GENEALOGY-CONDITIONED QUANTUM ARCHITECTURE SEARCH WITH SYNTHETIC LINEAGE PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantum architecture search (QAS) can automate the design of parameterized quantum circuits (PQCs), but testing an architecture typically requires an expensive inner-loop parameter optimization. Surrogate models can screen candidates before this evaluation, yet training the surrogate often requires the same quantum-evaluated architecture labels that it is meant to save. We introduce Genealogy-Conditioned Quantum Architecture Search (GCQAS), which pretrains a reusable offspring-ranking model without executing quantum circuits or collecting expert labels. GCQAS constructs synthetic lineages, assigns each family a different randomized scoring rule, and trains a contextual Transformer to infer how candidate structure, inheritance, and ancestral outcomes jointly determine local rankings. Because the rule differs across lineages, the model cannot memorize one and is forced to learn to infer the active rule from the ancestors' outcomes in its context. During a target search, the model remains frozen and uses the contextual information from candidate circuit's genealogy to predict the performance of each offspring in a genetic algorithm. We evaluate the same surrogate on distribution matching and variational quantum eigensolver under noiseless and noisy simulation, with target circuits of up to 16 qubits despite pretraining only on six-qubit lineages. GCQAS continues to guide search across this change in scale and objective, whereas a genetic algorithm with no surrogate, a genetic algorithm with in-context Bayesian optimization as a surrogate, and an online QAS surrogate do not achieve comparable solutions under the evaluation budget. GCQAS also improves on task-specific adaptive construction in the corresponding benchmark regimes. These results show that synthetic lineage pretraining can learn a transferable search strategy: task-general knowledge is stored in the model, while sparse target-specific evidence is supplied through the evolving context.

1 INTRODUCTION

Quantum circuits express quantum computations as ordered operations on qubits, followed by measurements to obtain task-relevant outcomes. Parameterized quantum circuits (PQCs) make a part of this program trainable; a classical optimizer adjusts continuous gate parameters using outputs from repeated circuit executions. This quantum-classical hybrid framework underlies applications including molecular-energy estimation, quantum simulation, generative modeling, and analysis of high-energy-physics data (Peruzzo et al., 2014; Liu & Wang, 2018; Peng et al., 2025; Delgado & Hamilton, 2022). A central design choice is the circuit architecture—which gates are used, where they act, and how they are connected. PQC design should balance representational capacity with the trainability: An overly restrictive architecture can represent only a subset of the state space. Conversely, adding layers to expand its representational capacity can lead to an optimization landscape where hardware noise accumulates and gradients vanish (Du et al., 2022; McClean et al., 2018; Wang et al., 2021).

Quantum architecture search (QAS) treats the task of finding a balanced PQC design as an optimization problem. As finding a balanced structure by hand becomes difficult when the depth and width of circuits grows, QAS has employed the weight sharing strategy (Du et al., 2022), differentiable relaxations (Wu et al., 2023), reinforcement learning (Kundu & Mangini, 2025), Bayesian

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.

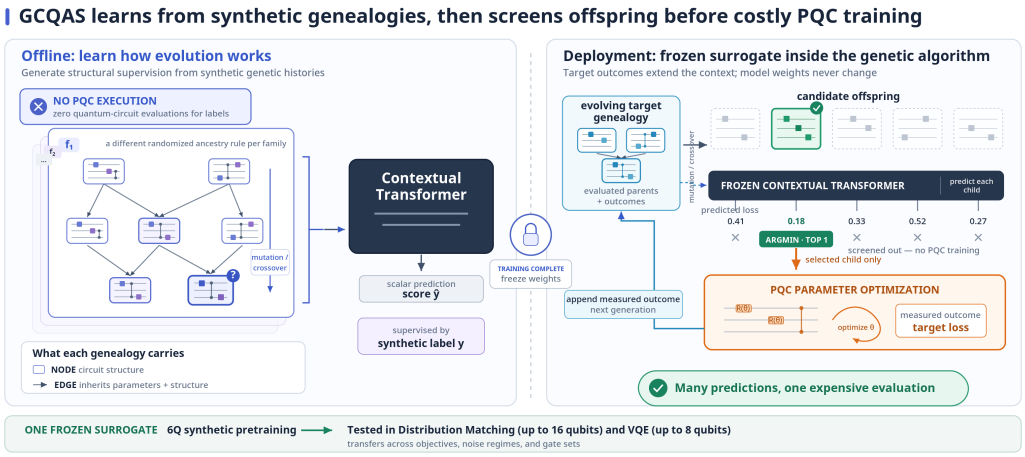


Figure 1: Overview of GCQAS. A contextual Transformer is pretrained on synthetic six-qubit genealogies labeled by independently randomized ancestry-dependent scoring rules, requiring no PQC execution. During target search, the frozen model uses previously evaluated ancestors to predict offspring losses, and only the best-predicted candidate undergoes PQC parameter optimization. The same surrogate transfers across distribution-matching and VQE tasks with different circuit sizes, objectives, noise regimes, and gate sets.

optimization (Situ et al., 2026a). The evolutionary algorithm can be a useful tool for performing QAS on a discrete search space (Ding & Spector, 2022); it maintains a population of circuit descriptions and explores them through selection, crossover, and mutation. However, its flexibility comes from a substantial evaluation burden; assigning genuine fitness to an offspring requires the classical optimization of the parameters embedded in a circuit and then measuring (or simulating) its performance, repeated across candidates and generations (Jin, 2005; Situ et al., 2026a).

Surrogate-assisted evolution has been introduced as a method to reduce the evaluation burden by learning the approximate fitness function and selecting promising candidates for evaluation (Jin, 2005; Zhang et al., 2018). However, training a surrogate can itself be costly when architecture–performance labels are obtained through quantum evaluations. Thus, neural predictors for gate sequences (Zhang et al., 2021), circuit graph representations (He et al., 2023), and ZX-calculus representations (Li et al., 2025) depend on quantum-evaluated corpora, while self-supervised and semi-supervised variants are based on evaluated seed examples (He et al., 2023; Situ et al., 2026b). Training-free methods replace learned predictors with predefined structural, expressibility, hardware, or kernel-based proxies (He et al., 2024b; Anagolum et al., 2024; Dao et al., 2026). Online surrogates adapt to the target task, but gather evaluation data anew for each search (Soloviev et al., 2024). Meta-transfer methods reuse previously optimized circuits, moving the cost to a collection of solved quantum tasks (He et al., 2024a). These approaches therefore retain a dependence on either quantum evaluation data or domain-informed proxy design, both of which can be costly to obtain for new quantum computing tasks.

To address these challenges, we introduce Genealogy-Conditioned Quantum Architecture Search (GCQAS). The key idea of GCQAS is to separate a task-general inference procedure from task-specific knowledge acquisition. Synthetic pretraining teaches the surrogate how candidate structure, ancestry, and observed outcomes can be combined to rank related offspring. A new quantum task then supplies its domain-specific information through the verified scores collected during search. To train this behavior, we generate synthetic circuit lineages and assign a different randomized scoring rule to each lineage. The surrogate has to infer the active rule from context rather than memorize one circuit-to-performance mapping. At deployment, its weights remain frozen while verified target outcomes update its context. The surrogate selects which proposals receive full evaluation, but the quantum objective remains the only source of population fitness.

Our contributions are:

- We formulate QAS surrogate pretraining as contextual offspring selection and construct synthetic circuit lineages that provide supervision without quantum simulation, quantum hardware, or previously solved target tasks.
- We develop GCQAS, a frozen contextual surrogate that combines candidate statistics, parent–child relations, and verified search history to select offspring while leaving final fitness decisions to the task objective.
- We evaluate the method on distribution matching and variational quantum chemistry across circuit widths, connectivity assumptions, and noiseless and noisy simulation.
- We will release the synthetic-data generator, trained surrogate, search code, and experiment configurations upon acceptance.

2 Related Work

Quantum circuit design and state preparation. QAS automates circuit design through weight sharing, differentiable gate selection, evolution, or reinforcement learning (Du et al., 2022; Wu et al., 2023; Ding & Spector, 2022; Patel et al., 2024). ACLBM grows Born-machine circuits through operator-pool gradients, while TensorRL combines molecular ansatz search with tensor-network initialization (Li & Cheng, 2025; Kundu & Mangini, 2025). These approaches organize search around target-task evaluations or optimization. AQER instead guides state-loading circuit construction by reducing target-state entanglement (Zhang et al., 2026); its state-fidelity objective differs from matching measurement probabilities or minimizing molecular energy, and does not assume an initially separable target.

Surrogates and evolutionary preselection. Surrogate-assisted evolution can screen offspring before expensive evaluation, while retaining measured rather than predicted fitness for population selection (Jin, 2005; Zhang et al., 2018). Quantum architecture predictors learn from evaluated circuits, with self-supervision reducing the label requirement (Zhang et al., 2021; He et al., 2023); training-free alternatives use structural or hardware-related proxies (He et al., 2024b; Anagolum et al., 2024). EDA-QAS instead fits a pairwise surrogate during each search (Soloviev et al., 2024). These choices require evaluated training data, predefined proxies, or target-time model fitting. Fitness inheritance already uses parent outcomes, but typically through a fixed aggregation rule rather than a learned interpretation of multi-generation ancestry (Smith et al., 1995).

