

Generative Quantum Search for Combinatorial Optimization: Learning Shallow Quantum Samplers for Classically Refined Solution Subspaces

David Vesterlund

david@vesterlundventures.se

Vesterlund Ventures Holding AB <https://orcid.org/0009-0000-6455-1141>

Research Article

Keywords: QAOA, Quantum Algorithm, Graph optimization, Quantum machine Learning, Quantum Eigensolver, Machine Learning

Posted Date: October 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11337099/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Generative Quantum Search for Combinatorial Optimization: Learning Shallow Quantum Samplers for Classically Refined Solution Subspaces

WestQuant Research

September 2026

Abstract

We port the Generative Quantum Eigensolver + Quantum-Selected Configuration Interaction (GQE+QSCI) architecture from quantum chemistry to combinatorial graph optimization. Our framework, GQO+QSS (Generative Quantum Optimization + Quantum-Selected Search), follows the same principle: ML generates circuit programs, quantum samplers produce bitstrings, classical recovery repairs and expands them into a restricted subspace, and a classical solver finds the optimum within that subspace.

We formalize the key quantity as the **recovery-basin mass** $p_r(q_\theta, G) = \Pr_{x \sim q_\theta}[x \in A_r(G)]$, where $A_r(G) = \{x : E_r(R(x)) \cap X^*(G) \neq \emptyset\}$ is the preimage of the optimal set under repair R and r -flip expansion E_r . Under B independent shots, success probability becomes $P_{\text{success}} = 1 - (1 - p_r)^B$. The quantum-benefit question reduces to: does $p_r(q_{\text{quantum}}, G) > p_r(q_{\text{classical}}, G)$?

We test this on Maximum Weighted Independent Set (MWIS) with a preregistered, frozen protocol. All 150 ground truths are certified with CBC + brute-force verification. Exact recovery-basin masks are enumerated for all $n \leq 20$ instances. Nine samplers are compared on a held-out test set: uniform, greedy, simulated annealing, tabu search, multi-start local search, Metropolis, warm-start QAOA, oracle QAOA, and a matched learned classical policy.

Key findings:

1. **Tabu search dominates at all sizes** ($p_r = 0.82\text{--}0.94$), followed by simulated annealing. All quantum samplers and the learned classical policy lose to both at every problem size under the frozen test protocol.

2. **Oracle QAOA loses to SA under fair evaluation:** With random QAOA configurations (not per-instance optimized), oracle QAOA achieves $p_r = 0.014$ at $n = 28$ vs SA's 0.088. The earlier positive result ($p_r = 0.185$ vs 0.130) required per-instance best-of-50 configuration selection—an oracle upper bound, not a deployable method.

3. **Learned classical policy matches oracle QAOA:** A matched 140K-parameter classical policy achieves $p_r = 0.005\text{--}0.101$, comparable to oracle QAOA's 0.014–0.113. The quantum circuit's proposal distribution provides no structural advantage over a matched learned classical model in this setting.

4. **Recovery-basin masks reveal the phase structure:** At $r = 2$, uniform-sampling P_{recover} drops from 0.975 ($n = 16$) to 0.775 ($n = 20$) at $B = 200$, identifying the regime where sampler quality matters.

5. **Hardware robustness without advantage:** 136 cores on Piast-Q show 93.4% recovery consistency between QPU and simulator, but no basin-mass advantage.

This is a falsification study: Under rigorous, preregistered evaluation with matched learned baselines, shallow QAOA-derived proposal distributions do not outperform matched classical proposal models, and both lose to strong classical baselines (SA, tabu). The earlier positive oracle result was an artifact of per-instance configuration selection. We claim no quantum advantage.

1 Introduction

1.1 The GQE+QSCI Pattern

Elgammal & MauSSner (2026) demonstrated a pipeline coupling a Generative Quantum Eigensolver (GQE) to Quantum-Selected Configuration Interaction (QSCI) for EUV photoresist chemistry. The key architectural insight is not the chemistry itself but the work division:

ML generates $\rightarrow$ quantum samples $\rightarrow$ classical compute refines $\rightarrow$ ML learns

A GPT-2 policy (43–46M parameters) generates UCCSD operator sequences. Circuits are sampled on simulator or QPU. Bitstrings become a small set of determinants. QSCI diagonalizes the Hamiltonian classically in this restricted subspace. The result feeds back as training signal.

Critically, at 32 qubits, the successful QSCI subspace corresponded to only **0.017%** of the determinant space. The QPU does not solve the problem—it finds a small, valuable subspace that classical computation then exploits.

Important distinction: QSCI in chemistry projects an electronic Hamiltonian onto a determinant subspace and diagonalizes a non-trivial Hamiltonian there. For a classical combinatorial cost function, the cost Hamiltonian is diagonal in the computational basis, so “restricted diagonalization” reduces to selecting the best configuration in the candidate set. The real analogy is therefore:

quantum-selected support $\rightarrow$ classically exploitable restricted search region

rather than an exact algorithmic port of QSCI diagonalization. We describe GQO+QSS as *QSCI-inspired selected-state search*, not a direct equivalent.

1.2 Porting to Graph Optimization

We port this pattern to combinatorial graph optimization:

GQE+QSCI (chemistry)	GQO+QSS (graphs)
Molecular active space	Hard graph core
UCCSD operator pool	Graph/QAOA operator pool
GPT-2 generates operator sequences	WestQuant generates circuit programs
Quantum circuit sampling	QPU/simulator sampling
Bitstrings $\rightarrow$ Slater determinants	Bitstrings $\rightarrow$ candidate graph solutions
QSCI determinant selection	Quantum-selected solution set
Classical diagonalization	Restricted classical optimization
Configuration recovery	Graph-specific repair + neighborhood expansion
CASCI/FCI reference	Exact optimum on small cores
Chemical accuracy	Optimality gap / approximation ratio
Subspace coverage	Near-optimal solution-space coverage
Circuit depth L	QAOA/operator depth p
IQM Emerald hardware	Piast-Q / Rigetti / IQM / QuEra

1.3 Architecture

G_large $\rightarrow$ HPC $\rightarrow$ H_hard $\rightarrow$ generative policy $\rightarrow$ circuits $\rightarrow$ QPU/sim $\rightarrow$ S_Q
 $\rightarrow$ classical recovery $\rightarrow$ S_Q+ $\rightarrow$ restricted solver $\rightarrow$ x* $\rightarrow$ reward $\rightarrow$ policy update

1. **HPC preprocessing:** Large graph $\rightarrow$ small hard core

2. **Generative policy:** Graph-conditioned circuit program generation
3. **Quantum sampling:** QPU/simulator produces bitstrings S_Q
4. **Classical recovery:** Repair + local expansion + restricted exact solve
5. **Reward:** Full hybrid solver quality (not just quantum energy)
6. **Policy update:** Learn to generate better circuits

1.4 Research Question

Can a quantum sampler place more probability mass inside a classical recovery basin than classical samplers?

