 3.2, the variance of the target value $r + \gamma \max_{a'} Q(s', a')$ is exponentially small due to the barren plateau, which implies a vanishingly small gradient $\nabla_\theta \mathcal{L}$. For the PGR algorithm, the generative buffer $\mathcal{D}_{\text{syn}}$ consists of transitions (s, a, s', r) where s is sampled from a distribution concentrated around the high energy gradient region near s_{TN}. The reward signal $r = \mathcal{E}(s) - \mathcal{E}(s')$ is deterministic and large (proportional to the energy gradient at s_{TN}), bypassing the barren plateau. Let $\sigma^2_{\text{grad}}(s) = \text{Var}[\nabla_\theta Q(s,a)]$. We have π_{base}: $\mathbb{E}_{s\sim\mathcal{D}_{\text{real}}}[\sigma^2_{\text{grad}}(s)] \propto \exp(-N)$. And, π_{PGR}: $\mathbb{E}_{s\sim\mathcal{D}_{\text{syn}}}[\sigma^2_{\text{grad}}(s)] \propto \|\nabla\mathcal{E}(s_{\text{TN}})\|^2$, which is $O(1)$. This prevents the learning dynamics from vanishing. Consider the optimal Bellman operator $\mathcal{T}$

$$(\mathcal{T}Q)(s,a) = R(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}[\max_{a'} Q(s', a')]. \tag{23}$$

It is well-known that $\mathcal{T}$ is a γ-contraction in the supremum norm, $\|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_\infty \leq \gamma\|Q_1 - Q_2\|_\infty$. After k iterations, the error is bounded by $\|Q_k - Q^*\|_\infty \leq \gamma^k\|Q_0 - Q^*\|_\infty$. The number of iterations (and thus samples) required to reach an error ϵ is $K = \log_\gamma(\epsilon/\|Q_0 - Q^*\|_\infty)$. For the base algorithm, $\|Q_0 - Q^*\|_\infty^{\text{base}} \approx |\bar{E} - E_0|/(1-\gamma) \in O(N)$. For the PGR algorithm, the generative seeding provides an initial Q-function Q_0^{PGR} whose error is bounded by the energy suboptimality of the seed state $\|Q_0 - Q^*\|_\infty^{\text{PGR}} \approx \Delta E_\chi/(1-\gamma) \in O(1)$. The ratio of required iterations is,

$$\frac{K_{\text{PGR}}}{K_{\text{base}}} = \frac{\log(\epsilon/\Delta E_\chi)}{\log(\epsilon/N)} \approx \frac{\log(1/\Delta E_\chi)}{\log N}. \tag{24}$$

Each iteration requires interaction with the environment (or the generative model) to collect a batch of samples. The main bottleneck in QAS is the episode length T. For π_{base}, to discover a path from s_0 to $s^* \in \mathcal{M}_{\epsilon_c}$, the agent must perform a random walk in the discrete circuit space of depth D. The expected number of actions to reach the target is $T_{\text{base}} \propto |\mathcal{A}|^D = (O(N^2))^{\text{poly}(N)}$. For π_{PGR}, the generative model G_ψ directly outputs transitions (s, a) that are high relevance. If we consider the synthetic replay as performing counterfactual reasoning generating the next gate action conditioned on the target energy, it effectively reduces the depth of the required search. The agent only needs to refine the pre-existing circuit s_{TN}, requiring a constant depth ΔD independent of N. Thus $T_{\text{PGR}} \propto |\mathcal{A}|^{\Delta D} \in \text{poly}(N)$.

Combining the iteration count K and the episode length T, the total sample complexity is $\mathcal{S} = K \times T$. Taking the ratio,

$$\frac{\mathcal{S}_{\text{PGR}}}{\mathcal{S}_{\text{base}}} = \left(\frac{\log(1/\Delta E_\chi)}{\log N}\right) \times \left(\frac{|\mathcal{A}|^{\Delta D}}{|\mathcal{A}|^{\text{poly}(N)}}\right) = O\left(\frac{1}{|\mathcal{A}|^{\text{poly}(N)}}\right) = O\left(\frac{1}{N^{2\,\text{poly}(N)}}\right). \tag{25}$$

This is a super-exponentially decaying ratio. There exists a constant $\kappa > 1$ such that this ratio is strictly bounded above by κ^{-N} for sufficiently large N, which completes the proof.

B GenQAS warm-start from TensorRL-QAS

GenQAS builds on the *TensorRL (fixed)* configuration of TensorRL-QAS Kundu & Mangini (2026). This component provides a problem-informed quantum state initialization before reinforcement learning (RL) architecture search begins. In contrast to TensorRL (trainable) the tensor network-derived warm start circuit is not included in the RL observation and its parameters are not subsequently optimized by the agent. Instead, it is retained as a fixed state-preparation prefix, while GenQAS searches for an additional circuit suffix. Thus, the complete circuit at RL step t is

$$U_t = V_t U_{\text{TN}}, \tag{26}$$

where U_{TN} is the fixed tensor network initialization and V_t is the circuit constructed by the GenQAS agent up to time t. This design separates the physically motivated state preparation from the learned architecture refinement. In particular, the RL state encodes only V_t, whereas the environment evaluates the full circuit in Eq. equation 26. Consequently, GenQAS retains the reduced RL-state dimensionality and reduced trainable parameter set of TensorRL (fixed), while augmenting its real experience with synthetic transitions generated by the learned dynamics model.

B.1 MPS approximation of the target state

For a target Hamiltonian H, we first obtain an approximate ground state using DMRG White (1992); Schollwöck (2011); Orús (2014):

$$|\psi_{\text{MPS}}\rangle \approx \arg\min_{|\psi\rangle} \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle}. \tag{27}$$

The resulting state is represented as an N-site matrix product state,

$$|\psi_{\text{MPS}}\rangle = \sum_{i_1,\ldots,i_N} A_1^{i_1} A_2^{i_2} \cdots A_N^{i_N} \, |i_1 i_2 \cdots i_N\rangle \,, \tag{28}$$

where $i_k \in \{0, 1\}$ and the tensors $A_k^{i_k} \in \mathbb{C}^{d_{k-1} \times d_k}$ have maximum bond dimension χ. The boundary dimensions are $d_0 = d_N = 1$, and $d_k \leq \chi$ for interior bonds.

The bond dimension controls the entanglement that can be represented by the MPS and therefore determines the trade-off between classical preprocessing cost and warm-start accuracy. DMRG optimizes the local MPS tensors through iterative sweeps until the energy in Eq. equation 27 converges. We use the resulting MPS only to construct the initial circuit; the MPS itself is not passed to the GenQAS agent. TensorRL-QAS employed this procedure to generate problem-aware, compact starting circuits before RL-QAS refinement. Figure 4 contrasts the trainable and fixed TensorRL-QAS initializations; GenQAS adopts the fixed configuration shown in Fig. 4(b).

B.2 Variational MPS-to-circuit mapping

We compile $|\psi_{\mathrm{MPS}}\rangle$ into a shallow parameterized circuit U_{TN} acting on $|0\rangle^{\otimes N}$ Berezutskii et al. (2025); Schön et al. (2005); Ran (2020). The circuit follows a nearest neighbour brickwork layout compatible with linearly connected quantum devices Kandala et al. (2017). It comprises M two-qubit unitary blocks,

$$U_{\mathrm{TN}} = U_M U_{M-1} \cdots U_1, \qquad U_k \in \mathrm{U}(4), \tag{29}$$

where each U_k acts on an adjacent pair of qubits. We determine these blocks by maximizing the overlap between the target MPS and the prepared state, equivalently minimizing

$$\mathcal{L}(\{U_k\}_{k=1}^{M}) = 1 - \left| \langle \psi_{\mathrm{MPS}} | \, U_M \cdots U_1 \, |0\rangle^{\otimes N} \right|^2 . \tag{30}$$

The brickwork layout is a one-dimensional hardware-efficient ansatz (HEA) consisting of alternating layers of nearest-neighbour two-qubit operations. Its local connectivity makes it compatible with devices arranged in a linear or effectively one-dimensional topology and reduces the need for additional routing operations. However, the practical suitability of an HEA depends on both its depth and the entanglement properties of the target-state family: deep HEAs can exhibit barren plateaus, whereas shallow HEAs can remain trainable in favourable regimes, particularly for states obeying an area-law entanglement structure Leone et al. (2024). In GenQAS, the shallow brickwork circuit is not used as a generic variational ansatz initialized at random; rather, its two-qubit blocks are optimized to reproduce the DMRG-derived MPS. This tensor network-informed initialization provides a physically meaningful starting point for the subsequent RL-based circuit refinement.

B.3 Riemannian optimization on the unitary manifold

The optimization variables in Eq. equation 30 belong to the unitary manifold $\mathrm{U}(4)$:

$$U_k^\dagger U_k = I_4, \qquad k = 1, \ldots, M. \tag{31}$$

We therefore use a Riemannian version of Adam with Cayley retraction, as in Appendix B of TensorRL-QAS. Let $G_k = \nabla_{U_k} \mathcal{L}$ denote the Euclidean gradient obtained by automatic differentiation Baydin et al. (2018). For unitary matrices, the Riemannian gradient can be written as

$$\mathrm{grad}\, \mathcal{L}(U_k) = G_k - U_k G_k^\dagger U_k. \tag{32}$$

This expression lies in the tangent space of $\mathrm{U}(4)$ at U_k.