Synthetic pretraining and contextual optimization. PFNs learn prediction from tasks sampled from a prior, and PFNs4BO uses the resulting frozen model for in-context Bayesian optimization (M"uller et al., 2022; Müller et al., 2023). Synthetic-function pretraining also supports transferable optimizers (Nguyen et al., 2023; Fadhel et al., 2026), while attention-based genetic algorithms learn population-level operators (Lange et al., 2023). These methods establish that search behavior can transfer without refitting a predictor on every target. However, generic input–output contexts and population representations do not explicitly encode which ancestors produced a candidate or what it inherited. This distinction motivates examining the structure of search history, not only the quantity of observations. Appendix B provides the detailed literature discussion.

3 Preliminaries

In this paper, we use *ansatz* or *architecture* for the discrete arrangement of quantum gates, and *parameters* for the continuous gate angles optimized after the architecture is chosen. We call an architecture produced by crossover or mutation an *offspring*. Its *genealogy* is the directed parent–child graph recorded by the search. We refer to running the task-specific parameter optimizer as a *full task evaluation*, and call the resulting objective value a *verified score*. A verified score comes from a simulator or quantum device, rather than from the surrogate.

3.1 Parameterized quantum circuits and verified evaluation

Let $a = (g_1, \ldots, g_L)$ denote a circuit architecture, where each g_ℓ specifies a gate type and its ordered qubit operands. Its parameterized gates carry $\theta \in \mathbb{R}^{d(a)}$, and the resulting unitary is denoted by

$U_a(\theta)$. The architecture a fixes the discrete computation and the dimension of its parameter space; optimizing θ searches within the family of states represented by that architecture.

Let $\mathcal{L}_{\mathcal{T}}(a, \theta)$ denote the loss for task $\mathcal{T}$. Given an initialization θ_0, optimizer $\mathcal{O}$, and parameter-update budget B_θ, the trained architecture score is

$$J_{\mathcal{T}}(a; \theta_0, \mathcal{O}, B_\theta) = \min_{0 \le b \le B_\theta} \mathcal{L}_{\mathcal{T}}(a, \theta_b), \qquad \theta_b = \mathcal{O}^b(a, \theta_0, \mathcal{T}). \tag{1}$$

The minimum retains the best inner-loop iterate. Throughout this paper, lower $J_{\mathcal{T}}$ is better. Because architecture quality depends on the optimizer, initialization, and budget, these quantities should be matched in a direct comparison. The benchmark-specific distribution-matching and VQE objectives are defined in Appendix C.

3.2 Genetic algorithms and genealogy

A genetic algorithm maintains a population $\mathcal{P}_t = \{a_{t,1}, \ldots, a_{t,N}\}$ at generation t. *Initialization* creates the first population, either randomly or from seed solutions. Each circuit's *genome* is its discrete gate or layer representation. Fitness-based *parent selection* favors better evaluated candidates; *elitism* retains selected strong candidates in the next population. *Crossover* combines regions from two parent genomes to form an offspring. *Mutation* makes a stochastic structural edit to a genome, including an output of crossover. A repair step enforces the circuit's gate, connectivity, and size constraints (Ding & Spector, 2022; Ma et al., 2024).

Parent–child links form a directed acyclic *genealogy*. We use *lineage* for a related circuit family or its recorded evolutionary history, not necessarily one chain of descent. For candidate c, $\mathcal{G}(c)$ is its ancestral graph, and $\mathcal{A}_t(c)$ is the set of verified ancestors available when it is proposed. Inheriting circuit structure does not require inheriting trained gate angles; parameter inheritance is an optional warm-start choice specified in Section 4. Appendix A gives a fuller explanation and the operator illustration in Figure 2.

4 Approach

4.1 Overview

GCQAS separates learning how to interpret evolutionary evidence from applying that knowledge to a quantum objective. The method has three stages. First, a generator produces synthetic evolutionary lineages containing circuit structures and parent–child relations. Second, each lineage receives an independently randomized scoring rule, and a contextual surrogate learns to infer enough of the active rule to rank related candidates. Finally, the frozen surrogate is inserted into a genetic QAS loop. It observes verified target-task outcomes as context but receives no target-time parameter update. Figure 1 summarizes this separation between synthetic pretraining and target-time verification.

The central design choice is to predict a local genetic decision rather than the absolute performance of an arbitrary circuit. Offspring generated from the same parents share much of their structure and optimization state. Their parents have also been evaluated already. GCQAS uses this local evidence to decide which child is worth the expensive inner optimization, while the quantum objective remains the only source of population fitness.

The method transfers neither a solved circuit nor parameters learned on another quantum task. It transfers a rule for interpreting a candidate relative to evaluated search history. Target observations can therefore adapt predictions through context even though the surrogate weights remain fixed.

4.2 Synthetic Dataset Generation

The first stage constructs families of related circuit genomes without executing a quantum circuit. Each family begins with random legal genomes. Descendants are then formed by randomly choosing parents and applying the same crossover, mutation, and repair rules used by the target-task search. Crucially, the synthetic generator does not sample parents according to fitness: its purpose is to expose the surrogate to diverse parent–child relations, not to imitate the population dynamics of

a solved task. Each record stores the candidate genome, parent identifiers, genealogy, and a 16-dimensional vector of circuit and inheritance statistics. The resulting dataset is entirely classical and requires neither quantum simulation nor expert labels.

The genome encodes raw strucure of circuits, layer by layer. Crossover retains the blocks layers of one parent while replacing a selected layer interval—or a layer-and-qubit region—with compatible gates from the other parent. Mutation replaces a selected region with newly sampled legal gates. The child is then simplified and repaired to satisfy the gate set, connectivity, and circuit-size constraints. By sharing this representation and these structural operators with the deployed search, the synthetic data preserve the meaning of the relations that the surrogate will later observe. They need not reproduce the same random choices or fitness-driven search trajectory.

Structure alone does not provide a useful learning target, so each synthetic family r is assigned its own randomized labeling rule. For query candidate c, let $\mathcal{V}^r(c) = \{c\} \cup \mathcal{A}^r(c)$ contain the query and the ancestors made available as context. Each node v has a candidate-aware vector $x_v \in \mathbb{R}^{16}$ containing circuit statistics, parent outcomes, and inheritance statistics. We draw a collection of weights $\{w_{r,h}\}$ indexed by relative genealogical position h and a bias b_r once for family r, and use that same rule to label every member of the family:

$$\widetilde{J}_r(c) = b_r + \sum_{v \in \mathcal{V}^r(c)} w_{r,h_c(v)}^\top x_v. \tag{2}$$

The query term $w_{r,0}^\top x_c$ distinguishes siblings with different circuit structures, while the remaining terms make their scores depend on shared ancestry. Parent-score fields are populated from synthetic outcomes already assigned earlier in the family.

Holding the rule fixed within a family makes observed ancestor outcomes informative about a new child. Redrawing it across families makes the same structural feature predict different outcomes in different contexts. This task-level data augmentation discourages the surrogate from memorizing one synthetic structure-to-score function; instead, it has to infer the active family rule from the labeled ancestry. The labels are not intended to approximate KL divergence, molecular energy, or any other target objective. They train the operation required at deployment: compare related candidates under a rule revealed only through context. In a proposal pool, siblings share their more distant ancestry, so contributions from that ancestry are common or nearly common and the local ordering can be easier to infer than absolute scores.

4.3 Contextual Surrogate Training

The surrogate is trained on query–context examples from synthetic families. Each query combines pooled circuit statistics with parent scores and ordered inheritance information; its context contains available ancestors and their synthetic outcomes. These inputs describe both the candidate and how it relates to its parents, rather than encoding a raw gate-token sequence. Appendix D.1 details the feature groups.

The model embeds the query and contextual input–outcome pairs, applies self-attention over the context, and predicts a scalar synthetic score from the query state. For a synthetic minibatch $\mathcal{B}$, its parameters are learned by

$$\mathcal{L}_{\text{syn}}(\phi) = \frac{1}{|\mathcal{B}|} \sum_{(H,\widetilde{J}) \in \mathcal{B}} \left(f_\phi(H) - \widetilde{J} \right)^2, \tag{3}$$

where H contains the query and its available synthetic ancestry. Feature normalization is fitted only on the synthetic training partition and stored with the checkpoint. Although Equation 3 trains a scalar regressor, the downstream search uses only its ordering within a related proposal pool. The checkpoint, feature transform, and prediction head are frozen before any target-task search begins.

4.4 Genetic PQC Search Aided by the Surrogate

At deployment, GCQAS initializes a population of legal circuits according to the task's initialization policy (Section 5) and obtains a verified score for every member by performing the full task evaluation in Equation 1. These evaluations establish both population fitness and the initial contextual

evidence. Subsequent generations retain the usual fitness-based selection of a genetic algorithm, but use the frozen surrogate to screen several possible structural edits before committing the expensive parameter-optimization budget.