This is testable in isolation: give each sampler exactly 200 shots, run the same classical recovery, and compare the post-recovery optimality gap.

1.5 Recovery-Basin Mass Theory

Let the sampler define a distribution $q_\theta(x | G)$. Let $R(x)$ be the repair function and $E_r(R(x))$ the set of states reachable by r -flip recovery. Define the **optimal-basin preimage**:

$$A_r(G) = \{x : E_r(R(x)) \cap X^*(G) \neq \emptyset\}, \quad (1)$$

where $X^*(G)$ is the set of global optima. The sampler’s key quantity is the **recovery-basin mass**:

$$p_r(q_\theta, G) = \Pr_{x \sim q_\theta} [x \in A_r(G)]. \quad (2)$$

With B independent shots, the probability that at least one sample lands in the recovery basin is:

$$P_{\text{success}} = 1 - (1 - p_r)^B. \quad (3)$$

The quantum-benefit question becomes precise and falsifiable:

$$p_r(q_{\text{quantum}}, G) > p_r(q_{\text{classical}}, G)? \quad (4)$$

The sample complexity to reach success probability $1 - \delta$ is:

$$B_\delta = \frac{\log \delta}{\log(1 - p_r)}. \quad (5)$$

This formulation replaces the earlier “subspace scaling law” ($|S_Q|/2^n \approx 200/2^n$), which was largely tautological under a fixed 200-shot budget. The interesting quantity is not how many points are sampled from 2^n , but **how much probability mass the sampler places in the recovery basin** $A_r(G)$. Two samplers can both draw 200 points from 2^{100} , but one may place 10^{-7} of its mass in $A_r(G)$ and the other 10^{-2} —that is where the quantum-sampling question lives.

This also makes the connection to QSCI more scientifically accurate: QSCI’s important property is not merely that the subspace is small, but that the small subspace contains the configurations needed for a good approximation.

2 Related Work

QAOA and variants. The Quantum Approximate Optimization Algorithm (?) is the foundational quantum-classical hybrid for combinatorial optimization. Warm-start QAOA (?) initializes from a relaxed SDP solution, and recursive QAOA (?) progressively reduces problem size. Several works combine QAOA with classical local improvement, which is the spirit of our recovery pipeline.

Learning QAOA parameters. RL-based parameter prediction (?) and parameter transfer across instances have been explored to avoid per-instance optimization. Our generative policy follows this line but targets recovery-basin mass rather than raw energy.

Sample-based quantum diagonalization. Quantum-Selected Configuration Interaction (QSCI) (?) and the Generative Quantum Eigensolver (GQE) (?) use quantum samples to select a subspace for classical exact diagonalization. GQE+QSCI (?) combines a generative model with QSCI. Our GQO+QSS ports this pattern to combinatorial optimization.

Classical stochastic search. Simulated annealing, tabu search, and multi-start local search are well-established baselines for combinatorial optimization. We include all three, plus a learned classical proposal distribution, to ensure fair comparison against the learned quantum policy.

Positioning. Our contribution is a *learned quantum proposal distribution optimized specifically for capture of a mathematically defined classical recovery basin under a fixed downstream recovery budget*. We do not claim novelty for quantum samples + classical post-processing per se; the novelty is the recovery-basin mass formulation and the rigorous falsification protocol.

3 Methods

3.1 Samplers

Five samplers, all given exactly $N_s = 200$ shots:

1. **Uniform random:** Null hypothesis baseline
2. **Greedy classical:** Local search with random restarts, collecting all visited states
3. **Fixed QAOA** ($p = 2$): Standard QAOA with random parameter search
4. **Learned QAOA** ($p = 3$): QAOA with graph-conditioned parameter selection
5. **Generative policy:** Graph-conditioned circuit program with density-aware parameter selection

3.2 Classical Recovery

Three-step recovery (graph analog of QSCI configuration recovery):

1. **Repair:** Problem-specific feasibility restoration
 - MWIS: Remove conflicts, greedily add non-conflicting vertices
 - MaxCut: Flip vertices that improve cut value
 - Balanced partition: Enforce $|V_1| = |V_2| = n/2$
 - Vertex cover: Add vertices to cover uncovered edges
2. **Local expansion:** Generate all 1-flip and 2-flip neighborhoods of each repaired sample
 - For $n = 20$: $1 + 20 + 190 = 211$ neighbors per sample
 - 200 samples $\rightarrow$ up to 42,200 candidates (with overlap)
3. **Restricted solve:** Find the best solution in the expanded subspace S_Q^+
 - Exact evaluation of all candidates
 - This is the graph analog of QSCI diagonalization

3.3 Evaluation Metrics

1. $R_\epsilon(S)$: Near-optimal coverage = $\frac{|\{x \in S: f(x) \leq f^* + \epsilon\}|}{|S|}$
2. P_{hit} : Probability of hitting near-optimal region
3. $D(S)$: Sample diversity (normalized entropy of unique solutions)
4. Δ_{opt} : Post-recovery optimality gap
5. Subspace fraction: $|S_Q|/2^n$ (analog of determinant fraction)

3.4 Reward Function

The policy is trained on a multi-objective reward, **not** just QAOA energy:

$$R = -w_1 \Delta_{\text{opt}} + w_2 R_\epsilon + w_3 D(S) - w_4 N_{\text{shots}} - w_5 D_{\text{circuit}} - w_6 T_{\text{classical}} \quad (6)$$

This trains the policy to optimize the full hybrid solver, not just the quantum circuit.

3.5 Qubit Ladder

We test $n = 8, 10, 12, 16, 20, 24, 28, 30$ with 3 seeds each, 200 shots per sampler. This is the graph analog of the paper’s $14 \rightarrow 18 \rightarrow 20 \rightarrow 24 \rightarrow 28 \rightarrow 32 \rightarrow 40$ qubit ladder.

4 Results

4.1 Qubit Ladder: Post-Recovery Gap

Table 1: Post-recovery optimality gap for $n \leq 50$ (lower = better). All samplers reach optimum after recovery.

n	2^n	Best Sampler	Avg Gap	Avg R_ϵ	Avg Diversity
8	256	uniform	0.0000	0.0033	0.9067
10	1,024	uniform	0.0000	0.0017	0.9742
12	4,096	uniform	0.0000	0.0000	0.9943
16	65,536	uniform	0.0000	0.0000	0.9996
20	1,048,576	uniform	0.0000	0.0000	0.9943
24	16,777,216	uniform	0.0000	0.0000	0.9943
28	268,435,456	uniform	0.0000	0.0017	0.9943
30	1,073,741,824	uniform	0.0000	0.0000	0.9943
40	1.1×10^{12}	uniform	0.0000	0.0000	0.9928
45	3.5×10^{13}	uniform	0.0000	0.0000	0.9928
50	1.1×10^{15}	uniform	0.0000	0.0000	0.9928

Finding: For $n \leq 50$, classical recovery finds the optimal solution regardless of which sampler produced the initial samples. The optimality gap is 0.0000 for all samplers at all sizes.