For each block, Riemannian Adam maintains first- and second-moment estimates Luchnikov et al. (2021),

$$m_k^{(t+1)} = \beta_1 m_k^{(t)} + (1-\beta_1)\, \mathrm{grad}\, \mathcal{L}(U_k^{(t)}), \tag{33}$$

$$v_k^{(t+1)} = \beta_2 v_k^{(t)} + (1-\beta_2) \left\| \mathrm{grad}\, \mathcal{L}(U_k^{(t)}) \right\|^2 , \tag{34}$$

followed by the bias-corrected tangent direction

$$V_k^{(t)} = -\eta_t \frac{\widehat{m}_k^{(t+1)}}{\sqrt{\widehat{v}_k^{(t+1)}} + \varepsilon}. \tag{35}$$

Here, η_t is the learning rate, β_1 and β_2 are the Adam decay parameters, and division by the scalar denominator is applied elementwise to the matrix-valued first moment.

To update the unitary while preserving Eq. equation 31, we apply Cayley retraction. Defining the skew-Hermitian matrix

$$W_k^{(t)} = V_k^{(t)} U_k^{(t)\dagger} - U_k^{(t)} V_k^{(t)\dagger}, \tag{36}$$

the updated block is

$$U_k^{(t+1)} = \left(I_4 - \frac{1}{2} W_k^{(t)}\right)^{-1} \left(I_4 + \frac{1}{2} W_k^{(t)}\right) U_k^{(t)}. \tag{37}$$

Because $W_k^{(t)}$ is skew-Hermitian, the Cayley map in Eq. equation 37 preserves unitarity up to numerical precision Luchnikov et al. (2021). The momentum variables are subsequently transported to the tangent space at the updated point before the next iteration. All two-qubit blocks are updated simultaneously until the overlap loss converges. The resulting MPS-derived brickwork circuit provides the warm-start circuit used by both TensorRL-QAS configurations in Fig. 4; GenQAS retains only the fixed variant in Fig. 4(b).

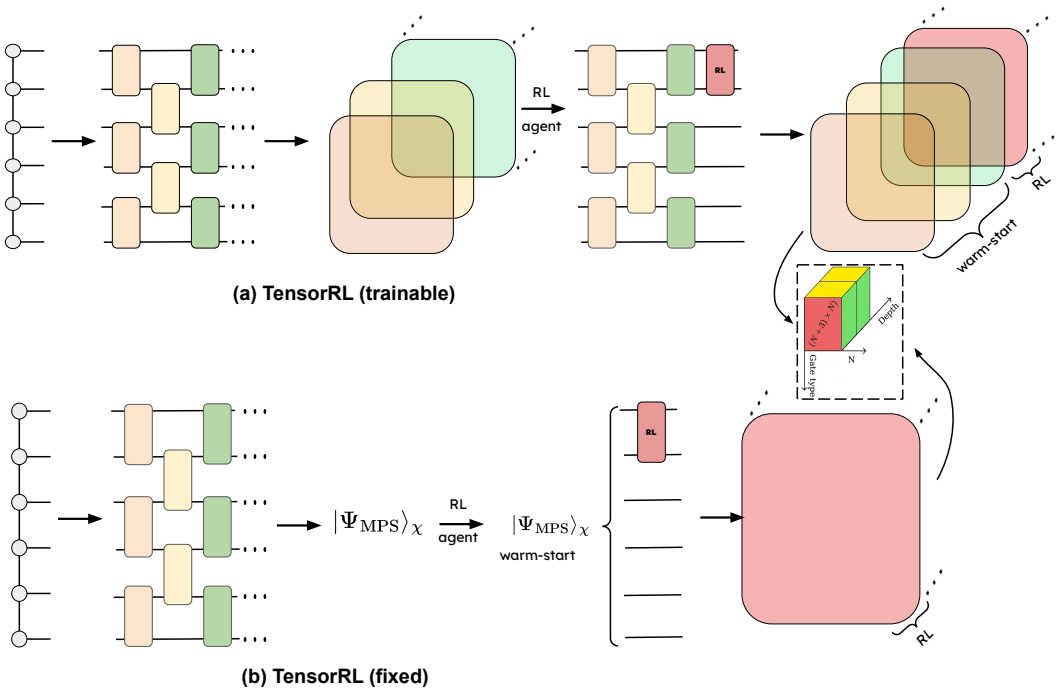


Figure 4: **Tensor network initializations in TensorRL-QAS.** (a) *TensorRL (trainable)*: the DMRG-derived MPS is mapped to a brickwork circuit and encoded in the RL state. The RL agent can subsequently modify the warm start circuit and optimize its parameters jointly with those of the newly appended gates. (b) *TensorRL-fixed TN-init*: the MPS-derived circuit prepares a fixed warm-start state $|\psi_{\mathrm{MPS}}\rangle_\chi$, while the RL state encodes only the circuit appended by the agent. The fixed initialization is excluded from the agent observation and remains non-trainable throughout RL search. GenQAS uses the TensorRL (fixed) configuration in (b), upon which the agent constructs the learnable circuit suffix V_t. Adapted from TensorRL-QAS Kundu & Mangini (2026).

B.4 Fixed initialization within GenQAS

After convergence of the mapping objective, the two-qubit unitaries are decomposed into the native search and execution gate set. As illustrated in Fig. 4(b), U_{TN} prepares the warm-start state but is excluded from the RL-state and is not modified by the GenQAS agent. In the TensorRL-QAS configuration adopted here, the compiled circuit is expressed in terms of single-qubit rotations and entangling gates, namely $\{\mathtt{RX}, \mathtt{RY}, \mathtt{RZ}, \mathtt{CNOT}\}$. The resulting circuit is denoted U_{TN} and remains unchanged throughout a GenQAS training run.

Specifically, the initial quantum state used by the environment is

$$|\psi_0\rangle = U_{\mathrm{TN}} |0\rangle^{\otimes N} . \tag{38}$$

At time t, an action selected by the agent appends a gate to the learnable suffix V_t. The energy, reward, and termination condition are evaluated using

$$E_t = \langle 0| U_{\mathrm{TN}}^{\dagger} V_t^{\dagger} H V_t U_{\mathrm{TN}} |0\rangle . \tag{39}$$

Crucially, the circuit representation presented to the agent contains only the appended architecture V_t. If the RL suffix has maximum depth D, the state dimension is therefore independent of the warm start depth D_{TN}. In contrast, a trainable TN initialization would encode both the TN prefix and the subsequently generated gates, increasing the representation from the suffix-only form to one that includes $D_{\mathrm{TN}} + D$ circuit layers. Retaining U_{TN} as a fixed prefix reduces the observation size, avoids optimizing its parameters during each RL episode, and maintains a lower-cost environment for the generative replay agent.

B.5 Relation to the GenQAS contribution

The tensor network warm start is inherited from TensorRL-QAS (fixed) and serves only as a fixed, problem informed initialization. GenQAS differs from TensorRL-QAS in the experience-learning mechanism after initialization: instead of training exclusively on transitions sampled from the real replay memory, GenQAS learns a transition-reward-termination model from real experience and uses it to generate synthetic transitions on demand.

Accordingly, the tensor network module affects the initial state distribution and the circuit evaluation prefix, whereas the generative replay module affects the distribution of transitions used for DDQN updates. The two components are complementary: U_{TN} reduces the burden of discovering a useful initial circuit from scratch, while model-based generative replay increases the effective density of training data in the remaining architecture search problem.

B.6 Warm-start resource cost and GenQAS refinement

Table 3 reports the resource requirements of the fixed MPS-derived warm-start circuits after decomposition into the native gate set $\{\mathtt{RX}, \mathtt{RY}, \mathtt{RZ}, \mathtt{CNOT}\}$. Here, $\mathrm{ROT} = N_{\mathrm{RX}} + N_{\mathrm{RY}} + N_{\mathrm{RZ}}$ denotes the total number of single-qubit rotations. The compiled warm-start circuits have a fixed depth of 27 across the considered systems, while the number of two-qubit gates increases from 15 CNOTs for 6-qubit BeH_2 to 42 CNOTs for the 15-qubit transverse field Ising model (TFIM).

Table 3: **Resource cost and energy error of the fixed MPS-derived warm-start circuits.** The MPS obtained by DMRG is mapped to a brickwork circuit and decomposed into the native gate set $\{\mathtt{RX}, \mathtt{RY}, \mathtt{RZ}, \mathtt{CNOT}\}$. ROT denotes the total number of single-qubit rotations. The initialization error is $\epsilon_{\mathrm{TN}} = |E_{\mathrm{TN}} - E_0|$, and $\epsilon_{\mathrm{GenQAS}}$ is the lowest error obtained by GenQAS.