For each open population slot j, the search selects parents using verified fitness and generates K legal alternatives with the same genetic operation, $\mathcal{C}_{t,j} = \{c_{t,j}^{(1)}, \ldots, c_{t,j}^{(K)}\}$. It constructs a contextual record $H_t(c)$ for every alternative and retains

$$\hat{c}_{t,j} = \arg\min_{c \in \mathcal{C}_{t,j}} f_\phi(H_t(c)) \,. \tag{4}$$

This pool-and-rank step is a specific design choice of GCQAS, rather than a defining component of genetic algorithms. It is applied first to crossover proposals and then, at the selected population positions, to mutation proposals. The complete generation proceeds as follows:

1. Copy the elites with their trained states and select breeding parents using verified fitness.
2. Generate and repair a pool of crossover alternatives for every remaining slot, rank them using Equation 4, and retain one.
3. At the selected population positions, generate mutation alternatives and apply the same contextual pool-and-rank procedure.
4. Parameter-optimize every member of the assembled population, including warm-started elite copies. Record the verified scores and parent relations for the next generation.

Only the retained offspring receive full task-level optimization. Their verified scores—never surrogate predictions—determine fitness and are appended to the genealogy available as context in later generations. When a parameterized gate in a retained child corresponds to one in a trained parent, its angle is copied as a warm start; newly introduced parameters follow the search initialization policy. A child can therefore reuse structural evidence through the surrogate and continuous optimization progress through parameter inheritance.

5 Experiment

We use the same pretrained surrogate checkpoint, with unchanged weights and feature normalization, for all 12 benchmark settings: nine distribution-matching targets and three molecular VQE settings. Only the evaluated circuits and their scores in the context change between tasks. The comparisons cover five baseline configurations for distribution matching and two TensorRL variants for VQE.

5.1 Benchmarks and baselines

Distribution matching. We use ACLBM's log-normal, bimodal, and triangular targets at 3 and 10 qubits, and Bars-and-Stripes (BAS) targets at 2×2, 3×3, and 4×4 (Li & Cheng, 2025). Preparing prescribed output probabilities supports generative modeling and quantum data loading. Smooth targets test probability allocation, whereas sparse BAS targets require structured correlations across qubits. Together, these tasks span 3–16 qubits. We measure forward KL divergence; Appendix C defines the objectives.

We compare GCQAS with constructive ACLBM, online-surrogate EDA-QAS (Soloviev et al., 2024), surrogate-free Pool GA, and PFNs4BO+GA with original or retrained weights (Müller et al., 2023). Pool GA shares the genetic operators and proposal-pool size but selects candidates uniformly. PFNs4BO+GA replaces the ranking model, retaining the evolutionary search. We evaluate both its original weights and a version retrained on the synthetic data used for GCQAS, testing whether access to that data alone closes the gap. These controls test the benefit of learned selection and the choice of surrogate; they do not isolate ancestry from all other differences between the models.

Variational molecular simulation. We additionally use TensorRL's six-qubit BeH_2 and eight-qubit H_2O Hamiltonians in noiseless simulation, and its eight-qubit H_2O shot-noise setting (Kundu & Mangini, 2025). VQE minimizes a molecular energy rather than matching output probabilities, testing transfer to a different scientific objective (Peruzzo et al., 2014). We report absolute energy

Table 1: KL divergence for distribution matching; lower is better. Entries report the best aggregate mean in the exported histories; ACLBM entries are published values. Values $\leq \epsilon_{\text{th}}$ reach the stopping threshold $\epsilon_{\text{th}} = 10^{-11}$. PFNs4BO+GA uses original (Orig.) or retrained (Retr.) weights. Bold marks the lowest values, including threshold ties.

Target	Qubits	ACLBM	Pool GA	EDA-QAS	PFNs4BO+GA		GCQAS
					Orig.	Retr.	
Log-normal	3	2.55×10^{-6}	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$
Bimodal	3	5.38×10^{-6}	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$
Triangular	3	2.20×10^{-6}	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$
Log-normal	10	3.25×10^{-4}	3.04×10^{-3}	2.04×10^{-2}	1.20×10^{-4}	1.98×10^{-3}	$\leq \epsilon_{\text{th}}$
Bimodal	10	3.95×10^{-4}	1.11×10^{-3}	5.07×10^{-3}	1.61×10^{-10}	9.85×10^{-4}	$\leq \epsilon_{\text{th}}$
Triangular	10	5.80×10^{-4}	1.86×10^{-3}	1.26×10^{-2}	4.33×10^{-9}	1.58×10^{-3}	$\leq \epsilon_{\text{th}}$
BAS 2×2	4	2.63×10^{-7}	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$	$\leq \epsilon_{\text{th}}$
BAS 3×3	9	8.09×10^{-6}	7.00×10^{-1}	1.94	1.01×10^{-7}	3.77×10^{-1}	$\leq \epsilon_{\text{th}}$
BAS 4×4	16	7.09×10^{-3}	4.52	6.50	3.54	4.72	$\mathbf{4.55 \times 10^{-3}}$

error in Hartree and compare with TensorRL's fixed and trainable tensor-network initialization variants. These provide the RL-based comparisons for VQE, whereas ACLBM represents a gradient based construction for distribution matching.

5.2 Protocol and reporting

The same surrogate remains frozen throughout evaluation. Synthetic generation and distribution-matching searches use the six-gate set $\{\mathrm{RX}, \mathrm{RZ}, \mathrm{SX}, \mathrm{X}, \mathrm{CZ}, \mathrm{RZZ}\}$, chosen to match gate types supported natively by IBM Heron processors, including the fractional RX and RZZ gates (IBM Quantum, 2026). VQE instead uses $\{\mathrm{RX}, \mathrm{RY}, \mathrm{RZ}, \mathrm{CNOT}\}$, following TensorRL (Kundu & Mangini, 2025). These sets share RX and RZ but differ in their other operations, including the entangling gates. VQE therefore tests transfer across gate sets as well as objectives, without retraining. Distribution searches stop when KL falls below 10^{-11}; such entries indicate the stopping threshold, not a measured numerical limit. The table reports the minimum over generations of the mean champion KL in each exported aggregate. We use published ACLBM values because we could not reproduce its reported KL values with the released implementation. TensorRL values also come from its paper.Aggregation conventions are detailed in Appendix D.

For BAS 3×3, Appendix E reports KL against generations and against a gradient-calculation budget. The latter accounts for ACLBM's candidate-action screening as well as parameter updates, avoiding a direct comparison of its epochs with evolutionary generations. It is a count-based comparison, not a measurement of total runtime.

VQE searches use a population of 20 for up to 500 generations, with COBYLA for noiseless optimization and Adam–SPSA for noisy optimization. Noiseless searches start from random circuits. For the noisy setting, we follow TensorRL's use of a matrix-product-state (MPS)-derived circuit to provide a physics-informed warm start (Kundu & Mangini, 2025). We initialize the population with this circuit and 19 mutations of it, concentrating the initial search around an approximate ground-state preparation rather than random circuits. Appendix D gives training settings, benchmark construction, optimizer rationale, and comparison conventions.

6 Results

6.1 Distribution matching

GCQAS reaches the stopping threshold on all three-, four-, nine- and ten-qubit smooth-distribution targets. On BAS 4×4, its best aggregate mean KL is 4.55×10^{-3}, achieving the best performance across the benchmarks. More broadly, a surrogate trained only on synthetic six-qubit circuit families is used without fine-tuning throughout the benchmark suite, including the 16-qubit distribution tasks. Its transfer across smooth, multimodal, and sparse combinatorial targets indicates that, over the

evaluated range, the learned selection rule is not confined to the circuit width or synthetic objective used during pretraining.

The small targets do not distinguish the selection methods: all evaluated search controls reach the stopping threshold on the three-qubit distributions and BAS 2×2. Pool GA, both PFNs4BO variants, and GCQAS already do so in the evaluated initial population. On the ten-qubit distributions and BAS 3×3, Pool GA retains substantially higher KL than GCQAS. Because Pool GA generates equally sized proposal pools but selects uniformly, these comparisons support the value of learned ranking beyond simply generating more alternatives. On BAS 3×3 (Figure 3), GCQAS reaches KL below 10^{-11} by generation 9, whereas Pool GA still has mean KL 0.70 at generation 20. The dotted ACLBM line is its published KL reference, not a generation trajectory.

The surrogate comparisons test different ways of using evaluated search history. PFNs4BO conditions jointly on an exchangeable set of input–output observations and predicts a distribution over a discretized output space (Müller et al., 2023). It does not explicitly represent which observations are parents, how circuit structure was inherited, or which subset forms the query's ancestry. GCQAS instead regresses a contextual score from typed parent–child relations and uses that score only for local offspring ranking. Original PFNs4BO performs substantially better than its retrained variant on the larger targets and achieves low KL on the ten-qubit bimodal and triangular tasks and BAS 3×3, although GCQAS reaches the stopping threshold on these tasks. The performance gap becomes apparent on wider targets: the methods tie at the stopping threshold on the three-qubit targets and BAS 2×2, but separate on the ten-qubit targets and larger BAS instances. GCQAS also reaches low KL earlier in the recorded searches. Its aggregate mean reaches the stopping threshold by generation 12 on all three ten-qubit distributions and generation 9 on BAS 3×3. Original PFNs4BO attains its best recorded mean at generations 19–20 on these targets, still above that threshold. Figure 3 shows the BAS 3×3 convergence curves. Retraining PFNs4BO on the synthetic corpus does not close the performance gap; the original weights perform better here, but converge more slowly than GCQAS. Thus, simply supplying the synthetic data to a different surrogate is insufficient in this comparison. This comparison motivates ancestry as useful context but does not isolate its causal effect from other surrogate differences. EDA-QAS also retains higher KL on these larger targets. Its online pairwise SVM uses fixed-size circuit encodings within a search space whose width, maximum depth, and gate choices should be specified in advance; it neither exploits an offline prior nor represents variable-depth genealogy (Soloviev et al., 2024). The comparison is consistent with the benefit of learning from richer provenance before the expensive target search begins.