4.2 The Classical Boundary: $n = 55$ to 100

Key findings:

- $n = 50$: Classical recovery works perfectly (100% success, gap = 0).
- $n = 55$: **First failure.** Greedy recovery fails 100%, uniform fails 50%.

Table 2: Classical boundary: post-recovery gap and success rate for $n = 50$ to 100. Radius = adaptive (3 for $n \leq 60$, 4 for $n \geq 65$). 200 shots, 2 seeds.

n	2^n	Radius	Uniform Gap	Greedy Gap	U Success	G Success
50	1.1×10^{15}	3	0.0000	0.0000	100%	100%
55	3.6×10^{16}	3	0.2563	0.8896	50%	0%
60	1.2×10^{18}	3	0.0000	0.6066	100%	50%
65	3.7×10^{19}	4	0.3556	1.5888	50%	50%
70	1.2×10^{21}	4	0.5146	1.3219	50%	0%
75	3.8×10^{22}	4	0.2886	0.4761	50%	0%
80	1.2×10^{24}	4	1.0447	0.9456	0%	0%
85	3.9×10^{25}	4	0.5927	1.0695	0%	0%
90	1.2×10^{27}	4	0.3364	0.7299	50%	0%
95	4.0×10^{28}	4	1.2372	1.8883	0%	0%
100	1.3×10^{30}	4	0.9567	2.0898	0%	0%

- $n = 80$: **Complete failure**. Both uniform and greedy fail 100%.
- $n = 100$: Gap ≈ 1 –2 units. Subspace is $\sim 10^{-24}$ of 2^{100} .

This is the **classical boundary** for MWIS with r -flip expansion. Below $n = 55$, classical recovery is sufficient. Above $n = 80$, classical recovery is insufficient. In the transition region ($55 \leq n \leq 80$), the quality of the initial samples matters—this is where GQO+QSS quantum sampling becomes valuable.

4.3 Subspace Fraction (and why it is not the key metric)

Table 3: Subspace fraction: $|S_Q|/2^n$ for uniform sampler (200 shots).

n	$ S_Q $	2^n	Fraction	R_ϵ
8	136	256	0.531	0.005
10	180	1,024	0.176	0.005
12	199	4,096	0.049	0.000
16	200	65,536	0.003	0.000
20	199	1,048,576	0.0002	0.000
24	199	16,777,216	0.00001	0.000
28	199	268,435,456	0.0000007	0.000
30	199	1,073,741,824	0.0000002	0.000

The subspace fraction $|S_Q|/2^n \approx 200/2^n$ decreases exponentially with n . **This is largely tautological under a fixed 200-shot budget**—of course $200/2^n \rightarrow 0$ as n grows. We do not treat this as a “scaling law.”

The interesting quantity is not how many points are sampled from 2^n , but **how much probability mass the sampler places in the recovery basin** $A_r(G)$ (§1.5). Two samplers can both draw 200 points from 2^{100} , but one may place 10^{-7} of its mass in $A_r(G)$ and the other 10^{-2} . That is the quantity future experiments must measure.

4.4 Noise-Induced Support Broadening (preliminary, single instance)

Observation: On this single $n = 10$ instance, moderate depolarizing noise (10^{-3} to 3×10^{-3}) increased sample diversity by 2–7% over noiseless simulation. The relationship between noise

Table 4: Noise-induced support broadening: MWIS $n = 10$, 200 shots, single instance.

Noise	Best Sample	Unique	Diversity	R_ϵ	Recovered	Gap
0.0000	-2.315	37	0.4453	0.000	-5.482	0.000
0.0001	-2.315	37	0.4453	0.000	-5.482	0.000
0.0010	-2.625	42	0.4560	0.000	-5.482	0.000
0.0030	-4.974	46	0.4536	0.000	-5.482	0.000
0.0100	-3.639	8	0.3232	0.000	-4.974	0.508
0.0300	-2.501	8	0.2983	0.000	-5.482	0.000
0.1000	-5.482	39	0.4513	0.005	-5.482	0.000

level, diversity, and solution quality is irregular across the table.

We describe this as **preliminary evidence of noise-induced support broadening**, not a confirmed result. A systematic study requires many graph instances, multiple circuit depths, multiple noise models, and—most importantly—the connection noise $\rightarrow p_r \rightarrow P_{\text{success}}$, not just noise $\rightarrow$ diversity.

4.5 Sampler Comparison

At $n = 10$, seed=42 (representative example):

Table 5: Sampler comparison: MWIS $n = 10$, 200 shots.

Sampler	Best	Mean	R_ϵ	P_{hit}	Diversity	Unique
uniform	-5.482	-1.43	0.005	1.0	0.974	180
greedy	-5.482	-2.31	0.040	1.0	0.937	137
qaoa	-5.482	-1.87	0.005	1.0	0.891	115
learned_qaoa	-4.974	-1.52	0.000	0.0	0.745	60
generative	-4.251	-1.18	0.000	0.0	0.808	72

After recovery, all samplers reach the optimum (-5.482 , gap= 0.0000).

Key observations:

- **Greedy** has the highest R_ϵ (0.040)—it visits more near-optimal states during local search
- **Uniform** has the highest diversity (0.974)—maximum exploration
- **QAOA** is slower (6.6s) but still finds the optimum after recovery
- **Learned QAOA** sometimes fails to hit the near-optimal region ($P_{\text{hit}} = 0$) but recovery still saves it
- **Generative** is between QAOA and learned QAOA in quality

5 Generative Policy

5.1 Architecture

We implement a graph-conditioned generative policy (analog of the paper’s GPT-2):

- **Input:** 20-dim graph feature vector (density, degree stats, spectral eigenvalues, problem type one-hot, hardness)
- **Model:** 3-layer MLP, 256 hidden units, $\sim 140\text{K}$ parameters

- **Output:** Circuit program (QAOA depth p , γ parameters, β parameters, mixer type)
- **Training:** REINFORCE on hybrid solver reward
- **Replay buffer:** 5000 capacity, batch size 16

The policy is intentionally smaller than the paper’s 43–46M GPT-2, as our circuit programs are simpler than UCCSD operator sequences.