System	ROT	CNOT	Depth	ϵ_{TN}	$\epsilon_{\mathrm{GenQAS}}$	R_ϵ
6-BeH_2	93	15	27	5.96×10^{-3}	2.3×10^{-5}	99.6%
8-H_2O	129	21	27	2.70×10^{-3}	1.2×10^{-3}	55.6%
10-H_2O	165	27	27	4.77×10^{-3}	4.1×10^{-4}	91.4%
12-H_2O	196	32	27	5.07×10^{-1}	2.2×10^{-2}	95.7%
15-TFIM	255	42	27	1.025	1.5×10^{-2}	98.5%

To quantify refinement beyond the fixed initialization, we define the warm-start and final GenQAS errors relative to the reference ground-state energy E_0 as

$$\epsilon_{\mathrm{TN}} = |E_{\mathrm{TN}} - E_0| , \qquad \epsilon_{\mathrm{GenQAS}} = |E_{\mathrm{GenQAS}} - E_0| . \tag{40}$$

The absolute error reduction achieved by GenQAS is

$$\Delta\epsilon = \epsilon_{\mathrm{TN}} - \epsilon_{\mathrm{GenQAS}}, \tag{41}$$

and the relative error reduction is

$$R_\epsilon = 100 \times \frac{\epsilon_{\mathrm{TN}} - \epsilon_{\mathrm{GenQAS}}}{\epsilon_{\mathrm{TN}}}. \tag{42}$$

GenQAS improves upon the fixed tensor network initialization across all evaluated systems. For example, for 12-qubit H_2O, GenQAS reduces the MPS-to-circuit mapping error from 5.07×10^{-1} to 2.2×10^{-2}. For the 15-qubit TFIM, GenQAS reduces the initialization error from 1.025 to 1.5×10^{-2}, corresponding to an absolute reduction of 1.010 and a relative reduction of 98.5%. These results demonstrate that the MPS-derived circuit provides a useful but not necessarily sufficient warm start, which is subsequently refined through RL-based architecture search.

C Supplementary for Methods Section 3

C.1 Action masking

A single qubit action on qubit q at layer m is illegal when the corresponding location is already occupied,

$$G_{m,q} = G_{m-1,q}. \tag{43}$$

A CNOT acting on qubit pair (q_1, q_2) is illegal when the same CNOT was applied in the preceding layer,

$$G_{m,(q_1,q_2)} = \texttt{CNOT} \ \wedge \ G_{m-1,(q_1,q_2)} = \texttt{CNOT}. \tag{44}$$

Actions violating maximum circuit depth or hardware connectivity constraints are masked in the same manner.

C.2 Full training algorithm

Algorithm 1 gives the complete GenQAS training procedure summarized in Section 3.4. It combines three components introduced separately in the main text: the fixed MPS warm start construction (Section 3.1), the DDQN circuit refinement loop, and the generative replay mechanism (Section 3.2).

Algorithm 1 GenQAS: Training with fixed MPS warm start and generative replay

Require: Hamiltonian H, MPS bond dimension χ, maximum RL depth D, replay memory $\mathcal{M}$, Q network Q_ϕ, target network Q_{ϕ^-}, generative model f_ψ, generation ratio G_r, synthetic replay frequency T_{syn}

Obtain $|\psi_{\text{MPS}}\rangle_\chi$ using DMRG for H
Map $|\psi_{\text{MPS}}\rangle_\chi$ to a brickwork circuit U_{TN} using Riemannian optimization
Initialize $\mathcal{M} \leftarrow \emptyset$ and $Q_{\phi^-} \leftarrow Q_\phi$
for each training episode **do**
    Initialize $V_0 \leftarrow I$ and RL state s_0
    for $t = 0, \ldots, D-1$ **do**
        Select a valid action a_t using ϵ greedy exploration
        Append the gate: $V_{t+1} \leftarrow a_t V_t$
        Evaluate $U_{t+1} = V_{t+1} U_{\text{TN}}$ and observe (r_t, s_{t+1}, d_t)
        Store $(s_t, a_t, r_t, s_{t+1}, d_t)$ in $\mathcal{M}$
        Sample real batch $\mathcal{B}_{\text{real}}$ from $\mathcal{M}$
        $\mathcal{B} \leftarrow \mathcal{B}_{\text{real}}$
        if $t \bmod T_{\text{syn}} = 0$ **then**
            Sample $G_r|\mathcal{B}_{\text{real}}|$ seed pairs from $\mathcal{B}_{\text{real}}$
            Generate $\mathcal{B}_{\text{syn}} = \{(s, a, \hat{r}, \hat{s}', \hat{d})\}$ using $f_\psi(s, a)$
            $\mathcal{B} \leftarrow \mathcal{B}_{\text{real}} \cup \mathcal{B}_{\text{syn}}$
        end if
        Update Q_ϕ using $\mathcal{B}$
        if generative model update step **then**
            Update f_ψ using only real transitions from $\mathcal{M}$
        end if
        if target network update step **then**
            Set $Q_{\phi^-} \leftarrow Q_\phi$
        end if
        if $d_t = 1$ **then**
            break
        end if
    end for
end for
return best circuit $U^\star = V^\star U_{\text{TN}}$

Lines 1 to 3: warm start and initialization. Lines 1 and 2 execute the tensor network pipeline described in Appendix B exactly once, before any RL interaction occurs. DMRG produces the MPS approximation $|\psi_{\mathrm{MPS}}\rangle_\chi$ of the target ground state, and the Riemannian optimization procedure of Appendix B.3 compiles it into the fixed brickwork circuit U_{TN}. This circuit is never modified again during training. Line 3 initializes an empty replay memory $\mathcal{M}$ and copies the randomly initialized online Q network Q_ϕ into the target network Q_{ϕ^-}, following standard DDQN practice Mnih et al. (2013).

Lines 4 to 5: episode and step initialization. Each training episode begins with an empty learnable suffix, $V_0 = I$, and the corresponding RL state s_0, which encodes the circuit before any agent action has been taken. The inner loop over $t = 0, \ldots, D-1$ constructs the circuit suffix one gate at a time, up to the maximum RL depth D.

Lines 6 to 9: environment interaction. At each step, the agent selects a valid action a_t under ϵ-greedy exploration over the masked action space described in Appendix F. The selected gate extends the suffix, $V_{t+1} \leftarrow a_t V_t$, and the environment evaluates the full circuit $U_{t+1} = V_{t+1} U_{\mathrm{TN}}$, which includes the fixed warm start prefix. This evaluation returns the reward r_t, the next RL state s_{t+1}, and the termination flag d_t. Because U_{TN} is fixed, this evaluation step is the only point in the algorithm where a genuine quantum classical circuit evaluation occurs; every other operation in the loop is purely classical.

Line 10: real transition storage. The resulting real transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ is stored in the persistent replay memory $\mathcal{M}$. This is the only mechanism by which $\mathcal{M}$ grows; synthetic transitions, introduced below, are never added to it.

Lines 11 to 12: real batch sampling. A batch $\mathcal{B}_{\mathrm{real}}$ is sampled from $\mathcal{M}$ according to the prioritized replay distribution described in Section 3.2, and initializes the update batch $\mathcal{B}$ used for this step's DDQN update.

Lines 13 to 17: on demand synthetic generation. Every T_{syn} steps, the algorithm augments the update batch with synthetic transitions. It samples $G_r|\mathcal{B}_{\mathrm{real}}|$ state action seeds from the just sampled real batch, applies the learned transition model f_ψ to each seed to obtain a predicted $(\hat{r}, \hat{s}', \hat{d})$, and forms the synthetic batch $\mathcal{B}_{\mathrm{syn}}$. The union $\mathcal{B}_{\mathrm{real}} \cup \mathcal{B}_{\mathrm{syn}}$ becomes the batch used for the Q network update on this step, matching the generation ratio mixing rule of Eq. equation 5. Restricting this augmentation to every T_{syn} steps, rather than every step, bounds the additional classical cost per training step and gives the transition model time to be refreshed between generation events, which is handled separately in line 19.

Line 18: Q network update. The online network Q_ϕ is updated using the mixed batch $\mathcal{B}$, following the Huber loss and DDQN target defined in Eq. equation 55. Because $\mathcal{B}$ can contain both real and synthetic transitions, this is the step at which generative replay directly influences the learned value function.

Lines 19 to 21: generative model refresh. On a separate, less frequent schedule than the Q network update, the transition model f_ψ is refreshed using only real transitions sampled from $\mathcal{M}$, as specified by T_{gen} in Section 3.2. This separation between the generation schedule (T_{syn}) and the model refresh schedule (T_{gen}) prevents the model from training on its own previously generated outputs, which would otherwise risk compounding model error over the course of training.

Lines 22 to 24: target network update. The target network Q_{ϕ^-} is synchronized with the online network Q_ϕ at a fixed interval, following standard DDQN practice. This update is independent of both the synthetic replay schedule and the generative model refresh schedule.

Lines 25 to 27: episode termination. The inner loop terminates early if the environment signals $d_t = 1$, indicating that the target criterion, described per benchmark in Appendix F.4, has been reached before the maximum depth D is exhausted. Otherwise, the loop runs for the full D steps.

Line 29: return value. After all training episodes, the algorithm returns the best discovered circuit $U^\star = V^\star U_{\mathrm{TN}}$, where $V^\star$ is the suffix associated with the lowest energy error observed across all episodes and seeds for that benchmark, consistent with the reporting convention used in Table 1.

D Detailed Results

This appendix collects the detailed material supporting the Results section: the full circuit statistics across all evaluated replay configurations, the signal density argument motivating the fixed budget scaling results, the generation ratio diagnostics assessing training stability, and the extended generalization experiment details for the 15 qubit transverse field Ising model and the 4 qubit Clifford synthesis task.