GCQAS also improves on ACLBM across the corresponding distribution benchmarks. The BAS 3×3 gradient-budget comparison (Figure 4) reaches the stopping threshold at approximately 4.7×10^4 counted calculations, while the supplied ACLBM trajectory reaches KL 8.09×10^{-6} at $99{,}900$. GCQAS therefore obtains lower KL with less than half the counted budget in this comparison. This accounts for ACLBM's repeated operator-pool screening (Li & Cheng, 2025), but does not equate the runtime of different gradient calculations or include offline surrogate training.

Table 2: VQE energy error (Hartree), depth, CNOT count, and rotation count (ROT). Lower is better; bold denotes the lowest reported entry in each benchmark, including ties. †Fixed TensorRL counts exclude its initial state preparation, so resource minima are not end-to-end comparisons.

Benchmark	Method	Error	Depth	CNOT	ROT
BeH_2-6	TensorRL (fixed)†	6.8×10^{-5}	**7**	**5**	**12**
Noiseless	TensorRL (trainable)	6.0×10^{-5}	33	22	96
	GCQAS	$\mathbf{2.68 \times 10^{-6}}$	13	15	32
H_2O-8	TensorRL (fixed)†	8.9×10^{-4}	**6**	**9**	**15**
Noiseless	TensorRL (trainable)	2.0×10^{-4}	36	30	146
	GCQAS	$\mathbf{1.74 \times 10^{-4}}$	21	**9**	103
H_2O-8	TensorRL (fixed)†	1.5×10^{-4}	**3**	**4**	**1**
Shot noise	TensorRL (trainable)	8.7×10^{-5}	30	28	131
	GCQAS	$\mathbf{3.44 \times 10^{-5}}$	27	21	129

6.2 VARIATIONAL MOLECULAR SIMULATION

GCQAS records lower energy errors than both TensorRL variants on the three reported settings (Table 2). All three errors are below the 1.6×10^{-3}-Hartree chemical-accuracy reference. The noiseless results extend synthetic-lineage pretraining to molecular-energy optimization from random initial populations. The shot-noise result instead uses an MPS-seeded population, showing compatibility with a physics-informed starting point. These comparisons concern achieved errors under the stated protocols; they do not establish a matched-budget advantage or statistical significance.

GCQAS also reports lower depth and gate counts than trainable TensorRL in these cases. Fixed TensorRL reports smaller added circuits, but omits its fixed state-preparation circuit from those counts (Kundu & Mangini, 2025). The table therefore preserves the source's resource convention rather than interpreting these entries as total implementation costs.

7 CONCLUSIONS

We introduced GCQAS, a method for pretraining a quantum-architecture selection surrogate without expert annotations, previously solved quantum tasks, or quantum-evaluated architecture labels. Synthetic lineages teach the model to infer a local scoring rule from candidate statistics, parent–child relations, and contextual outcomes. The frozen model can then use verified observations from a new task without being refitted. This separates learning how to interpret search history from learning the physics of any one target.

Across distribution matching and molecular-Hamiltonian optimization, GCQAS transfers the same frozen surrogate across objective functions, connectivity assumptions, and both noiseless and noisy simulation. The target circuits scale to 16 qubits even though surrogate pretraining uses only simpler six-qubit synthetic lineages. This gap is central to the result: synthetic pretraining supplies a reusable procedure for interpreting evolutionary evidence, while verified target observations provide the task-specific information through context. The experiments therefore support GCQAS as a reusable search framework rather than a model fitted to one circuit width or benchmark. On BAS 3×3, the convergence curves show that learned selection improves both the attained KL and the generations needed relative to Pool GA and the PFNs4BO controls. The gradient-budget comparison also reaches lower KL than the supplied ACLBM trajectory with less than half the counted calculations. These results support learning a reusable selection model from synthetic data, without establishing which component of that model causes the gain.

8 FUTURE WORK

An immediate next step would be to evaluate GCQAS on quantum hardware. Our distribution experiments use exact statevector probabilities, and the VQE noise environments are simulated. These settings isolate architecture selection under the source benchmark protocols; hardware studies would test how the search responds to finite measurement budgets, calibration drift, and device constraints that are not fully represented in simulation. Such validation is needed to establish device-level performance.

Further work can also characterize the range of tasks over which the frozen surrogate transfers reliably. The current experiments transfer from six-qubit synthetic lineages with loop connectivity to 3–16-qubit circuits, including all-to-all VQE connectivity. Testing larger circuits and additional gate sets and connectivity graphs would establish where this transfer holds and where the synthetic training distribution needs to be extended. Because synthetic lineage objectives teach contextual inference rather than quantum physics, these studies must continue to verify selected circuits through target-task evaluation.

Beyond distribution matching and the molecular-energy benchmarks studied here, future works could also cover larger molecular problems, including ammonia, and other scientific objectives. A broader direction would be to apply synthetic-lineage pretraining to low-data scientific search in which experiments or high-fidelity simulations are costly and candidate relationships provide useful context. Establishing this extension will require task-specific validation under different evaluation costs and noise conditions

AI USE STATEMENT

In this work, we used generative AI tools to assist with literature search, manuscript revision, and code review. The research idea, method, experiments, and drafts were developed by the authors. All AI-assisted revisions were reviewed and edited by the authors, who verified the citations and take responsibility for the final text, claims, and artifacts.

ETHICS STATEMENT

GCQAS aims to lower the cost of exploring quantum circuits for scientific tasks. Its surrogate requires no circuit-evaluation data during pretraining and is reused across tasks without retraining. Together with the observed search efficiency, this may reduce the resources needed to investigate quantum algorithms, broaden participation in quantum-computing research, and support scientific discovery. The noisy VQE simulations provide evidence relevant to noise-aware circuit design, but do not establish practical quantum advantage or performance on quantum hardware.

As a general-purpose search method, GCQAS could also support unintended or harmful downstream applications; its demonstrated scope is distribution matching and molecular energy estimation. Environmental benefits likewise require careful assessment. Fewer circuit evaluations do not by themselves imply lower total energy use: accounting must include synthetic-data generation, model training, inference, simulation, and, for future hardware deployment, control and cooling. Potential energy advantages over classical computation depend on the workload and hardware, rather than following automatically from quantum computation (Jaschke & Montangero, 2023). We therefore make no claim of measured energy savings.

REPRODUCIBILITY STATEMENT

Section 4 describes synthetic-data generation, surrogate training, and genetic search. Section 5 summarizes the evaluation protocol; Appendices C and D provide the objectives, input representation, training settings, and benchmark protocols. Appendix E explains the convergence plots and budget-counting conventions. Distribution-matching targets follow ACLBM (Li & Cheng, 2025), and the VQE benchmarks build on the TensorRL codebase (Kundu & Mangini, 2025); the appendix distinguishes published baseline results from our runs. We retain run commands, resolved configurations, random seeds, search histories, and aggregate metrics for reproducibility. Upon acceptance, we expect to release the synthetic-data generator, surrogate-training and search code, together with experiment configurations, feature normalization, and checkpoint metadata.

REFERENCES

Mohamed S. Abdelfattah, Abhinav Mehrotra, Lukasz Dudziak, and Nicholas D. Lane. Zero-cost proxies for lightweight NAS. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0cmMMy8J5q.

Sashwat Anagolum, Narges Alavisamani, Poulami Das, Moinuddin Qureshi, Eric Kessler, and Yunong Shi. Élivágar: Efficient quantum circuit search for classification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2024. doi: 10.1145/3620665.3640354.

Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):45, 2019. doi: 10.1038/s41534-019-0157-8.

Maumita Bhattacharya. Evolutionary approaches to expensive optimisation. *arXiv preprint arXiv:1303.2745*, 2013.

Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Richard Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc'Aurelio Ranzato, Sagi Perel, and Nando de Freitas. Towards learning universal hyperparameter optimizers with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Brian Coyle, Daniel Mills, Vincent Danos, and Elham Kashefi. The born supremacy: quantum advantage and training of an ising born machine. *npj Quantum Information*, 6(1):60, 2020. doi: 10.1038/s41534-020-00288-9.

Tung Dao, Son N. Tran, and Huynh Thi Thanh Binh. Zero-shot quantum neural architecture search. *arXiv preprint arXiv:2605.27410*, 2026.

Andrea Delgado and Kathleen E. Hamilton. Unsupervised quantum circuit learning in high energy physics. *Physical Review D*, 106(9):096006, 2022. doi: 10.1103/PhysRevD.106.096006. URL https://doi.org/10.1103/PhysRevD.106.096006.

Li Ding and Lee Spector. Evolutionary quantum architecture search for parametrized quantum circuits. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '22)*, 2022. doi: 10.1145/3520304.3534012.