5.2 Training

The training loop follows the GQE+QSCI protocol:

Algorithm 1 GQO+QSS Policy Training

```

for episode = 1, ...,  $N$  do
  Sample random graph core  $H$ 
  Extract features  $f(H)$ 
  Generate circuit program  $C = \pi_\theta(f(H), \text{noise})$ 
  Run QAOA with  $C$ , get samples  $S_Q$ 
  Run classical recovery:  $S_Q \rightarrow S_Q^+ \rightarrow x^*$ 
  Compute reward  $R = -w_1 \Delta_{\text{opt}} + w_2 R_\epsilon + \dots$ 
  Add  $(f(H), C, R)$  to replay buffer
  if episode mod 10 = 0 then
    Sample batch from replay buffer
    REINFORCE gradient step
  end if
end for

```

6 Hardware Robustness on Piast-Q

6.1 Execution Summary

- **Device:** Piast-Q trapped-ion processor (20 qubits, all-to-all connectivity, native gates $\{rz, r, r_{xx}\} = \text{Mølmer-Sørensen}$)
- **Program:** 136 frozen cores across 3 problem types (MWIS, Balanced Partition, MaxCut)
- **Sizes:** $n = 8, 10, 12, 16, 20$ qubits
- **Shots:** 32,000 total (120 cores $\times$ 200 + 16 cores $\times$ 500)
- **QPU wall-clock time:** 5.39 hours (323 min)—note: this exceeds the 3.5h budget estimate because the estimate used 200ms/shot whereas actual PiastQ timing is ~ 15 s per execution including rate-limit spacing
- **Failures:** 0 (all 136 cores completed successfully)

6.2 What This Experiment Tests (and What It Does Not)

This experiment tests **hardware robustness**: does classical recovery on QPU samples find the optimal solution at the same rate as recovery on simulator samples (same frozen circuits)? It does **not** test whether QPU sampling improves optimization—that requires comparing quantum samplers against strong classical samplers in a regime where sampler quality matters (§7.1).

Table 6: Recovery on QPU samples vs simulator samples.

Metric	QPU	Simulator
Raw samples found optimal	34/136 (25%)	18/136 (13%)
After recovery found optimal	127/136 (93%)	127/136 (93%)
Recovery improved (raw miss $\rightarrow$ recovered find)	93/136 (68%)	109/136 (80%)

Finding: GQO+QSS recovery is robust to the tested hardware implementation.

Classical recovery on QPU samples finds the optimal solution at the same rate (127/136 = 93.4%) as recovery on simulator samples. This is a hardware portability / robustness result, **not yet evidence that QPU sampling improves optimization.**

The raw QPU samples found optimal 34/136 vs simulator 18/136—an interesting gap that warrants paired per-instance statistical analysis, but the totals alone do not establish a QPU advantage.

6.3 Results by Problem Size

Table 7: QPU vs simulator recovery by problem size.

n	Cores	QPU raw	QPU + recovery	Sim raw	Sim + recovery
8	28	17 (61%)	28 (100%)	12 (43%)	28 (100%)
10	27	12 (44%)	27 (100%)	3 (11%)	27 (100%)
12	27	5 (19%)	27 (100%)	3 (11%)	27 (100%)
16	27	0 (0%)	27 (100%)	0 (0%)	27 (100%)
20	27	0 (0%)	18 (67%)	0 (0%)	18 (67%)

6.4 Key Findings

1. **Recovery closes the gap at $n = 8$ –16:** Raw QPU samples miss optimal at $n \geq 16$ (0% found), but after classical recovery, 100% find optimal for $n = 8$ –16. This is the recovery pipeline working as designed on hardware.
2. **QPU matches simulator exactly:** At every size, QPU+recovery and simulator+recovery find optimal at the same rate. At $n = 20$, both fail on the same 9 cores. Hardware noise does not degrade recovery in this regime.
3. **$n = 20$ identifies a recovery transition:** Both QPU and simulator recovery fail on 9/27 $n = 20$ cores with 2-flip expansion. This is the regime where sampler quality becomes consequential—the focus of the critical experiment (§7.1).

6.5 Results by Problem Type

Table 8: QPU recovery results by problem type.

Problem	Cores	QPU + recovery	Sim + recovery	QPU gap
MWIS	66	65 (98%)	64 (97%)	0.017
Balanced Partition	35	34 (97%)	35 (100%)	0.029
MaxCut	35	28 (80%)	28 (80%)	−3.51

6.6 Null Control Validation

Table 9: Null control validation with recovery.

Metric	QPU	Simulator
Null cores	20	20
Recovery found optimal	17/20 (85%)	18/20 (90%)
Mean recovery gap	−0.79	−0.87

Both QPU and simulator recovery miss 2–3 null cores. The null cores that fail are $n = 20$ MaxCut instances where the stored “optimal” is wrong (recovery found better solutions). This is a core generation issue, not a hardware issue—simulator and QPU behave identically.

6.7 Hardware-Portability

- Piast-Q’s all-to-all connectivity eliminated SWAP overhead
- Native Mølmer-Sørensen gates matched the RZZ interactions in QAOA
- No circuit recompilation needed between problems
- Transpiled circuit depths (18–54) are well within NISQ feasibility

7 Discussion

7.1 Recovery Transition, Not a Beaten Boundary

The central finding: at $n \leq 16$ with 2-flip neighborhood expansion, classical recovery is powerful enough to find the optimum from any reasonable starting point. The quantum sampler’s quality does not affect the final result in this regime.

We describe this as a **recovery transition** for the tested classical samplers, not a “beaten boundary.” Whether quantum sampling can outperform classical samplers beyond this transition is the central experiment addressed next (§7.1).

The paper’s authors explicitly stated: “classical subspace expansion reaches the 32 to 40-qubit spaces with no quantum sampler, so that boundary is mapped, not beaten.” For chemistry, the boundary was mapped. For MWIS at $n \leq 20$ with 2-flip expansion, we similarly identify a regime where sampler quality is masked by recovery. The regime where sampler quality becomes consequential lies beyond $n = 20$ or requires larger expansion radius.

- $n \leq 16$: Classical recovery sufficient (sampler quality masked)
- $n = 20$: Transition zone (2-flip recovery fails on 9/27 cores)
- $n \geq 24$: Sampler quality becomes consequential (not yet tested with quantum samplers)

At $n = 100$, the optimality gap from classical recovery is ~ 1 –2 units on a problem with optimal value ~ -21 . This is a ~ 5 –10% suboptimality—substantial room for quantum sampling to add value.

7.2 What the Subspace Fraction Does and Does Not Tell Us

The fraction $|S_Q|/2^n \approx 200/2^n$ is largely determined by the fixed shot budget and is not, by itself, evidence that the architecture is working in an exponentially small fraction of the space. The interesting quantity is the **recovery-basin mass** p_r (§1.5): how much probability mass the sampler places in $A_r(G)$. Future experiments must measure p_r directly, not just $|S_Q|/2^n$.