D.1 Full circuit statistics across all replay configurations

Table 1 in the main text reports, for each system, only the best error configuration of TensorRL-QAS, GenQAS, and HEA, alongside UCCSD, so that the comparison remains compact. Here we report the complete set of evaluated configurations, including both TensorRL-QAS replay strategies, all four GenQAS generation ratios, and all three HEA depths, for BeH_2 at 6 qubits and H_2O at 8, 10, and 12 qubits. This allows a direct check of which configuration was selected as the best error representative in the main text table, and shows the full accuracy resource trade-off surface traced out by G_r.

Across all four systems, the general pattern is consistent with the main text summary. Increasing G_r tends to reduce CNOT count and circuit depth at the cost of an increased rotation count, since the agent compensates for fewer entangling operations with additional single qubit rotations, and the relationship between G_r and final error is non-monotonic, with an intermediate ratio typically minimizing error at smaller system sizes and a higher ratio becoming competitive or superior at 12 qubits. TensorRL-QAS with uniform replay is consistently at least as accurate as, and often more accurate than, TensorRL-QAS with PER on these molecular benchmarks, which is why uniform replay is used as the TensorRL-QAS representative throughout the main text.

D.2 Signal density argument for generative replay

This appendix expands the signal density argument used to interpret the scaling results in Section 4.1. The argument connects Theorems 3.1 and 3.2, which characterize the sparsity of useful transitions under passive exploration, to the mixed update distribution defined in Eq. 66.

Let $\rho_{\mathrm{real}}(N)$ denote the expected fraction of transitions in the real replay buffer that lie within the ϵ-accurate set $\mathcal{G}_\epsilon$ defined in Theorem 3.1, after the agent has been initialized at the MPS derived warm start state s_{TN}. Because the warm start already lies within trace distance δ of the ground state by Lemma 3.4, early exploration proceeds from a region substantially closer to $\mathcal{G}_\epsilon$ than a uniformly random initialization would allow. Nonetheless, as the agent explores the suffix space V_t under an ϵ-greedy policy, the accessible fraction of ϵ-accurate circuit states relative to the full reachable suffix space still follows the same qualitative scaling established in Theorems 3.1 and 3.2: it decreases with increasing N at a rate governed by the Hilbert space dimension 2^N and, for the multi-term case, by the number of local Hamiltonian terms M and locality k.

Let $\rho_{\mathrm{syn}}(N)$ denote the corresponding fraction of generated transitions that lie in $\mathcal{G}_\epsilon$, produced by the learned transition model f_ψ from real state action seeds. Because synthetic transitions are seeded from real state action pairs already present in replay memory, rather than sampled uniformly over the full state space, $\rho_{\mathrm{syn}}(N)$ is not subject to the same uniform sampling argument used in Theorems 3.1 and 3.2. Instead, $\rho_{\mathrm{syn}}(N)$ depends on two factors: the fraction of real seeds that are themselves informative, and the fidelity of f_ψ in predicting the local consequences of an action taken from an informative seed. When the transition model achieves low prediction error on the transition, reward, and termination heads, as measured empirically in Section D.3, generated transitions preserve the local structure of their real seeds, so that $\rho_{\mathrm{syn}}(N)$ remains close to the fraction of informative real seeds rather than decaying at the same rate as $\rho_{\mathrm{real}}(N)$ would under unconditioned sampling.

The mixed update distribution has density

$$\rho_{\mathrm{mix}}(N) = (1 - G_r)\rho_{\mathrm{real}}(N) + G_r \rho_{\mathrm{syn}}(N). \tag{45}$$

Table 4: **Full circuit statistics for BeH_2 (6 qubits) and H_2O (8, 10, and 12 qubits), extending Table 1.** For each configuration we report two-qubit gate count (CNOT), single-qubit rotation count (ROT), circuit depth, and error for the best seed. HEA errors are taken from the best of 10 random seeds per ansatz depth (1, 2, 3 layers); UCCSD statistics are from a single deterministic run.

System	Method	CNOT	ROT	Depth	Error
6-qubit BeH_2	TensorRL-QAS (Uniform)	9	9	9	7.22×10^{-5}
	TensorRL-QAS (PER)	16	3	14	7.89×10^{-5}
	GenQAS ($G_r = 0.2$)	14	5	14	7.62×10^{-5}
	GenQAS ($G_r = 0.4$)	5	2	6	7.18×10^{-5}
	GenQAS ($G_r = 0.6$)	5	10	5	2.37×10^{-5}
	GenQAS ($G_r = 0.8$)	4	12	6	7.59×10^{-5}
	HEA (1-layer)	6	24	10	5.92×10^{-3}
	HEA (2-layer)	12	36	18	1.98×10^{-3}
	HEA (3-layer)	18	48	26	3.37×10^{-4}
	UCCSD	272	620	475	5.92×10^{-3}
8-qubit H_2O	TensorRL-QAS (Uniform)	14	6	8	1.24×10^{-3}
	TensorRL-QAS (PER)	14	2	11	1.26×10^{-3}
	GenQAS ($G_r = 0.2$)	12	2	10	2.22×10^{-3}
	GenQAS ($G_r = 0.4$)	3	2	4	1.19×10^{-3}
	GenQAS ($G_r = 0.6$)	7	10	10	1.22×10^{-3}
	GenQAS ($G_r = 0.8$)	2	13	8	1.20×10^{-3}
	HEA (1-layer)	8	32	12	2.63×10^{-3}
	HEA (2-layer)	16	48	22	2.63×10^{-3}
	HEA (3-layer)	24	64	32	2.96×10^{-3}
	UCCSD	1312	2596	2148	2.63×10^{-3}
10-qubit H_2O	TensorRL-QAS (Uniform)	15	17	17	4.15×10^{-4}
	TensorRL-QAS (PER)	23	14	19	4.17×10^{-4}
	GenQAS ($G_r = 0.2$)	4	5	7	4.17×10^{-4}
	GenQAS ($G_r = 0.4$)	14	15	20	3.92×10^{-4}
	GenQAS ($G_r = 0.6$)	19	19	20	3.93×10^{-4}
	GenQAS ($G_r = 0.8$)	26	36	24	4.32×10^{-4}
	HEA (1-layer)	10	40	14	4.69×10^{-3}
	HEA (2-layer)	20	60	26	4.84×10^{-3}
	HEA (3-layer)	30	80	38	1.14×10^{-2}
	UCCSD	3440	5932	5336	4.69×10^{-3}
12-qubit H_2O	TensorRL-QAS (Uniform)	18	2	6	2.22×10^{-2}
	TensorRL-QAS (PER)	17	2	11	2.48×10^{-2}
	GenQAS ($G_r = 0.2$)	14	2	10	2.33×10^{-2}
	GenQAS ($G_r = 0.4$)	18	2	4	2.28×10^{-2}
	GenQAS ($G_r = 0.6$)	3	10	10	2.45×10^{-2}
	GenQAS ($G_r = 0.8$)	2	4	2	2.44×10^{-2}
	HEA (1-layer)	12	48	16	2.52×10^{-2}
	HEA (2-layer)	24	72	30	3.29×10^{-2}
	HEA (3-layer)	36	96	44	6.31×10^{-2}
	UCCSD	6976	10628	10356	2.51×10^{-2}

If $\rho_{\mathrm{syn}}(N)$ decays more slowly in N than $\rho_{\mathrm{real}}(N)$, which is the qualitative behaviour we expect when the transition model remains accurate on the seed distribution actually encountered during training, then $\rho_{\mathrm{mix}}(N)$ decays more slowly than $\rho_{\mathrm{real}}(N)$ alone for any fixed $G_r > 0$. This gives a qualitative account of the results in Section 4.1: as N increases and $\rho_{\mathrm{real}}(N)$ falls toward its sparsity limited regime, increasing G_r compensates by weighting the update batch toward the more slowly decaying $\rho_{\mathrm{syn}}(N)$ term, which is consistent with the empirically observed shift in the optimal G_r from 0.4 at 6 to 10 qubits to 0.8 at 12 qubits.

We emphasize that this argument is a qualitative account of the observed scaling behaviour, not a closed form derivation of $\rho_{\mathrm{syn}}(N)$ from first principles. Deriving $\rho_{\mathrm{syn}}(N)$ analytically would require characterizing the joint distribution of real state action seeds under a trained ϵ-greedy policy and the generalization error of f_ψ on that distribution, both of which depend on the training dynamics of DDQN and are not tractable in closed form for the general Hamiltonians considered here. The empirical loss diagnostics in Section D.3 are intended as a direct, measured proxy for the behaviour of $\rho_{\mathrm{syn}}(N)$ in place of this analytical derivation, since low and stable transition, reward, and termi-