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. doi: 10.1038/s41534-022-00570-y.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Efficient multi-objective neural architecture search via lamarckian evolution. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1804.09081.

Azza Fadhel, The Hung Tran, Trong Nghia Hoang, and Jana Doppa. Black-box optimization from small offline datasets via meta learning with synthetic tasks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3862–3870, 2026.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2208.01066.

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*, 10:3007, 2019. doi: 10.1038/s41467-019-10988-2.

Kathleen E. Hamilton, Eugene F. Dumitrescu, and Raphael C. Pooser. Generative model benchmarks for superconducting qubits. *Physical Review A*, 99(6):062323, 2019. doi: 10.1103/PhysRevA.99.062323.

Muqi Han, Xiaobin Li, Kai Wu, Xiaoyu Zhang, and Handing Wang. Enhancing zero-shot black-box optimization via pretrained models with efficient population modeling, interaction, and stable gradient approximation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1117. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2fd37205c429a30cc1bb0afdbbf96857-Abstract-Conference.html.

Zhimin He, Maijie Deng, Shenggen Zheng, Lvzhou Li, and Haozhen Situ. GSQAS: Graph self-supervised quantum architecture search. *Physica A: Statistical Mechanics and its Applications*, 630:129286, 2023. doi: 10.1016/j.physa.2023.129286.

Zhimin He, Chuangtao Chen, Zhengjiang Li, Haozhen Situ, Fei Zhang, Shenggen Zheng, and Lvzhou Li. A meta-trained generator for quantum architecture search. *EPJ Quantum Technology*, 11, 2024a. doi: 10.1140/epjqt/s40507-024-00255-9.

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, 2024b. doi: 10.1609/aaai.v38i11.29135.

Zhimin He, Zhengjie Zeng, Haozhen Situ, Shenggen Zheng, Yan Zhou, and Lvzhou Li. Realigning quantum architecture search with a top-tier-focused training paradigm for performance predictors. *Physical Review Applied*, 26:014094, 2026. doi: 10.1103/wz9c-h89g.

IBM Quantum. Fractional gates. IBM Quantum Documentation, 2026. URL https://quantum.cloud.ibm.com/docs/en/guides/fractional-gates. Accessed September 26, 2026.

Daniel Jaschke and Simone Montangero. Is quantum computing green? an estimate for an energy-efficiency quantum advantage. *Quantum Science and Technology*, 8(2):025001, 2023. doi: 10.1088/2058-9565/acae3e.

Yaochu Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005. doi: 10.1007/s00500-003-0328-5.

Aaron Klein, Herilalaina Rakotoarison, Luca Thale-Bombien, and David Salinas. An open-source training dataset for foundation models for black-box optimization. *arXiv preprint arXiv:2605.23417*, 2026.

Akash Kundu and Stefano Mangini. TensorRL-QAS: Reinforcement learning with tensor networks for improved quantum architecture search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Robert Tjarko Lange, Tom Schaul, Yutian Chen, Chris Lu, Tom Zahavy, Valentin Dalibard, and Sebastian Flennerhag. Discovering attention-based genetic algorithms via meta-black-box optimization. In *Genetic and Evolutionary Computation Conference*, 2023. URL https://arxiv.org/abs/2304.03995.

Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zo"e Holmes, and M. Cerezo. Barren plateaus in variational quantum computing. *Nature Reviews Physics*, 7:174–189, 2025. doi: 10.1038/s42254-025-00813-9.

Chun-Tse Li and Hao-Chung Cheng. Adaptive circuit learning of born machine: towards realization of amplitude embedding and quantum data loading. *Quantum Science and Technology*, 10(2):025019, 2025. doi: 10.1088/2058-9565/adaede.

Shanchuan Li, Daisuke Tsukayama, Jun-ichi Shirakashi, Tetsuo Shibuya, and Hiroshi Imai. Quantum architecture search with neural predictor based on ZX-calculus. *EPJ Quantum Technology*, 12:106, 2025. doi: 10.1140/epjqt/s40507-025-00410-w.

Xiaobin Li, Kai Wu, Yujian Li, Xiaoyu Zhang, Handing Wang, and Jing Liu. Pretrained optimization model for zero-shot black box optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, 2018. doi: 10.1103/PhysRevA.98.062324.

Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Comparison-based optimizers need comparison-based surrogates. In *Parallel Problem Solving from Nature – PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pp. 364–373, 2010. doi: 10.1007/978-3-642-15844-5_37.

Xudong Lu, Kaisen Pan, Ge Yan, Jiaming Shan, Wenjie Wu, and Junchi Yan. QAS-Bench: Rethinking quantum architecture search and a benchmark. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 22880–22898, 2023.

QuanGong Ma, ChaoLong Hao, XuKui Yang, LongLong Qian, Hao Zhang, NianWen Si, MinChen Xu, and Dan Qu. Continuous evolution for efficient quantum architecture search. *EPJ Quantum Technology*, 11:54, 2024. doi: 10.1140/epjqt/s40507-024-00265-7.

Darya Martyniuk, Johannes Jung, Daniel Barta, and Adrian Paschke. Benchmarking quantum architecture search with surrogate assistance. *arXiv preprint arXiv:2506.06762*, 2025.

Satvik Mashkaria, Siddarth Krishnamoorthy, and Aditya Grover. Generative pretraining for black-box optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, 2023.

Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1): 4812, 2018. doi: 10.1038/s41467-018-07090-4.

Samuel M"uller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2112.10510.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for bayesian optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 25444–25470, 2023.

Tung Nguyen, Sudhanshu Agrawal, and Aditya Grover. ExPT: Synthetic pretraining for few-shot experimental design. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Mateusz Ostaszewski, Lea M. Trenkwalder, Wojciech Masarczyk, Eleanor Scerri, and Vedran Dunjko. Reinforcement learning for optimization of variational quantum circuit architectures. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9724412729185d53a2e3e7f889d9f057-Abstract.html.

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 (ICLR)*, 2024. URL https://openreview.net/forum?id=rINBD8jPoP.

Yifeng Peng, Xinyi Li, Samuel Yen-Chi Chen, Kaining Zhang, Zhiding Liang, Ying Wang, and Yuxuan Du. TITAN: A trajectory-informed technique for adaptive parameter freezing in large-scale VQE. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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*, 5:4213, 2014. doi: 10.1038/ncomms5213.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.15063.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 4780–4789, 2019. doi: 10.1609/aaai.v33i01.33014780.

Manuel S. Rudolph, Sacha Lerch, Supanut Thanasilp, Oriel Kiss, Oxana Shaya, Sofia Vallecorsa, Michele Grossi, and Zo"e Holmes. Trainability barriers and opportunities in quantum generative modeling. *npj Quantum Information*, 10:116, 2024. doi: 10.1038/s41534-024-00902-0.

Mehrdad Salami and Tim Hendtlass. A fast evaluation strategy for evolutionary algorithms. *Applied Soft Computing*, 2(3):156–173, 2003. doi: 10.1016/S1568-4946(02)00067-4.

Kumara Sastry, Martin Pelikan, and David E. Goldberg. Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 720–727, 2004. doi: 10.1109/CEC.2004.1330930.

Haozhen Situ, Zhimin He, and Lvzhou Li. Recent advances in quantum architecture search. *arXiv preprint arXiv:2604.26289*, 2026a.

Haozhen Situ, Guohong Li, Zhengjiang Li, Zhimin He, Yongyao Li, and Lvzhou Li. Data-efficient predictor-based quantum architecture search with semi-supervised learning. *Physical Review A*, 113(1):012402, 2026b. doi: 10.1103/9nc8-pjbf.

Robert E. Smith, B. A. Dike, and S. A. Stegmann. Fitness inheritance in genetic algorithms. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pp. 345–350, 1995. doi: 10.1145/315891.316014.

Vicente P. Soloviev, Vedran Dunjko, Concha Bielza, Pedro Larrañaga, and Hao Wang. Trainability maximization using estimation of distribution algorithms assisted by surrogate modelling for quantum architecture search. *EPJ Quantum Technology*, 11(1):69, 2024. doi: 10.1140/epjqt/s40507-024-00282-6.

Yize Sun, Zixin Wu, Volker Tresp, and Yunpu Ma. Quantum architecture search with unsupervised representation learning. *Quantum*, 10:1994, 2026. doi: 10.22331/q-2026-02-03-1994.

Rong-Xi Tan, Ke Xue, Shen-Huan Lyu, Haopu Shang, Yao Wang, Yaoyuan Wang, Sheng Fu, and Chao Qian. Offline model-based optimization by learning to rank. In *International Conference on Learning Representations (ICLR)*, 2025.

Chao Wang, Licheng Jiao, Lingling Li, Jiaxuan Zhao, Guanchun Wang, Fang Liu, and Shuyuan Yang. Task-free adaptive meta black-box optimization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/899511e37a8e01e1bd6f6f1d377cc250-Abstract-Conference.html.

Hanrui Wang, Yongshan Ding, Jiaqi Gu, Zirui Li, Yujun Lin, David Z. Pan, Frederic T. Chong, and Song Han. QuantumNAS: Noise-adaptive search for robust quantum circuits. In *IEEE International Symposium on High-Performance Computer Architecture*, 2022. doi: 10.1109/HPCA53966.2022.00057. URL https://arxiv.org/abs/2107.10845.

Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12(1):6961, 2021. doi: 10.1038/s41467-021-27045-6.

Colin White, Arber Zela, Binxin Ru, Yang Liu, and Frank Hutter. How powerful are performance predictors in neural architecture search? In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2104.01177.

Wenjie Wu, Ge Yan, Xudong Lu, Kaisen Pan, and Junchi Yan. QuantumDARTS: Differentiable quantum architecture search for variational quantum algorithms. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 37745–37764, 2023.

Yixing Xu, Yunhe Wang, Kai Han, Yehui Tang, Shangling Jui, Chunjing Xu, and Chang Xu. ReNAS: Relativistic evaluation of neural architecture search. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4411–4420, 2021. doi: 10.1109/CVPR46437.2021.00439.

Jinyuan Zhang, Aimin Zhou, Ke Tang, and Guixu Zhang. Preselection via classification: A case study on evolutionary multiobjective optimization. *Information Sciences*, 465:388–403, 2018. doi: 10.1016/j.ins.2018.06.073.

Kaining Zhang, Xinbiao Wang, Yuxuan Du, Min-Hsiu Hsieh, and Dacheng Tao. AQER: A scalable and efficient data loader for digital quantum computers. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.02165.

Shi-Xin Zhang, Chang-Yu Hsieh, Shengyu Zhang, and Hong Yao. Neural predictor based quantum architecture search. *Machine Learning: Science and Technology*, 2(4):045027, 2021. doi: 10.1088/2632-2153/ac28dd.

Shi-Xin Zhang, Chang-Yu Hsieh, Shengyu Zhang, and Hong Yao. Differentiable quantum architecture search. *Quantum Science and Technology*, 7(4):045023, 2022. doi: 10.1088/2058-9565/ac87cd.

A Genetic algorithms, initialization, and lineage records

This appendix expands the terminology used in Section 3. It describes generic genetic operators; the particular layer edits, synthetic parent sampling, and surrogate-assisted selection used by GCQAS are specified in Section 4.

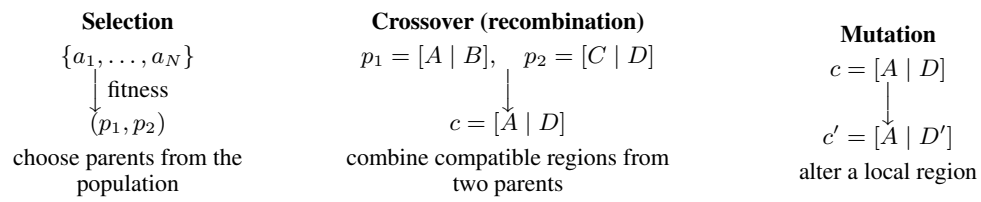


Figure 2: Schematic genetic operators. Blocks denote contiguous gate or layer regions, not individual real-valued parameters. Implementations may use different crossover boundaries and mutation edits, but selection, recombination, and local variation play the roles shown here.

Initialization and evaluation. Initialization constructs the first population before reproduction begins. Random initialization samples legal genomes from the allowed circuit grammar; seeded initialization starts from existing circuits and may use mutations to create additional candidates. Neither choice defines a genetic algorithm. Architecture initialization is also distinct from initializing the continuous angles inside each PQC. For target-task search, parameter optimization supplies verified fitness; the synthetic generator instead assigns its artificial labels without executing a circuit. The different initialization policies used in the experiments are stated in Section 5.

Population, selection, and representation. Genetic algorithms maintain a population of candidate solutions and improve it over generations through selection and variation. Each candidate has a *genotype*, the representation manipulated by the algorithm, and a *phenotype*, the solution obtained when that representation is decoded. In QAS, a genotype can be an ordered or layered list of gate units, and its phenotype is the executable quantum circuit. At generation t, let $\mathcal{P}_t = \{a_{t,1}, \ldots, a_{t,N}\}$ denote the population. The candidates are evaluated, and parent selection gives better-scoring candidates a greater chance to reproduce. Elitism may copy a small number of strong candidates unchanged; it is a survivor policy, distinct from selecting parents for new offspring (Ding & Spector, 2022; Ma et al., 2024).

Crossover, or recombination, takes two parents and constructs a child from regions of both genomes. It does not average the two circuits: for a sequence or layer encoding, it may retain one region from the first parent and replace another with a compatible region from the second. *Mutation* starts from one genome—or from the output of crossover—and makes a stochastic local edit, such as inserting, deleting, replacing, or relocating gates. Crossover exploits structures already in the population, whereas mutation can introduce structures not present in the chosen parents. Because either operation may violate a circuit grammar, a repair step can remove conflicts and enforce the gate set, device connectivity, and size constraints. Figure 2 illustrates these roles; the operators used here are specified in Section 4.

The parent links induced by reproduction form a directed acyclic graph called a genealogy. For candidate c, $\mathcal{G}(c)$ denotes the graph obtained by recursively following its parent links, and $\mathcal{A}_t(c)$ denotes the verified ancestors available when c is proposed. A conventional GA inherits discrete genome content through crossover and mutation; it need not inherit the trained continuous parameters associated with that content. PQC search permits an additional choice: when a child retains a parameterized gate from a parent, its angle may be copied as a warm start instead of being initialized anew. Such parameter transfer can accompany either crossover or mutation, but it is not an intrinsic component of a genetic algorithm. Section 4 states where the proposed search uses this optional mechanism.

Reading a lineage. A parent is a direct predecessor; an ancestor is reachable by following one or more parent links backward. Two offspring can share parents yet differ because of their crossover regions or mutation draws. A crossover child can have two parents, and the same ancestor can contribute through several paths, so the record is generally a graph rather than a single ancestry chain or a tree. Candidates elsewhere in an earlier population are not ancestors unless such a path connects them to the query. This distinction matters when constructing context: a record of all previous evaluations and a record of the query's ancestry contain different information. The notation $\mathcal{A}_t(c)$ also restricts context to ancestors whose verified outcomes are already available, without using the query's unknown outcome.

B Extended Related Work

B.1 Quantum architecture search

QAS automates the discrete design of parameterized quantum circuits. Early weight-sharing methods amortize parameter training through a supercircuit (Du et al., 2022). Differentiable QAS first optimized relaxed distributions over gate choices end to end (Zhang et al., 2022), while Quantum-DARTS uses Gumbel–Softmax and can transfer a searched subcircuit to a larger construction (Wu et al., 2023). Other approaches search directly over discrete circuit descriptions: evolutionary methods update a population through selection and genetic operators, whereas reinforcement-learning (RL) methods construct a circuit as a sequence of gate-placement actions (Ding & Spector, 2022; Kundu & Mangini, 2025). QuantumNAS combines evolutionary circuit-and-mapping search with a trained SuperCircuit whose inherited parameters estimate subcircuit performance under noise (Wang et al., 2022); its amortization is consequently tied to training that task-specific supernetwork. QAS-Bench introduced controlled circuit-generation and unitary-approximation search spaces for comparing such algorithms (Lu et al., 2023); SQuASH more recently learned benchmark surrogates to make repeated comparisons within predefined spaces less expensive (Martyniuk et al., 2025). These families amortize search in different ways, but architecture quality should ultimately be established under the continuous parameter optimization and hardware constraints of the target task.

RL-based ansatz design first demonstrated feedback-driven curriculum learning for LiH VQE while balancing energy accuracy and circuit depth (Ostaszewski et al., 2021). Curriculum Reinforcement Learning for Quantum Architecture Search (CRLQAS) develops this direction using a three-dimensional circuit representation, invalid-action restrictions, randomized halting, and Adam–SPSA under ideal and noisy simulation (Patel et al., 2024). The curriculum structures a difficult sequential decision problem, and the action constraints exclude invalid circuits. Its search policy is nevertheless optimized for each target; reuse across objectives or substantially larger action spaces is not part of the formulation.

Predictor-based QAS screens circuits using learned sequence, graph, or ZX-calculus representations (Zhang et al., 2021; He et al., 2023; Li et al., 2025). Graph self-supervision reduces the labeled data required by a downstream predictor (He et al., 2023). More recent work either learns a predictor-free circuit latent space before applying RL or Bayesian optimization (Sun et al., 2026), or combines a circuit-native Transformer and learnability-aware pretraining with a top-tier-focused ranking loss (He et al., 2026). These methods substantially reduce supervision or align prediction with selection, but they do not use a frozen predictor whose target-time evidence is organized by genetic ancestry. Training-free methods instead replace learned performance labels with structural, expressibility, hardware, or kernel-based proxies (He et al., 2024b; Anagolum et al., 2024; Dao et al., 2026); their rankings are consequently tied to proxies fixed before observing target-task outcomes. EDA-QAS takes an online route, fitting a pairwise SVM to circuits evaluated during the current search and combining its ranking with a trainability objective (Soloviev et al., 2024). It adapts to the target, but begins without a cross-task selection prior. Meta-trained generators and transferable cells instead reuse experience across tasks (He et al., 2024a; Wu et al., 2023). Their meta-training data contain previously optimized circuits and task losses, moving rather than removing the cost of quantum-evaluated supervision.