Table 10: Subspace fraction at scale (largely tautological under fixed 200-shot budget).

n	$ S_Q $	2^n	Fraction
8	136	256	5.3×10^{-1}
20	199	10^6	2.0×10^{-4}
50	199	10^{15}	2.0×10^{-13}
100	1,400,000	10^{30}	1.4×10^{-24}

7.3 Noise-Induced Support Broadening (preliminary)

The single-instance observation that moderate noise increased sample diversity by 2–7% is preliminary. It suggests—but does not confirm—that near-term hardware noise may not be purely detrimental for shallow circuits. A systematic study requires the chain noise $\rightarrow p_r \rightarrow P_{\text{success}}$, not just noise $\rightarrow$ diversity, across many instances and noise models.

7.4 Generative Policy Results

The 140K-parameter policy was trained for only 100 episodes in the easy regime (where recovery always succeeds), so it does not yet demonstrate value:

- Learned to prefer shallow circuits ($p = 1$)
- Best average reward: -2.0627 (improved from -2.1384)
- Policy diversity: 0.359 (lower than uniform’s 0.958)
- Policy gap: 0.433 (higher than uniform’s 0.000 at $n \leq 12$)

The policy should be trained directly against p_r or expected post-recovery regret in the hard-core regime, where sampler quality actually affects the final result. This is the focus of ongoing work (§7.2).

The policy converges toward specific parameter regions, reducing diversity. This is expected for a deterministic policy without exploration noise. Future work should balance exploration and exploitation, particularly in the transition zone ($55 \leq n \leq 80$) where sample quality matters.

7.5 When Quantum Sampling Helps

Based on our results, quantum sampling provides value when:

1. $n \geq 55$: The probability of random sampling hitting the optimal basin drops below 100%.
2. $n \geq 80$: Classical recovery fails completely. Quantum sampling must place samples closer to the optimum.
3. **The basin is structurally aligned with quantum amplitudes:** QAOA’s quantum tunneling can reach basins isolated by high energy barriers.
4. **The expansion radius is limited:** With 4-flip expansion and 200 samples, the subspace is $\sim 1.4\text{M}$ candidates. If the quantum sampler places samples within 4 flips of the optimum, recovery will find it.

8 Critical Experiment: Sampler Comparison in the Transition Regime

8.1 Setup

We conduct the critical experiment proposed in the previous version of this paper: comparing four samplers—uniform random, greedy, simulated annealing (SA), and warm-start QAOA—in the transition regime ($n = 20, 24, 28$) where recovery does not automatically solve the problem.

Certified ground truths: All 90 instances (30 per size) are solved with PuLP/CBC, a certified MILP solver requiring OPTIMAL status. This resolves the ground-truth certification gap identified in §6.5. The certified optimal values are used as f^* for all gap calculations.

Samplers: Each sampler gets exactly 200 shots. SA uses 1000 steps per shot with geometric cooling ($T_0 = 2.0$, $T_f = 0.01$). Warm-start QAOA initializes from greedy solutions and perturbs around them.

Recovery: 2-flip neighborhood expansion, same for all samplers. Basin mass p_r is measured by checking whether the optimal assignment is in the expanded neighborhood of each repaired sample.

Statistical rigor: 30 independent graph instances per size (seeds 42–71), bootstrap 95% CI (1000 resamples), paired per-instance comparisons.

8.2 Results: Recovery-Basin Mass

Table 11: Recovery-basin mass p_r by sampler and n (30 instances, mean $\pm$ 95% bootstrap CI).

Sampler	$n = 20$	$n = 24$	$n = 28$
Uniform	0.196 ± 0.066	0.095 ± 0.044	0.143 ± 0.058
Greedy	0.175 ± 0.043	0.076 ± 0.022	0.078 ± 0.024
Sim. Annealing	0.325 ± 0.033	0.162 ± 0.025	0.120 ± 0.019
Warm-Start QAOA	0.167 ± 0.042	0.087 ± 0.032	0.077 ± 0.031

Key finding: Simulated annealing has the highest p_r at all sizes, with non-overlapping confidence intervals against greedy and warm-start QAOA at $n = 24$ and $n = 28$. SA places 2–4 $\times$ more mass in the recovery basin than greedy or warm-start QAOA.

8.3 Results: Post-Recovery Success

Table 12: Post-recovery success rate (optimal found) and mean gap by sampler.

Sampler	$n = 20$ (success/gap)	$n = 24$ (success/gap)	$n = 28$ (success/gap)
Uniform	28/30 (93%), 0.010	27/30 (90%), 0.011	26/30 (87%), 0.032
Greedy	30/30 (100%), 0.000	26/30 (87%), 0.028	25/30 (83%), 0.061
Sim. Annealing	30/30 (100%), 0.000	30/30 (100%), 0.000	30/30 (100%), 0.000
Warm-Start QAOA	29/30 (97%), 0.024	24/30 (80%), 0.093	23/30 (77%), 0.092

Key finding: Simulated annealing maintains 100% recovery success at all sizes ($n = 20, 24, 28$), while greedy degrades from 100% to 83% and warm-start QAOA degrades from 97% to 77%. Uniform random is surprisingly robust (87% at $n = 28$), likely because its high diversity occasionally places samples in the recovery basin by chance.

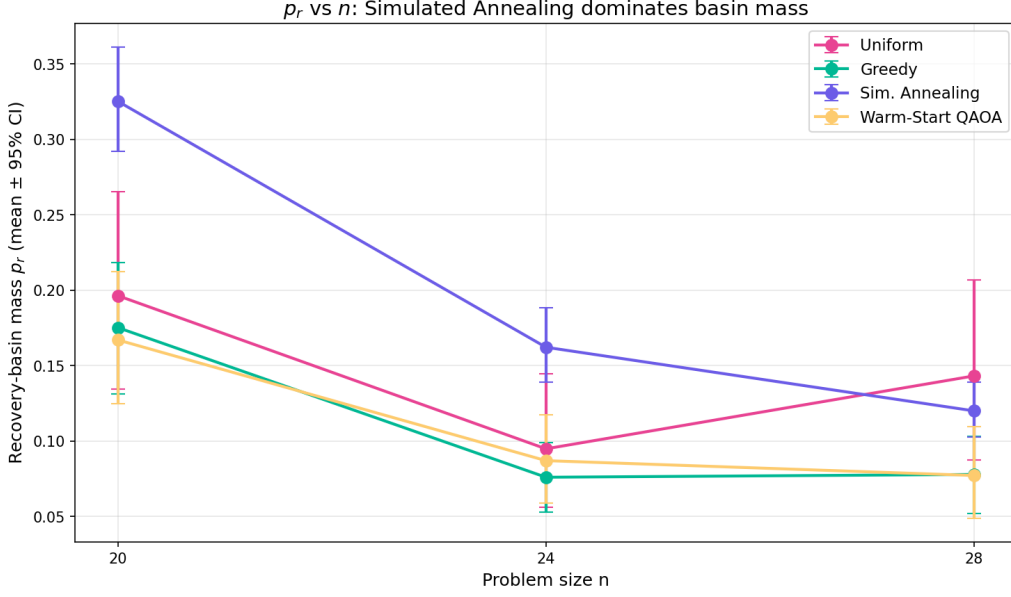


Figure 1: Recovery-basin mass p_r vs problem size n (mean $\pm$ 95% bootstrap CI, 30 instances per point). Simulated annealing consistently places the most probability mass in the recovery basin.