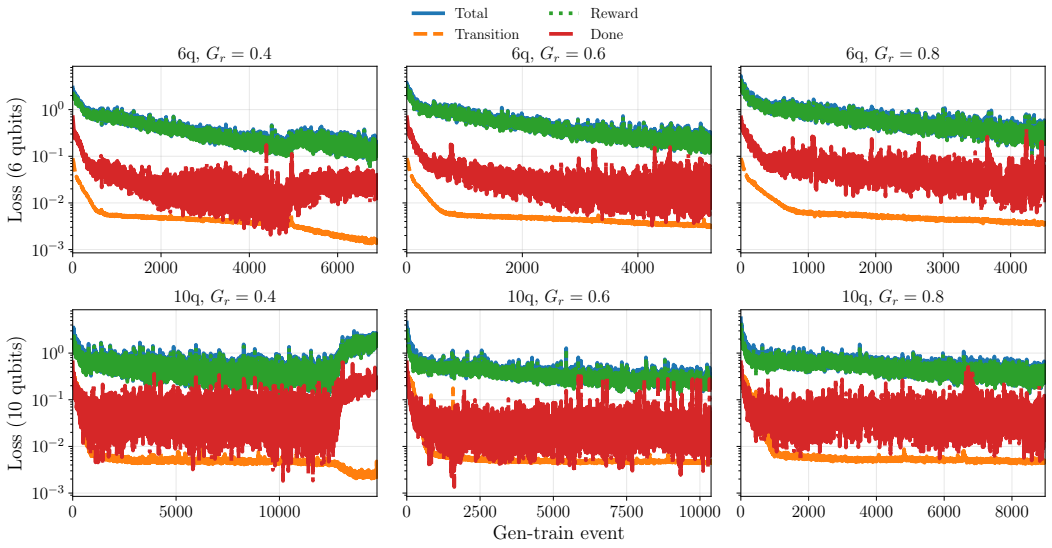


Figure 5: **Generator loss decomposition at 6 and 10 qubits**. Total, transition, reward, and termination losses for $G_r \in \{0.4, 0.6, 0.8\}$. At 10 qubits, $G_r = 0.4$ exhibits a late increase in total loss, while $G_r = 0.6$ and $G_r = 0.8$ remain more bounded.

nation losses are a necessary condition for $\rho_{\text{syn}}(N)$ to remain close to the informative fraction of its real seeds.

D.3 Generation ratio diagnostics

Synthetic replay is beneficial only when the learned transition model remains sufficiently accurate for stable value learning. We examine generator losses and weighted DDQN Huber losses at 6 and 10 qubits for $G_r \in \{0.4, 0.6, 0.8\}$ to assess whether the dynamics model and value function remain numerically well behaved under different synthetic fractions; these diagnostics do not directly measure the density of low energy synthetic transitions.

At 6 qubits, transition loss decreases substantially for all G_r, reaching values near 10^{-3}, with no sustained divergence in the reward or termination components. At 10 qubits, $G_r = 0.6$ and $G_r = 0.8$ keep the total loss bounded, while $G_r = 0.4$ develops a late increase driven by the reward and termination heads, indicating that stability depends on the evolving state distribution and reward landscape, not only on G_r.

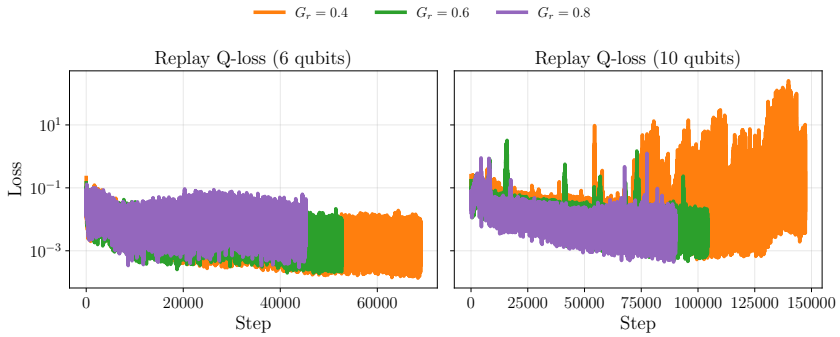


Figure 6: **Weighted DDQN Huber loss under different generation ratios**. At 6 qubits all configurations remain bounded after initial training. At 10 qubits, $G_r = 0.4$ exhibits late high variance loss spikes, while $G_r = 0.6$ and $G_r = 0.8$ remain comparatively stable.

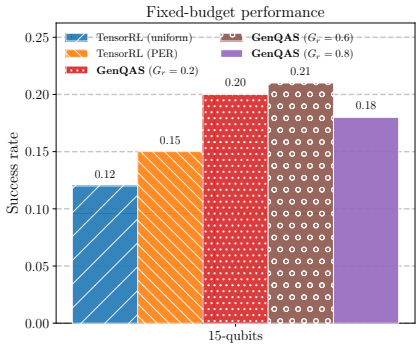
At 10 qubits, $G_r = 0.4$ develops large late stage Q loss spikes, while $G_r = 0.6$ and $G_r = 0.8$ remain more stable. This identifies a generation ratio trade off: a moderate synthetic fraction yields strong final success probability at moderate system sizes, while a larger synthetic fraction provides a more stable critic loss trajectory later in training, and at larger system sizes. Together, these diagnostics show that on demand synthetic replay can be incorporated into DDQN training without systematic divergence, and that G_r should be treated as a task dependent control parameter rather than a fixed constant.

D.4 Generalization experiment details

In the following we investigate how GenQAS performs beyong quantum chemistry task and evaluate it on 2 separate kinds of quantum optimization problems: (1) to prepare the ground state of a 12-qubit transverse field Ising model (TFIM) and (2) 4-qubit clifford synthesis task where the aim is to prepare a target Clifford gateset using fiscrete gate library.

D.4.1 15 qubit transverse field Ising model

To test whether the benefit of generative replay extends beyond molecular ground state preparation, we evaluate GenQAS on a 15 qubit transverse field Ising model (TFIM). Unlike the molecular benchmarks, the TFIM Hamiltonian is not derived from a chemistry package and does not have a fixed active space structure; it instead provides a spin model ground state target with a qualitatively different locality structure, since every term acts on at most two neighbouring qubits rather than on the more delocalized fermionic excitation structure underlying the molecular Hamiltonians. This makes the TFIM a useful check on whether GenQAS's benefits are specific to electronic structure problems or extend to a broader class of local Hamiltonians covered by Theorem 3.2.



(a) Fixed-budget performance.

Method	Error	CNOT	ROT
TensorRL (uniform)	1.53×10^{-2}	10	8
TensorRL (PER)	1.54×10^{-2}	10	8
GenQAS ($G_r = 0.2$)	1.53×10^{-2}	7	10
GenQAS ($G_r = 0.4$)	1.53×10^{-2}	10	6
GenQAS ($G_r = 0.6$)	1.54×10^{-2}	14	6
GenQAS ($G_r = 0.8$)	1.53×10^{-2}	3	13

(b) Circuit statistics.

Figure 7: 15-qubit TFIM performance: fixed-budget success (a) and best circuit statistics (b).

We use the same MPS derived warm start and DDQN backbone as the molecular experiments. All methods, TensorRL-QAS with uniform replay, TensorRL-QAS with PER, and GenQAS across $G_r \in \{0.2, 0.4, 0.6, 0.8\}$, are trained under the same fixed episode budget and share identical action spaces, reward construction, and warm start circuit, so that the comparison isolates the effect of the replay mechanism, consistent with the molecular experiments in Section 4.3.

Figure 7a reports the final success probability under this matched training budget. GenQAS matches or exceeds the success probability of both passive TensorRL-QAS baselines across every tested

G_r. The best passive configuration, PER, reaches a success probability of 15%, while GenQAS with $G_r = 0.6$ reaches 21%, a relative improvement of approximately 40%. This is a smaller absolute margin than observed on the 12 qubit H_2O benchmark, which is consistent with the TFIM Hamiltonian's shorter range interactions producing a less severe sample starvation regime at this system size than the more delocalized molecular Hamiltonians, though both settings show the same qualitative direction of improvement.

Table 7b reports the corresponding circuit statistics for the best seed of each method. The final energy error is nearly identical across all six configurations, ranging only from 1.53×10^{-2} to 1.54×10^{-2}, indicating that the replay mechanism used does not measurably affect the achievable solution quality on this benchmark; the differences between methods are concentrated entirely in the circuit resources required to reach that quality. Uniform and PER TensorRL-QAS both require 10 CNOTs and 8 rotations. GenQAS with $G_r = 0.4$ matches this CNOT count exactly while reducing the rotation count from 8 to 6, a 25% reduction at equal entangling gate cost and equal error. GenQAS with $G_r = 0.2$ reduces CNOT count to 7 at the cost of 2 additional rotations relative to $G_r = 0.4$. The most aggressive reduction in entangling gates occurs at $G_r = 0.8$, which identifies a circuit using only 3 CNOTs, less than a third of the passive baselines, at comparable error, though this comes at the cost of the highest rotation count observed for this benchmark, 13, indicating that the agent substitutes single qubit expressivity for two qubit connectivity as the synthetic replay fraction increases. GenQAS with $G_r = 0.6$, which achieves the highest success probability in Figure 7a, uses the largest CNOT count among the GenQAS configurations, 14, but the smallest rotation count, 6, suggesting that the generation ratio maximizing fixed budget success probability is not necessarily the same generation ratio that minimizes circuit resources for this benchmark.

Overall, the TFIM results reinforce two conclusions from the main text. First, the improvement from generative replay is not restricted to chemistry derived Hamiltonians; it also holds for a spin Hamiltonian with a distinct locality structure, supporting the generality of the mechanism described by Theorem 3.2. Second, the generation ratio G_r continues to function as a practical control parameter over the accuracy resource trade-off rather than a setting with one universally optimal value, since the G_r that yields the highest success probability here, 0.6, differs from the values that were most effective on the molecular benchmarks in Section 4.1 and Section 4.3.