B.2 Surrogate-assisted evolution and architecture prediction

Evolutionary optimization has long used estimated fitness when evaluations are expensive (Jin, 2005; Bhattacharya, 2013). Fitness inheritance assigns a child the mean or similarity-weighted mean of its parents (Smith et al., 1995; Salami & Hendtlass, 2003; Sastry et al., 2004). These estimators show that parent information can save evaluations. They nevertheless compress the parent–child relationship into a fixed aggregation rule and do not learn how the relevance of ancestry changes across optimization problems.

External surrogates create a model-management risk: predicted fitness can move the population toward a false optimum (Jin, 2005). Preselection limits that risk by predicting which proposal should receive evaluation rather than replacing evaluated fitness (Zhang et al., 2018). Under preselection, a surrogate error can reject a useful child, but it cannot insert an unverified score into the population. The remaining model-design question is what information should condition this selection

when few evaluations are available. For comparison-based evolutionary optimizers, rank-based surrogates additionally preserve invariance to monotone transformations of fitness (Loshchilov et al., 2010). Attention-based genetic algorithms take a broader meta-learning route by learning selection and mutation-rate adaptation across optimization tasks (Lange et al., 2023); they transfer learned operators, but do not represent the persistent multi-generation genealogy of an individual.

Classical NAS reaches the same distinction from architecture prediction. Comparisons of NAS predictors show that rank quality should be connected to downstream search, and zero-cost proxies can assist several search algorithms (White et al., 2021; Abdelfattah et al., 2021). ReNAS directly learns relative architecture order (Xu et al., 2021), while order-preserving objectives can align better with design selection than global squared error (Tan et al., 2025). Regularized evolution established population-based NAS at scale, and LEMONADE showed how Lamarckian weight inheritance can warm-start offspring after architecture edits (Real et al., 2019; Elsken et al., 2019). These results motivate rank-based selection, but architecture predictors ordinarily describe each candidate through its own sequence or graph. Directed provenance—how a candidate was produced, what it inherited, and how its ancestors performed—is not a first-class input in these formulations.

B.3 Pretrained and in-context black-box optimization

Learned optimizers amortize search across problem instances. Prior-Data Fitted Networks (PFNs) learn Bayesian-style prediction by sampling tasks from a prior and masking query labels (M"uller et al., 2022); PFNs4BO adapts this construction to Bayesian optimization (Müller et al., 2023). OptFormer models logged hyperparameter-optimization histories, while BONET models ordered offline design trajectories (Chen et al., 2022; Mashkaria et al., 2023). PFNs4BO is particularly relevant because its pretrained surrogate conditions on observed input–output pairs without target-time weight updates. Those observations form an exchangeable set of generic design points, however, without typed parent–child or parameter-inheritance relations.

Population-oriented pretrained optimizers extend this idea beyond one-point prediction. POM and EPOM learn transferable population-level behavior, while ABOM adapts evolutionary operators online (Li et al., 2024; Han et al., 2025; Wang et al., 2026). Population state and operator adaptation capture useful search dynamics, but do not by themselves represent the directed genealogy that produced an individual.

Synthetic tasks can supply pretraining scale without solved target problems. ExPT and OptBias learn from sampled functions, whereas BBO-Pile emphasizes empirical optimization trajectories (Nguyen et al., 2023; Fadhel et al., 2026; Klein et al., 2026). Synthetic functions remove the need for target-domain evaluations during pretraining, but their transfer depends on which invariances and conditional structure the task generator teaches. Transformers can learn unseen linear functions from demonstrations without parameter updates, while the emergence of such behavior depends on sufficient pretraining-task diversity (Garg et al., 2022; Raventós et al., 2023). Conversely, empirical trajectory corpora preserve realistic search behavior but require a sufficiently broad collection of completed optimization runs. Neither route alone resolves how to pretrain on synthetic data while exploiting structured, target-specific search history at deployment.

B.4 Quantum circuit Born machines

A QCBM represents a classical distribution through the measurement probabilities of a parameterized quantum state (Liu & Wang, 2018; Benedetti et al., 2019). Distribution matching is a test of whether an architecture can prepare prescribed, correlated outcomes, a primitive used in quantum generative modeling and data loading. AQER constructs approximate state-loading circuits through entanglement reduction, product-state approximation of the transformed state, and parameter refinement (Zhang et al., 2026). The product-state step does not restrict the original target to a separable state. Its objective is fidelity to a specified quantum state; matching measurement probabilities does not, by itself, fix that state's phases. Our distribution objective does not require a supplied product-state decomposition, but this does not guarantee efficient preparation of every distribution. AQER is therefore relevant state-preparation literature, not an evaluated baseline or an algorithm shown here to be inferior. Bars-and-Stripes is a sparse controlled instance: only $2^n + 2^m - 2$ of the 2^{nm} binary $n \times m$ images are valid (Benedetti et al., 2019; Hamilton et al., 2019). Estimating Born-machine objectives from samples can itself be costly (Coyle et al., 2020). Moreover, explicit QCBM losses

such as KL divergence can develop loss-induced barren plateaus, making architecture trainability part of the benchmark rather than merely an optimizer detail (Rudolph et al., 2024; Larocca et al., 2025).

Adaptive Circuit Learning of Born Machines (ACLBM) constructs a QCBM by evaluating operator-pool gradients, appending selected operators, and reoptimizing the expanded ansatz (Li & Cheng, 2025). It follows the adaptive-ansatz principle established by ADAPT-VQE, which grows a molecular ansatz one operator at a time (Grimsley et al., 2019). This adaptive construction avoids fixing the final ansatz in advance, but it repeatedly evaluates the operator pool and grows the circuit sequentially; both demands increase as the candidate pool and circuit grow. VQE provides a complementary architecture-search setting in which circuits minimize a molecular Hamiltonian rather than match a probability distribution (Peruzzo et al., 2014). Together, distribution learning and VQE expose search methods to distinct objectives, circuit widths, and noise conditions.

C Task objectives and performance measures

C.1 Distribution matching

For an n-qubit circuit, measuring $U_a(\theta)|0\rangle^{\otimes n}$ in the computational basis defines the Born distribution

$$q_{a,\theta}(x) = \left|\langle x|U_a(\theta)|0\rangle^{\otimes n}\right|^2, \qquad x \in \{0,1\}^n. \tag{5}$$

Given a target distribution $p^\star$, we minimize the forward Kullback–Leibler (KL) divergence

$$\mathcal{L}_{\mathrm{KL}}(a,\theta) = D_{\mathrm{KL}}(p^\star \| q_{a,\theta}) = \sum_x p^\star(x) \log \frac{p^\star(x)}{q_{a,\theta}(x)}. \tag{6}$$

Lower KL indicates that the circuit's measurement distribution more closely matches the target distribution.

C.2 Variational molecular simulation

For VQE, a molecular Hamiltonian $H = \sum_k h_k P_k$ is expressed as a weighted sum of Pauli strings. The circuit prepares a variational state and minimizes its expected energy (Peruzzo et al., 2014),

$$\mathcal{L}_{\mathrm{VQE}}(a,\theta) = \langle 0|U_a^\dagger(\theta) H U_a(\theta)|0\rangle. \tag{7}$$

When a reference ground-state energy E_0 is available, performance is the absolute energy error

$$\Delta E = \left|\mathcal{L}_{\mathrm{VQE}}(a,\theta) - E_0\right|, \tag{8}$$

reported in Hartree. Lower error indicates a state closer in energy to the known ground state.

D Experimental details

D.1 Surrogate input representation

Table 3 summarizes the candidate-aware information used by the surrogate. Circuit statistics describe the candidate's structure. Aggregate relations summarize what is shared, added, removed, or inherited; ordered parent fields retain which parent supplied that structure. Thus, two candidates with similar circuit counts can still have different relational inputs.

Parent-score fields use labels already assigned within the synthetic family during pretraining and verified task outcomes during deployment. The query's own outcome is the prediction target, not an observed input. Feature normalization is fitted on the synthetic training partition and stored with the checkpoint; neither the normalization nor the model weights are refitted during target search. The next subsection gives the pretraining and model settings.

Table 3: Candidate and parent-relation information supplied to GCQAS. Scores are synthetic during pretraining and verified during target search.

Group	Information represented
Circuit structure	Layer, parameter, total-gate, and two-qubit-gate counts.
Aggregate relation	Shared, added, removed, and inherited structure.
Parent outcomes	Parent count and available parent scores (synthetic in pretraining; verified during search).
Directed inheritance	Amount and fraction inherited from each ordered parent.

D.2 SYNTHETIC PRETRAINING

The synthetic corpus contains 200,000 records from six-qubit lineages. Each lineage evolves a population of 20 for 12 generations. Randomized scoring rules consume the 16-dimensional candidate-aware statistics described in Section 4. The surrogate has hidden width 64, one four-head context-attention layer, and two residual refinement steps. Training uses 2,000 epochs, batch size 8,192, and learning rate 3×10^{-4}. The feature transform and selected checkpoint are frozen for deployment.

D.3 DISTRIBUTION-MATCHING PROTOCOL

Target distributions follow Li & Cheng (2025), rather than being introduced here. Log-normal, bimodal, and triangular targets test different probability profiles on discretized supports. BAS assigns probability to valid binary images with uniform rows or uniform columns. An $n \times m$ grid has $2^n + 2^m - 2$ valid patterns among 2^{nm} possible images, making the wider instances a test of sparse, correlated output distributions.