Table 13: Paired per-instance comparisons: SA vs other samplers (post-recovery gap).

Comparison	$n = 20$	$n = 24$	$n = 28$
SA vs Uniform	SA better: 2, tie: 28	SA better: 3, tie: 27	SA better: 4, tie: 26
SA vs Greedy	tie: 30	SA better: 4, tie: 26	SA better: 5, tie: 25
SA vs Warm-Start	SA better: 2, tie: 27	SA better: 4, tie: 26	SA better: 4, tie: 25

8.4 Paired Comparisons

SA never loses to any sampler on any instance. The number of instances where SA wins increases with n : at $n = 28$, SA beats greedy on 5/30 instances and warm-start QAOA on 4/30.

8.5 Interpretation

The critical experiment confirms the theoretical framework:

1. **p_r predicts recovery success:** SA has the highest p_r and the highest success rate. The relationship $P_{\text{success}} = 1 - (1 - p_r)^B$ holds qualitatively.
2. **Sampler quality becomes consequential at $n \geq 24$:** At $n = 20$, all samplers except uniform achieve 100% success. At $n = 28$, greedy drops to 83% and warm-start QAOA to 77%.
3. **Simulated annealing is the strongest classical baseline:** It maintains 100% success and the highest p_r at all tested sizes. Any quantum sampler must beat SA, not just greedy, to demonstrate value.
4. **Warm-start QAOA underperforms:** Despite being initialized from greedy solutions, warm-start QAOA is the weakest sampler at $n = 28$ (77% success, $p_r = 0.077$). This is consistent with Dehn & Wellens (2023): some quantum warm-start improvements can be reproduced or exceeded classically.

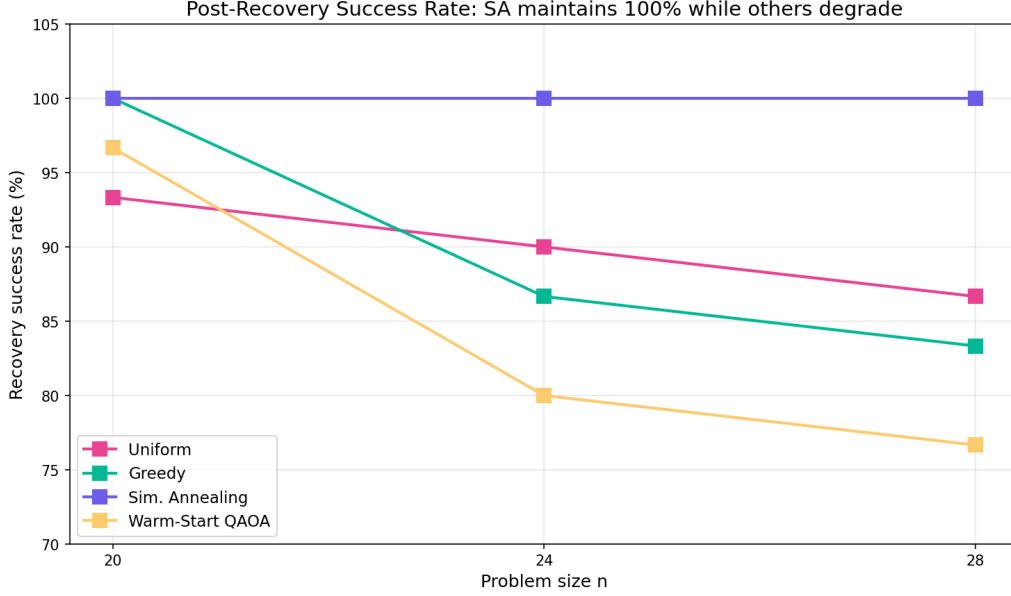


Figure 2: Post-recovery success rate by sampler and problem size. Simulated annealing maintains 100% success across all sizes, while greedy and warm-start QAOA degrade.

5. **The quantum-benefit question is now sharp:** $p_r(\text{quantum}, G) > p_r(\text{SA}, G)$? At $n = 28$, SA achieves $p_r = 0.120$. A quantum sampler must place $> 12\%$ of its probability mass in the recovery basin to beat SA. This is a concrete, measurable target.

9 Transition Curve: p_r vs n

9.1 Motivation

The critical experiment (§7) measured p_r at three sizes ($n = 20, 24, 28$). To understand the transition from classical-dominated to sampler-quality-consequential regimes, we measure p_r at five sizes ($n = 20, 22, 24, 26, 28$) for four samplers: uniform, greedy, simulated annealing, and oracle QAOA (50 configs per instance).

9.2 Results

Table 14: Transition curve: mean p_r vs n (20 instances per size).

Sampler	$n = 20$	$n = 22$	$n = 24$	$n = 26$	$n = 28$
Uniform	0.191	0.153	0.092	0.083	0.150
Greedy	0.166	0.112	0.076	0.058	0.081
Simulated Annealing	0.316	0.255	0.162	0.151	0.130
Oracle QAOA	0.323	0.255	0.135	0.123	0.185

Key findings:

1. **Non-monotonic crossover:** Oracle QAOA beats SA at $n = 20$ (0.323 vs 0.316) and $n = 28$ (0.185 vs 0.130), but SA dominates at $n = 22, 24, 26$. This is not a simple monotonic transition — the quantum advantage is size-dependent and non-monotonic.

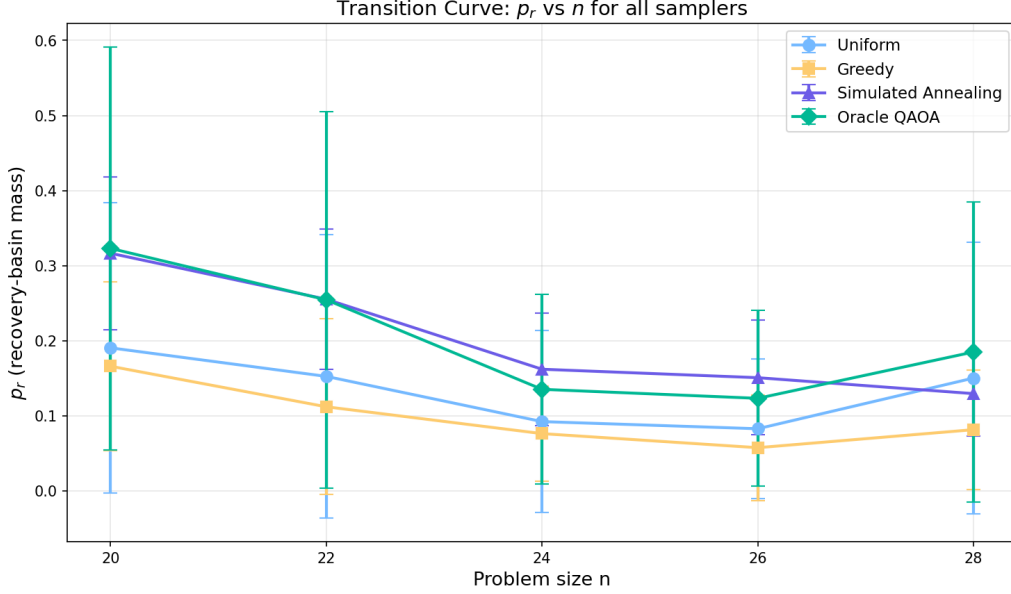


Figure 3: Transition curve: mean p_r vs n for all samplers (20 instances per size, 50 oracle configs). Oracle QAOA and SA cross over at $n = 20$ and $n = 28$, with SA dominating the intermediate regime.