D.4.2 4 QUBIT CLIFFORD SYNTHESIS

The agent constructs a circuit exactly matching a target Clifford operation using the discrete gate library $\mathcal{A}_{\text{Cliff}} = \{\mathtt{H}_q, \mathtt{S}_q, \mathtt{S}_q^\dagger, \mathtt{CNOT}_{c,t}\}$, $q \in \{0, \ldots, N-1\}$, $c \neq t$, with reward determined by exact target matching and no variational parameter optimization. Target generation, action space construction, and evaluation protocol are given in Appendix E.

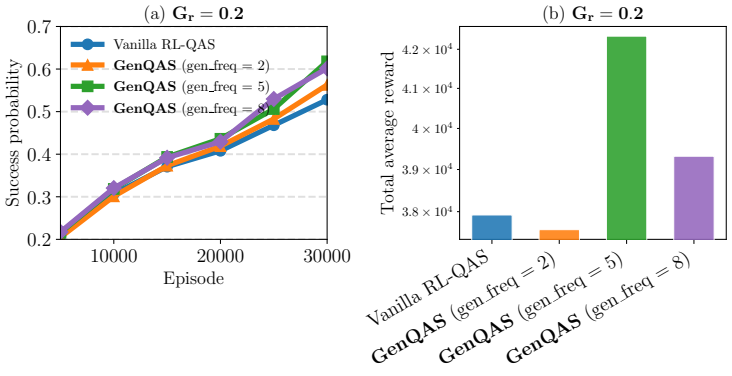


Figure 8: **4-qubit Clifford synthesis.** (a) A small generation ratio ($G_r = 0.2$) is sufficient to match or exceed the passive RL-QAS baseline in this discrete, non-variational setting, isolating the effect of the conditional generative model without a quantum classical optimization loop. (b) Cumulative reward under varying generator update frequency; an intermediate frequency ($\mathtt{gen_freq} = 5$) achieves the highest average cumulative reward.

D.5 Comparison with learning based QAS methods

The main text compares GenQAS against HEA and UCCSD, the two ansatz families most widely used in quantum chemistry and physics applications, and against TensorRL-QAS, the direct ablation isolating the effect of generative replay. Here we additionally compare GenQAS against other learning based QAS methods reported in the literature, including reinforcement learning based approaches (vanilla RL-QAS Ostaszewski et al. (2021), curriculum RL-QAS (CRLQAS) Patel et al. (2024), random agent QAS (RA-QAS) Kundu et al. (2024)), a non-RL search based approach (simulated annealing QAS (SA-QAS) Lu et al. (2023)), a differentiable search approach (quantumDARTS Wu et al. (2023)), a training free proxy metric approach (TF-QAS He et al. (2024)), and a fixed tensor network initialization followed by standard variational optimization without RL refinement (TN-VQE). Values for prior methods are reported as in Kundu & Mangini (2026), evaluated on the same three molecular systems used by GenQAS; for compactness we report a single `Gates` column combining CNOT and single qubit rotation counts.

Table 5: **Comparison with other learning based QAS methods on BeH_2 (6 qubits) and H_2O (8 and 10 qubits).** `Gates` reports the combined CNOT and single qubit rotation count. GenQAS attains the lowest error at 6 and 8 qubits and a competitive error at 10 qubits, while using substantially fewer total gates and, in most cases, a shallower circuit than every listed method. Values for prior methods are as reported in Kundu & Mangini (2026). NA indicates data not reported by the original method. All numbers reported in the table are the best performing statistics given in the respective papers and are adopted directly from prior work rather than reproduced independently, since results can vary across hardware and implementation details..

System	Method	Error	Depth	Gates
6-qubit BeH_2	RA-QAS Kundu et al. (2024)	5.8×10^{-4}	21	43
	TF-QAS He et al. (2024)	1.8×10^{-3}	NA	57
	SA-QAS Lu et al. (2023)	5.6×10^{-3}	36	73
	TN-VQE Kundu & Mangini (2026)	5.4×10^{-3}	10	20
	GenQAS ($G_r = 0.6$)	2.37×10^{-5}	**5**	**15**
8-qubit H_2O	CRLQAS Patel et al. (2024)	1.8×10^{-4}	75	140
	Vanilla RL-QAS Ostaszewski et al. (2021)	1.7×10^{-4}	96	165
	quantumDARTS Wu et al. (2023)	3.1×10^{-4}	64	219
	RA-QAS Kundu et al. (2024)	1.1×10^{-3}	56	128
	SA-QAS Lu et al. (2023)	2.6×10^{-3}	40	95
	TN-VQE Kundu & Mangini (2026)	2.7×10^{-3}	10	20
	GenQAS ($G_r = 0.4$)	1.19×10^{-3}	**4**	**5**
10-qubit H_2O	CRLQAS Patel et al. (2024)	3.4×10^{-4}	114	183
	Vanilla RL-QAS Ostaszewski et al. (2021)	2.5×10^{-4}	73	128
	RA-QAS Kundu et al. (2024)	9.9×10^{-4}	80	210
	SA-QAS Lu et al. (2023)	4.7×10^{-3}	24	56
	TN-VQE Kundu & Mangini (2026)	1.1×10^{-3}	26	65
	GenQAS ($G_r = 0.4$)	3.92×10^{-4}	**20**	**29**

E Clifford synthesis agent and environment details

We evaluate GenQAS on a discrete 4-qubit Clifford synthesis task in which the agent constructs a circuit that exactly implements a target Clifford operator. This experiment complements the molecular and TFIM benchmarks by removing variational parameter optimization and energy-expectation evaluation. The target is instead specified by an exact Clifford transformation, and success is determined through exact equality of the synthesized and target Clifford operators.

Our formulation is conceptually related to prior reinforcement learning approaches to Clifford circuit synthesis, which also frame synthesis as a sequential decision problem over a Clifford gate set and use stabilizer tableau representations of the target operator Kremer et al. (2024). However, our environment is independently implemented and differs in several important respects. In particular, we construct circuits forward from the identity, represent both the current and target Clifford operators in the observation, use QASM-defined target circuits, include the tableau phase information, and train the GenQAS DDQN agent with prioritized generative replay.

E.1 TARGET CIRCUITS AND ACTION SPACE

Target Clifford circuits are loaded from QASM files organized by qubit number and circuit depth category and generated using the QCircuitBench dataset Yang et al. (2026). At the beginning of each episode, one target circuit is sampled uniformly from the dataset entries. The target circuit is converted to a Qiskit `Clifford` object, denoted C_{tar}, while the agent starts from the identity circuit with Clifford representation $C_0 = I$. The agent sequentially appends gates to the current circuit until the synthesized Clifford operator exactly equals C_{tar} or the maximum synthesis horizon $T_{\max}$ is reached. We use $T_{\max} = 20$ in the reported Clifford synthesis experiments. The action space contains

$$\mathcal{A}_{\mathrm{Cliff}} = \left\{ \mathtt{H}_q,\ \mathtt{S}_q,\ \mathtt{S}_q^\dagger,\ \mathtt{CNOT}_{c,t} \right\}, \qquad q \in \{0, \ldots, N-1\}, \quad c \neq t. \tag{46}$$

The resulting action-space size is

$$|\mathcal{A}_{\mathrm{Cliff}}| = 3N + N(N-1). \tag{47}$$

For the 4-qubit benchmark, this gives 24 possible actions: 12 single-qubit Clifford actions and 12 directed `CNOT` actions.

We apply local action masking to suppress immediately repeated or cancelling gate sequences on the same qubit. In particular, the environment masks consecutive identical H, S, or $S^\dagger$ actions on a qubit, masks consecutive S and $S^\dagger$ actions on the same qubit, and masks a repeated CNOT with the same ordered control-target pair. All actions are masked once the maximum episode length is reached. These constraints reduce locally redundant trajectories while retaining the full Clifford synthesis objective.

E.2 STATE REPRESENTATION

A Clifford operator on N qubits is represented using the stabilizer tableau and a phase vector obtained by the Qiskit `Clifford` class. Let $\mathsf{T}(C)$ denote the binary tableau representation of a Clifford operator C, and let $\mathbf{p}(C)$ denote the associated phase vector. We define the vectorized Clifford representation as

$$\mathbf{v}(C) = \left[\mathrm{vec}(\mathsf{T}(C)), \mathrm{vec}(\mathbf{p}(C)) \right]. \tag{48}$$

Unlike representations that omit phase information, the present environment retains both tableau and phase entries because exact operator equality, including phase, defines successful synthesis. At time step t, the observation supplied to the agent consists of the target Clifford representation, the current Clifford representation, and a normalized progress variable:

$$s_t = \left[\mathbf{v}(C_{\mathrm{tar}}), \mathbf{v}(C_t), \frac{t}{T_{\max}} \right]. \tag{49}$$

This construction makes the target transformation, the operator synthesized so far, and the remaining effective episode horizon available to the DDQN agent.