Search uses population size 20, three elites, and 20 candidate proposals per surrogate-controlled selection. Circuit optimization uses at most 250 Adam updates with learning rate 0.05, the core gate pool, loop connectivity, inherited parameters, and PennyLane's `default.qubit` statevector simulator. Exact output probabilities separate architecture selection from measurement noise. The stopping condition is KL $< 10^{-11}$.

For each method, best mean KL is the minimum over generations of the aggregate mean champion KL. On BAS 4×4, GCQAS and original PFNs4BO use three runs, while Pool GA and retrained PFNs4BO use ten. Original PFNs4BO's exported history spans generations 0–9, whereas the other methods' histories span 0–20. These differences preclude interpreting this row as a matched-horizon comparison. ACLBM reports the best result over 20 trials (Li & Cheng, 2025). We use those values because running its released implementation did not recover the reported KL values. They are reference results under the original protocol, not an upper bound on ACLBM or a controlled comparison of compute.

D.4 MOLECULAR SIMULATION PROTOCOL

We use the molecular problems and simulation machinery released with TensorRL-QAS (Kundu & Mangini, 2025).[1] The benchmarks are six-qubit BeH_2 and eight-qubit H_2O in the STO-3G basis. TensorRL reference errors and circuit statistics come from its Table 1 and the shot-noise rows of Table 2. The latter are distinct from its depolarizing-noise results.

Initialization and optimization. Noiseless GCQAS runs initialize random circuits and use COBYLA for parameter optimization. The noisy run initializes one MPS-derived circuit and fills the remaining 19 population slots by mutation. It uses Adam–SPSA with learning rate 0.05. SPSA estimates a gradient using two perturbed objective evaluations, independent of the parameter dimension; Adam adds adaptive moment updates. CRLQAS studies this combination for noisy VQE (Patel et al., 2024). This motivates its use with stochastic energy estimates, without implying that it is uniformly better than other optimizers.

[1] `https://github.com/Aqasch/TensorRL-QAS`

Noise and evaluation. The noisy experiment follows the TensorRL repository's shot-noise simulation and evaluation procedure. TensorRL describes a 10^4-sample shot-noise setting (Kundu & Mangini, 2025); the separate depolarizing settings are not the reference rows used here. Simulated shot noise is not a hardware experiment, and a small sampled energy error alone does not certify the same accuracy for an exact expectation value.

Search budget and resource conventions. The cap of 500 generations with population size 20 corresponds to 10,000 population slots. This is not an exact count of distinct optimized circuits: initialization, elite reuse, and termination affect that count. An RL episode contains multiple circuit-construction decisions and parameter updates, so episode counts cannot be equated with generations or circuit evaluations. We make no compute-efficiency claim from these budgets. TensorRL's fixed variant excludes its MPS-derived state preparation from reported depth and gate counts, whereas the trainable variant includes the trainable initialization. Bold resource entries in the main table identify minima under the reported conventions, not hardware-cost dominance.

E BAS 3×3: CONVERGENCE AND GRADIENT BUDGET

Figure 3 compares the exported aggregate mean champion KL at each generation. GCQAS reaches its stopping threshold by generation 9. Both PFNs4BO variants receive the same evolutionary search setup; the retrained variant additionally receives the synthetic training data, while the original variant retains its pretrained weights. This comparison tests whether changing the training data suffices, rather than attributing the gap to the absence of synthetic examples in the original PFNs4BO model.

Counting convention. Generations and ACLBM epochs represent different work. For the supplied nine-qubit ACLBM trajectory, each epoch is counted as 224 candidate-action gradient checks plus one backpropagation-based parameter update, giving $225e$ calculations after e epochs. Its 444 recorded epochs therefore correspond to $99{,}900$ calculations. Evolutionary runs use their logged cumulative training-evaluation counters; EDA-QAS uses its gradient/update counter, which excludes the information-content random-walk objective calls. The comparison tracks this selected component of optimization effort, not all simulator calls, scalar derivatives, quantum measurement shots, or wall-clock time. Circuit size and differentiation method can change the cost of an individual calculation. Offline pretraining and surrogate inference are not included in this axis.

The ACLBM curve is the supplied epoch-level trajectory; the dotted line is the published reference used in Table 1. The trajectory is not an aggregate of matched-seed reruns. Figure 4 therefore complements the endpoint comparison under the stated counting convention, without establishing an end-to-end runtime ratio.

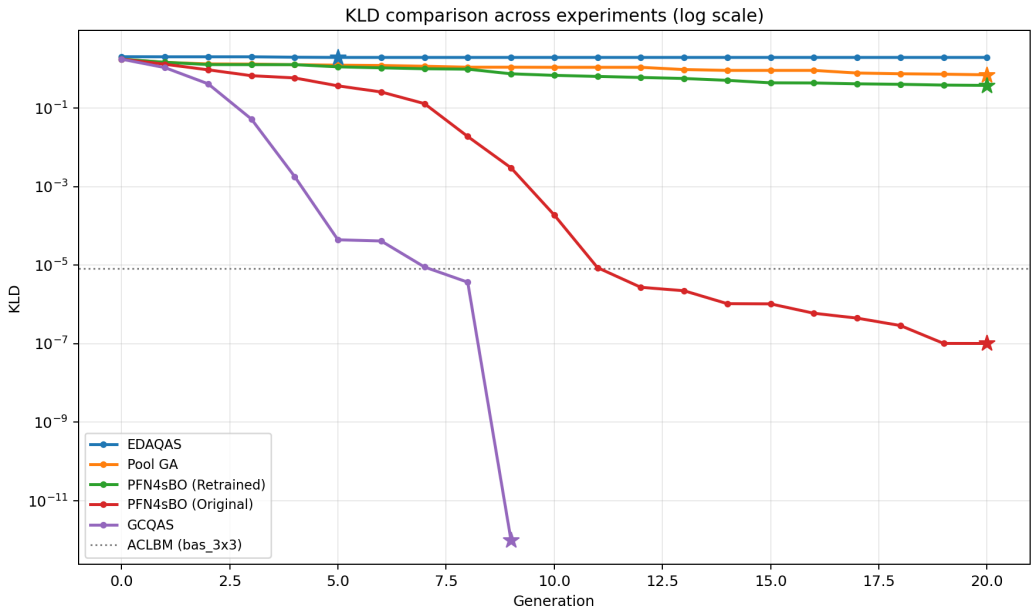


Figure 3: BAS 3×3 convergence against evolutionary generations. The vertical axis shows KL divergence on a logarithmic scale. GCQAS reaches the stopping threshold by generation 9, while Pool GA and the PFNs4BO variants retain higher aggregate mean KL over the displayed histories. The dotted horizontal line denotes ACLBM's published performance, not its progress per generation.

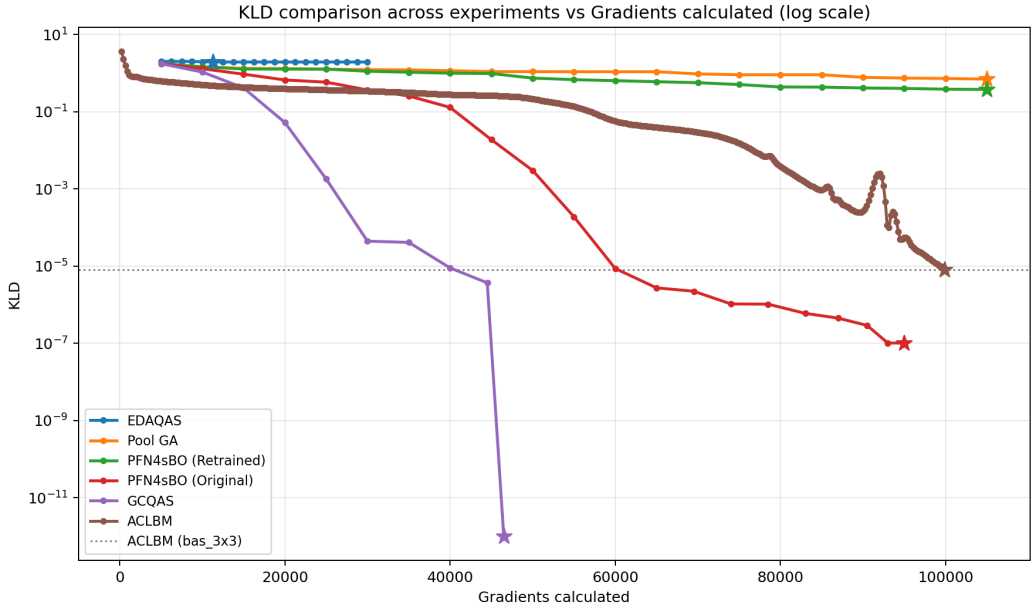


Figure 4: BAS 3×3 convergence against the counted gradient/update budget. ACLBM's epoch axis is converted using 224 candidate-action checks and one parameter-update calculation per epoch. GCQAS reaches lower KL at approximately 4.7×10^4 calculations than the supplied ACLBM trajectory reaches at 99,900, using less than half this counted budget. The dotted line is the published ACLBM reference. Counts are not equivalent to total runtime; see the accounting conventions in Appendix E.