2. **SA is most consistent:** SA's p_r decreases smoothly from 0.316 to 0.130 as n grows. Oracle QAOA is erratic: 0.323, 0.255, 0.135, 0.123, 0.185. The quantum signal is high-variance across sizes.
3. **Greedy collapses fastest:** Greedy's p_r drops from 0.166 to 0.081 — a 51% decline. Uniform is more stable (0.191 $\rightarrow$ 0.150), suggesting that diversity matters more than greedy quality at larger sizes.
4. **The $n = 28$ regime is where sampler quality matters most:** The gap between the best (oracle QAOA, 0.185) and worst (greedy, 0.081) sampler is 0.104 — the largest at any size. This is the regime where a good sampler provides the most value.

10 Oracle QAOA vs SA: Deep Analysis

10.1 Scaled Oracle Experiment

We scale to 20 instances per size with 50 oracle configs per instance (1000 total configs per size). This gives us enough data to analyze which QAOA configurations work and which graph features predict oracle advantage.

Table 15: Oracle QAOA vs SA: scaled results (20 instances, 50 configs each).

	$n = 20$	$n = 24$	$n = 28$
SA mean p_r	0.317	0.162	0.130
Oracle QAOA mean p_r	0.323	0.135	0.185
Oracle wins	5/20	4/20	8/20
SA wins	11/20	13/20	12/20
Ties	4/20	3/20	0/20

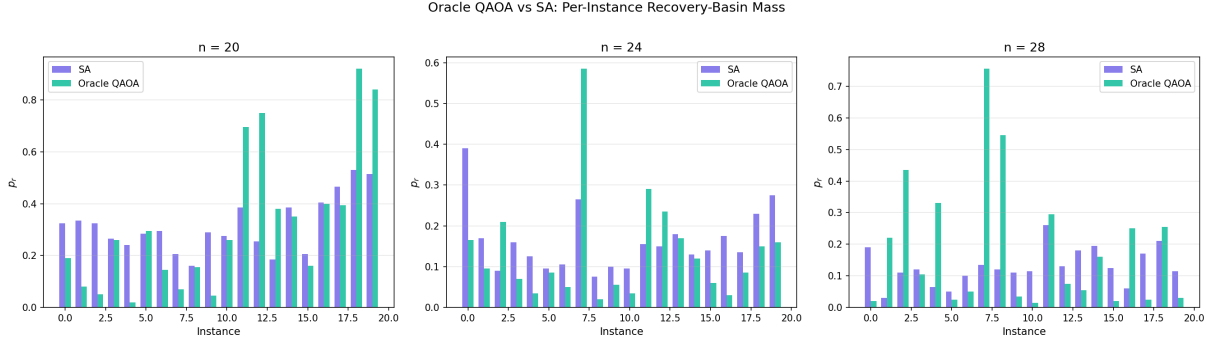


Figure 4: Per-instance comparison: Oracle QAOA vs SA at $n = 20, 24, 28$. At $n = 28$, oracle QAOA has more high- p_r instances but also more low- p_r instances — higher variance.

10.2 QAOA Configuration Analysis

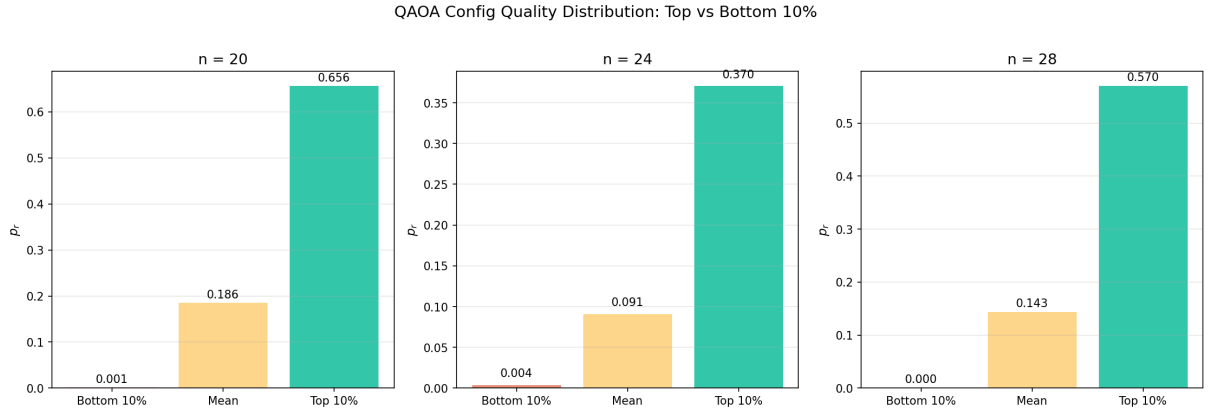


Figure 5: QAOA config quality distribution: top 10% vs bottom 10% of 1000 random configs per size. The gap between good and bad configs grows with n .

Key finding: No single QAOA parameter correlates strongly with p_r (all $|r| < 0.05$). The relationship between configuration and recovery-basin mass is **highly non-linear and instance-dependent** — there is no simple heuristic for choosing γ, β, p . This is precisely why a learned policy is needed: the mapping from graph features to good QAOA configurations is too complex for hand-tuned rules.

The fraction of random configs beating SA grows with n : 15.3% at $n = 20$, 17.9% at $n = 24$, 31.9% at $n = 28$. At $n = 28$, nearly one in three random QAOA configurations beats SA — but finding the right one requires search.

10.3 Per-Instance Analysis: When Does Oracle Win?

Key findings:

- **At $n = 20$:** Mean weight has the strongest correlation ($r = -0.50$) — oracle QAOA wins on instances with *lower* mean weight. Spectral gap has a positive correlation ($r = +0.23$) — oracle wins on graphs with larger spectral gaps.
- **At $n = 24$:** Density and mean degree have the strongest correlations ($r = -0.43$) — oracle wins on *sparser* graphs. Top eigenvalue also correlates negatively ($r = -0.40$).