E.3 REWARD AND TERMINATION

To provide dense feedback before exact synthesis is achieved, we compare the current and target Clifford tableaux entrywise. Let T_t, $\mathbf{p}_t$ and $\mathsf{T}_{\mathrm{tar}}$, $\mathbf{p}_{\mathrm{tar}}$ denote the tableau and phase vector of the current and target Clifford operators, respectively. The tableau-match fraction is defined as

$$m_t = \frac{\sum_i \mathbb{I}\left[\mathsf{T}_{t,i} = \mathsf{T}_{\mathrm{tar},i}\right] + \sum_j \mathbb{I}\left[p_{t,j} = p_{\mathrm{tar},j}\right]}{|\mathsf{T}_t| + |\mathbf{p}_t|}. \tag{50}$$

The associated tableau-distance proxy is

$$d_t = 1 - m_t. \tag{51}$$

The reward after applying action a_t is

$$r_t = -\lambda_{\mathrm{step}} - \lambda_{\mathrm{CX}} \mathbb{I}[a_t \in \mathrm{CNOT}] + \lambda_{\mathrm{partial}} \left(m_{t+1} - m_t\right) + R_{\mathrm{solve}} \mathbb{I}[C_{t+1} = C_{\mathrm{tar}}], \tag{52}$$

where

$$\lambda_{\mathrm{step}} = 0.01, \qquad \lambda_{\mathrm{CX}} = 0.02, \qquad \lambda_{\mathrm{partial}} = 0.5, \qquad R_{\mathrm{solve}} = 10. \tag{53}$$

The first term penalizes long circuits, the second applies an additional cost to entangling operations, the third rewards local improvement in tableau agreement, and the final term provides a sparse terminal reward for exact Clifford synthesis.

An episode terminates when

$$C_t = C_{\text{tar}} \tag{54}$$

or when $t = T_{\max}$. Exact equality is evaluated using the Qiskit Clifford representation, rather than through an approximate state or unitary-distance threshold. Therefore, a successful episode corresponds to exact synthesis of the target Clifford operation within the allowed number of construction steps.

F GenQAS implementation details

F.1 DDQN backbone.

We employ a Double Deep Q-Network (DDQN) Mnih et al. (2013), parameterized by $Q_\phi(s, a)$, for problems up to 8 qubits, and an n-step trajectory rollout variant Sutton et al. (1998) with $n = 5$ for larger problems. The Q network is a multilayer perceptron. The agent follows an ϵ-greedy policy with ϵ decaying from 1.0 to $\epsilon_{\min} = 0.05$. For a transition $(s_t, a_t, r_t, s_{t+1}, d_t)$, DDQN uses the online network to select the next action and a target network Q_{ϕ^-} to evaluate it,

$$y_t = r_t + \gamma(1 - d_t)Q_{\phi^-}(s_{t+1}, a^\star_{t+1}), \qquad a^\star_{t+1} = \arg\max_{a \in \mathcal{A}_{\text{valid}}} Q_\phi(s_{t+1}, a). \tag{55}$$

The online network is optimized using the Huber loss between $Q_\phi(s_t, a_t)$ and y_t with the Adam optimizer Kingma & Ba (2015). For variable depth search spaces, the per step discount factor is

$$\gamma = (\gamma_{\text{final}})^{1/D}, \tag{56}$$

so that the effective discount over the full circuit depth remains fixed. Full agent and environment hyperparameters are reported in this appendix.

F.2 Hyperparameters

GenQAS was implemented as a prioritized generative replay extension of a Double Deep Q-Network (DDQN) agent as discussed in the main text Sec 3. The agent used a five-layer multilayer perceptron with 1000 neurons per hidden layer, LeakyReLU activations, no dropout, and an ADAM learning rate of 3×10^{-4}. All experiments employed a replay memory of 20000 transitions and mini-batches of 1000 transitions. The target network was updated every 100 optimization steps. The replay buffer was prioritized, and the conditional generative transition model was trained every 10 steps using real replay transitions. Synthetic experience was generated on-the-fly and mixed with real transitions at generation ratios $G_r \in \{0.2, 0.4, 0.6, 0.8\}$; all other agent and environment settings were kept fixed within each system size. For the 10- and 12-qubit H_2O tasks, we used an n-step DDQN update with $n = 5$; the same $n = 5$ roll-out setting was used throughout the reported GenQAS experiments for consistency.

The exploration policy was initialized with $\epsilon = 1.0$ and decayed multiplicatively with factor 0.99995 until reaching $\epsilon_{\min} = 0.05$. We used a per-layer discount factor derived from a final discount value of $\gamma_{\text{final}} = 0.005$. Circuit construction was initialized from a tensor-network prior obtained using bond dimension $\chi = 2$, with no zero-parameter initialization. All noiseless results were obtained using exact expectation-value evaluations ($n_{\text{shots}} = 0$), without error mitigation or stochastic early termination. Molecular Hamiltonians were constructed using the Jordan-Wigner mapping with tapering enabled, and continuous circuit parameters were optimized after each architectural modification using COBYLA.

F.3 Computational resources

All experiments were executed on the DelftBlue high-performance computing system Delft High Performance Computing Centre (DHPC). The GenQAS training and simulation workloads used the `gpu-a100` partition, comprising nodes with two Intel Xeon Gold 6448Y processors (32 cores per processor), four NVIDIA Tesla A100 GPUs with 80 GB memory per GPU, 500 GB system

Table 6: Main GenQAS hyperparameters for the noiseless molecular benchmarks. The generation ratio G_r was swept over $\{0.2, 0.4, 0.6, 0.8\}$ while all remaining settings were held fixed for a given qubit number.

Hyperparameter	**6-qubit**	**8-qubit**	**10-qubit**	**12-qubit**
System	BeH_2	H_2O	H_2O	H_2O
Training episodes	4000	7000	5000	5000
Maximum circuit layers	47	47	107	71
Acceptance error	1.6×10^{-3}	1.6×10^{-3}	1.6×10^{-3}	2.54×10^{-2}
Generation ratio, G_r	$\{0.2, 0.4, 0.6, 0.8\}$	$\{0.2, 0.4, 0.6, 0.8\}$	$\{0.2, 0.4, 0.6, 0.8\}$	$\{0.2, 0.4, 0.6, 0.8\}$
Replay memory size	20000	20000	20000	20000
Batch size	1000	1000	1000	1000
Hidden layers	$[1000]^5$	$[1000]^5$	$[1000]^5$	$[1000]^5$
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Generator training frequency	10	10	10	10
Prioritized replay	Yes	Yes	Yes	Yes
n-step return	5	5	5	5
Target-network update	100	100	100	100
ϵ decay / minimum	0.99995/0.05	0.99995/0.05	0.99995/0.05	0.99995/0.05
Final discount, γ_{final}	0.005	0.005	0.005	0.005
TN init. / bond dim.	Yes / 2	Yes / 2	Yes / 2	Yes / 2
Optimizer	COBYLA	COBYLA	COBYLA	COBYLA

memory, and 1.5 TB local storage. Unless otherwise stated, each training run used 2 NVIDIA A100 GPU(s) and 4 CPU core(s). For some of the comparative analysis for GenQAS, we utilise a single node system equipped with NVIDIA RTX A6000. The molecular Hamiltonian construction, tensor network initialization, RL training, and Qulacs statevector simulations were executed using the computational resources allocated to each run.

The source code, complete hyperparameter configurations, random seeds, and instructions for reproducing the experiments will be made available in the associated code repository upon publication.

F.4 Molecular Hamiltonian construction

We generated the molecular electronic-structure Hamiltonians using the quantum chemistry module of PennyLane (`qml.qchem`) Bergholm et al. (2018); Arrazola et al. (2021). For each molecule, we specify the atomic symbols, Cartesian nuclear coordinates, total charge $Q = 0$, spin multiplicity $\text{mult} = 1$, and an active-space definition characterized by the number of active electrons and active orbitals. The resulting second-quantized electronic Hamiltonian was mapped to a qubit Hamiltonian using the Jordan-Wigner transformation,

$$H = \sum_j c_j P_j, \tag{57}$$

where $c_j \in \mathbb{R}$ are Pauli coefficients and $P_j \in \{I, X, Y, Z\}^{\otimes N}$ are N-qubit Pauli words. The Pauli decomposition $\{(c_j, P_j)\}_j$ was stored and used as the target Hamiltonian in the variational energy objective,

$$E(\boldsymbol{\theta}, \mathcal{A}) = \langle 0|U^\dagger(\boldsymbol{\theta}, \mathcal{A}) H U(\boldsymbol{\theta}, \mathcal{A})|0\rangle, \tag{58}$$

where $\mathcal{A}$ denotes the circuit architecture generated by the RL agent.

We considered BEH_2, and H_2O molecules. For BEH_2, the nuclear geometry was specified as H$(0, 0, -1.330)$-Be$(0, 0, 0)$-H$(0, 0, 1.330)$, with four active electrons in three active orbitals. For H_2O, we used the geometry H$(-0.021, -0.002, 0.000)$, O$(0.835, 0.452, 0.000)$, and H$(1.477, -0.273, 0.000)$, with four active electrons. The standard H_2O instances used four active orbitals; larger active-space instances used the `6-31g` basis to support the corresponding higher 10-qubit Hamiltonians. Unless a larger H_2O active space was required, molecular integrals were calculated in the `STO-3G` basis.

Our implementation used PennyLane for molecular Hamiltonian generation, PennyLane NumPy for differentiable array handling, and SciPy linear algebra routines for exact diagonalization of the Hamiltonian matrix. We additionally constructed an equivalent `SparsePauliOp` representation using Qiskit Quantum Information Javadi-Abhari et al. (2024). This enabled an independent consistency check between the PennyLane and Qiskit representations: we verified the agreement of

the dense Hamiltonian matrices and their ground-state eigenvalues before storing each Hamiltonian. The final data file contained the dense Hamiltonian matrix, its eigenvalue spectrum, the Pauli strings, the associated coefficients, and a zero energy shift. This validation ensures that the Pauli Hamiltonian used by the RL-QAS environment is consistent with the molecular Hamiltonian generated by the quantum chemistry workflow.

F.5 Transverse field Ising Hamiltonian construction

We generated the 15-qubit transverse field Ising model (TFIM) Hamiltonian directly as a weighted sum of Pauli strings, rather than through a quantum chemistry package, since the model is defined natively on a qubit lattice rather than derived from a molecular electronic structure problem. For N qubits with open boundary conditions, the Hamiltonian is

$$H = -J \sum_{i=1}^{N-1} Z_i Z_{i+1} - h \sum_{i=1}^{N} X_i, \tag{59}$$

where J sets the nearest-neighbour coupling strength and h sets the transverse field strength. As in the molecular case, we stored the resulting Pauli decomposition $\{(c_j, P_j)\}_j$ with $P_j \in \{I, X, Y, Z\}^{\otimes N}$ and used it directly as the target Hamiltonian in the variational energy objective, with $\mathcal{A}$ denoting the circuit architecture generated by the RL agent.

We fixed $J = 1$ and set $h = 1$, placing the model at its quantum critical point. At this coupling ratio, the ground state exhibits the largest correlation length and the highest two-body concurrence Odavić et al. (2023) in the phase diagram, rather than the more weakly entangled ground states found in the ordered ($h \ll J$) or disordered ($h \gg J$) regimes. We selected this regime deliberately: a highly entangled ground state is harder to prepare with a shallow circuit and therefore constitutes a more demanding QAS target than a Hamiltonian whose ground state is close to a product state, making $h = 1$ a meaningful stress test for GenQAS beyond the more weakly entangled regimes.

As with the molecular Hamiltonians, we constructed an equivalent `SparsePauliOp` representation using Qiskit Quantum Information Javadi-Abhari et al. (2024) and cross-checked it against the directly assembled Pauli decomposition of Eq. 59. For $N = 15$, exact diagonalization was performed using sparse eigensolvers (SciPy `eigsh`) rather than dense diagonalization, since the $2^{15} \times 2^{15}$ Hamiltonian matrix is only tractable in sparse form at this system size. We verified agreement between the sparse ground state energy and the DMRG-derived ground-state energy used to construct the MPS warm start (Appendix B) before storing the Hamiltonian. The final data file contained the Pauli strings, coefficients, the sparse ground-state eigenvalue, and a zero energy shift, consistent with the format used for the molecular Hamiltonians.

F.6 Quantum simulators

Although the molecular Hamiltonians were generated using PennyLane, the tensor-network warm-start component inherited from TensorRL-QAS was implemented using Qiskit Javadi-Abhari et al. (2024) and Quimb Gray (2018). In particular, Quimb was used to construct and optimize the matrix product state (MPS) approximation of the target ground state with bond dimension $\chi = 2$. The resulting MPS was subsequently converted into a quantum circuit representation using the TensorRL-QAS warm-start procedure, implemented through Qiskit. This circuit provides the physics-informed initial architecture for the RL agent before sequential gate level architecture search begins.

All quantum state simulations and energ expectation value evaluations reported in this work were performed using the Qulacs Suzuki et al. (2021). For a circuit architecture $\mathcal{A}$ and variational parameters $\boldsymbol{\theta}$, Qulacs prepares the state

$$|\psi(\boldsymbol{\theta}, \mathcal{A})\rangle = U(\boldsymbol{\theta}, \mathcal{A}) \, |0\rangle^{\otimes N}, \tag{60}$$

and evaluates the molecular energy as

$$E(\boldsymbol{\theta}, \mathcal{A}) = \langle \psi(\boldsymbol{\theta}, \mathcal{A})| \, H \, |\psi(\boldsymbol{\theta}, \mathcal{A})\rangle = \sum_j c_j \, \langle \psi(\boldsymbol{\theta}, \mathcal{A})| \, P_j \, |\psi(\boldsymbol{\theta}, \mathcal{A})\rangle \, . \tag{61}$$

Here, $H = \sum_j c_j P_j$ is the Jordan-Wigner-mapped Pauli Hamiltonian. Qulacs was used consistently throughout the RL-training loop.

F.7 MECHANISTIC SIGNAL-DENSITY ANALYSIS

This section provides a mechanistic interpretation of how the generation ratio controls the effective density of useful transitions during GenQAS training. The analysis is conditional on the assumptions of Theorems 3.1 and 3.2, Lemma 3.4, and Theorem 3.5. It does not assert that every generated transition is useful; rather, it identifies the conditions under which synthetic replay can compensate for the decreasing density of useful real transitions in large QAS problems.

Let $\mathcal{G}_\epsilon$ denote the set of circuit states with energy within ϵ of the ground state energy E_0. For a replay distribution $\mathcal{D}$, we define the useful-transition density as

$$\rho(\mathcal{D}) = \Pr_{\tau\sim\mathcal{D}}\left[s(\tau) \in \mathcal{G}_\epsilon\right], \tag{62}$$

where $s(\tau)$ denotes the circuit state associated with transition τ. This quantity measures the probability that a transition sampled for a DDQN update originates from the target low-energy region.

For passive replay, the replay buffer contains only transitions collected through interaction with the QAS environment. Let $\rho_{\text{real}}(N)$ denote the useful-transition density in this distribution for an N qubit problem. Under the weakly informed exploration and Hamiltonian assumptions used in Theorems 3.1 and 3.2, this density satisfies

$$\rho_{\text{real}}(N) \le \exp\left[-c\,2^N \log\left(\frac{1}{\epsilon}\right)\right], \qquad c > 0. \tag{63}$$

Thus, as the number of qubits increases, useful transitions constitute a rapidly decreasing fraction of the real interaction history. Uniform replay and prioritized experience replay can alter the sampling frequency of previously observed transitions, but neither can change the probability that a useful transition was collected initially.

At each DDQN update, GenQAS constructs a batch of total size B by mixing real transitions sampled from the prioritized replay memory with synthetic transitions generated on demand. The generation ratio $G_r \in [0, 1)$ denotes the synthetic fraction of the total update batch:

$$B_{\text{real}} = \lfloor (1 - G_r)B \rfloor, \qquad B_{\text{syn}} = B - B_{\text{real}}. \tag{64}$$

Consequently, $G_r = 0.4$ produces an update batch containing approximately 60% real and 40% synthetic transitions, while $G_r = 0.8$ produces approximately 20% real and 80% synthetic transitions. The synthetic samples are generated from state action seeds drawn from real replay memory and are not stored persistently. The replay buffer therefore remains a record of real environment interactions.

Let $\rho_{\text{syn}}(N)$ denote the useful transition density among synthetic samples. Lemma 3.4 shows that the MPS derived warm start can be prepared within trace distance δ of the relevant low energy manifold under the stated MPS approximation and circuit mapping conditions. Let $q \in [0, 1]$ denote the probability that the learned local transition model preserves this useful neighborhood when generating a synthetic transition. Under this model alignment condition,

$$\rho_{\text{syn}}(N) \ge q(1 - \delta). \tag{65}$$

The factor q separates the fidelity of the MPS derived initialization from the local predictive accuracy of the learned transition model. Accordingly, Eq. equation 65 is a theoretical condition, not a guarantee that follows automatically from neural network training. The generator and critic diagnostics in Figures 5 and 6 assess whether learning remains stable for the investigated generation ratios.

The useful-transition density of the mixed GenQAS update distribution is

$$\rho_{\text{mix}}(N) = (1 - G_r)\rho_{\text{real}}(N) + G_r\rho_{\text{syn}}(N). \tag{66}$$

Combining Eqs. equation 63 and equation 65 gives

$$\rho_{\text{mix}}(N) \ge (1 - G_r)\rho_{\text{real}}(N) + G_r q(1 - \delta). \tag{67}$$

The corresponding gain relative to passive replay is

$$\begin{aligned}\frac{\rho_{\text{mix}}(N)}{\rho_{\text{real}}(N)} &\ge (1 - G_r) + G_r q(1 - \delta)\frac{1}{\rho_{\text{real}}(N)} \\ &\ge (1 - G_r) + G_r q(1 - \delta)\exp\left[c\,2^N \log\left(\frac{1}{\epsilon}\right)\right].\end{aligned} \tag{68}$$

Equation equation 68 describes the proposed mechanism of generative replay. If the MPS warm start remains sufficiently accurate, such that δ is small, and the learned dynamics model preserves local useful-transition structure with nonzero probability q, the synthetic contribution does not vanish with system size. This contrasts with the useful density of passive replay, which decreases rapidly under the assumptions of Eq. equation 63.

Finally, consider a regime in which a policy requires K useful replay transitions to achieve a fixed reduction in value-estimation error. The expected number of sampled replay transitions required to obtain these transitions scales approximately as

$$\mathbb{E}[M] \approx \frac{K}{\rho}. \tag{69}$$

Therefore,

$$\frac{\mathbb{E}[M_{\mathrm{real}}]}{\mathbb{E}[M_{\mathrm{mix}}]} \approx \frac{\rho_{\mathrm{mix}}(N)}{\rho_{\mathrm{real}}(N)}. \tag{70}$$

Together, Eqs. equation 68 and equation 70 provide an intuitive connection between useful-transition densification and the episode complexity result in Theorem 3.5. The finite-size results in Figure 2a do not establish an asymptotic scaling law, but they are consistent with the predicted increase in the relative value of generative replay as the passive replay signal becomes sparse.