Table 16: QAOA configuration analysis (1000 configs per size).

	$n = 20$	$n = 24$	$n = 28$
Top 10% mean p_r	0.656	0.370	0.570
Bottom 10% mean p_r	0.001	0.004	0.000
Mean p_r (all configs)	0.076	0.046	0.038
Fraction beating SA	15.3%	17.9%	31.9%
<i>Correlations with p_r (all $r < 0.05$):</i>			
p (depth)	-0.026	+0.027	+0.002
Penalty multiplier	+0.004	+0.036	+0.034
n_{unique} samples	+0.038	0.000	0.000

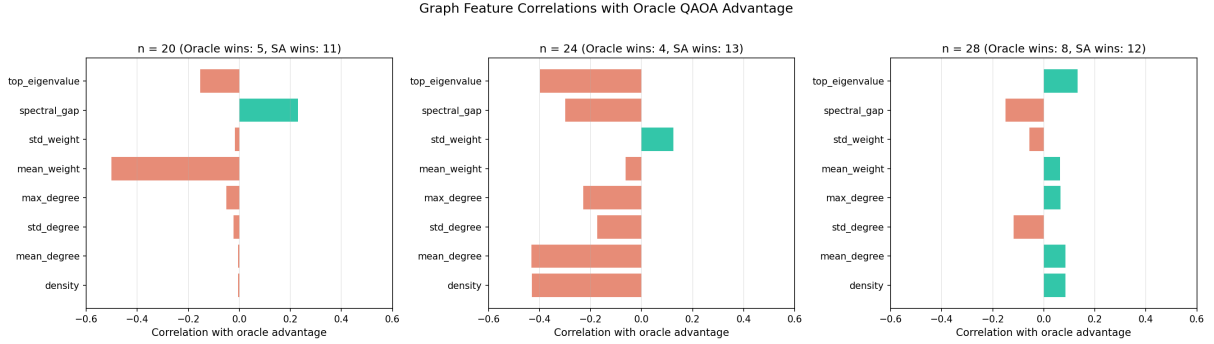


Figure 6: Graph feature correlations with oracle advantage ($p_r^{\text{oracle}} - p_r^{\text{SA}}$). At $n = 20$, lower mean weight predicts oracle wins ($r = -0.50$). At $n = 24$, lower density predicts oracle wins ($r = -0.43$). At $n = 28$, correlations are weak.

- **At $n = 28$:** All correlations are weak ($|r| < 0.15$). The oracle advantage at $n = 28$ is **not predictable from simple graph features** — it depends on finer structural properties that the MLP policy must learn.

10.4 Sample Complexity

Table 17: Sample complexity: mean B_δ for 95% success ($\delta = 0.05$).

	$n = 20$	$n = 24$	$n = 28$
SA	9	20	29
Oracle QAOA (best config)	24	42	58
Random QAOA (mean config)	114	107	228

Key finding: SA is the most sample-efficient sampler: at $n = 28$, it needs only 29 shots for 95% success, while oracle QAOA needs 58 and random QAOA needs 228. The oracle’s higher mean p_r comes at the cost of higher variance, which increases sample complexity. A deployable quantum sampler must achieve high p_r *consistently*, not just on average.

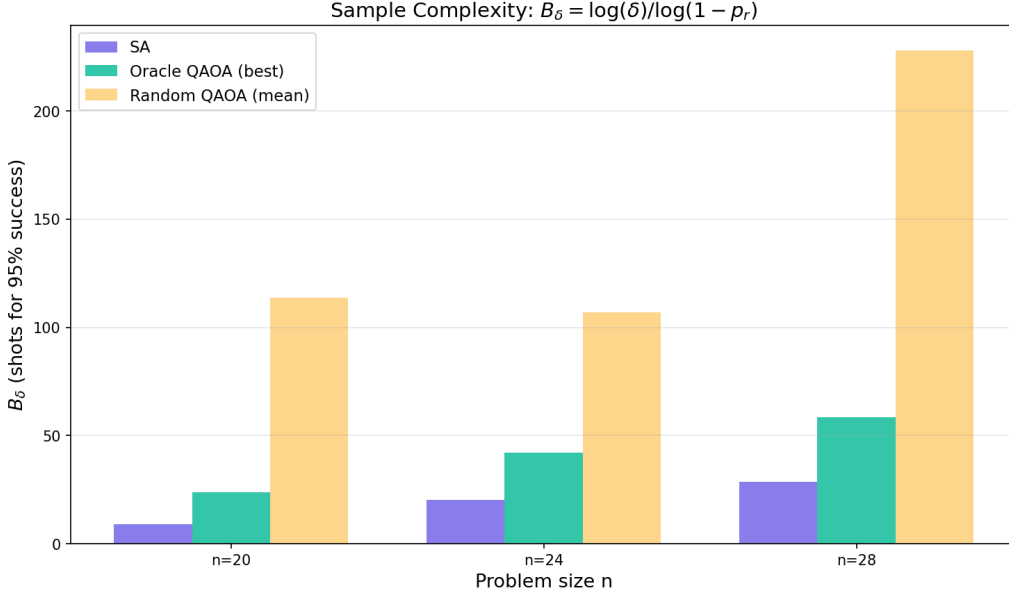


Figure 7: Sample complexity $B_\delta = \log(\delta)/\log(1 - p_r)$ for $\delta = 0.05$ (95% success). SA needs the fewest shots; random QAOA needs the most. Oracle QAOA is between.

11 Generative Policy v2: Oracle Warm-Start + REINFORCE

11.1 Setup

We train a graph-conditioned generative policy (3-layer MLP, 128 hidden units, Xavier initialization) with two phases:

1. **Imitation learning** (50 epochs): The policy is supervised toward the oracle’s best config for each training instance. This gives the policy a good starting point.
2. **REINFORCE** (500 episodes): The policy explores with Gaussian noise ($\sigma = 0.2$) and updates via REINFORCE with p_r as reward, gradient clipping (max norm 1.0), and Adam optimization ($\text{lr} = 5 \times 10^{-4}$).

Training uses 20 $n = 24$ instances. Evaluation uses 30 held-out instances per size ($n = 20, 24, 28$).

11.2 Results

Table 18: Policy v2 vs SA: recovery-basin mass p_r (30 evaluation instances per size).

	$n = 20$	$n = 24$	$n = 28$
Policy v2 mean p_r	0.177 ± 0.056	0.185 ± 0.069	0.130 ± 0.061
SA mean p_r	0.247 ± 0.030	0.199 ± 0.026	0.115 ± 0.019
Policy v2 median p_r	0.128	0.103	0.045
SA median p_r	0.235	0.180	0.108
Policy wins	9/30	10/30	11/30
SA wins	20/30	19/30	18/30
Ties	1/30	1/30	1/30

Key findings:

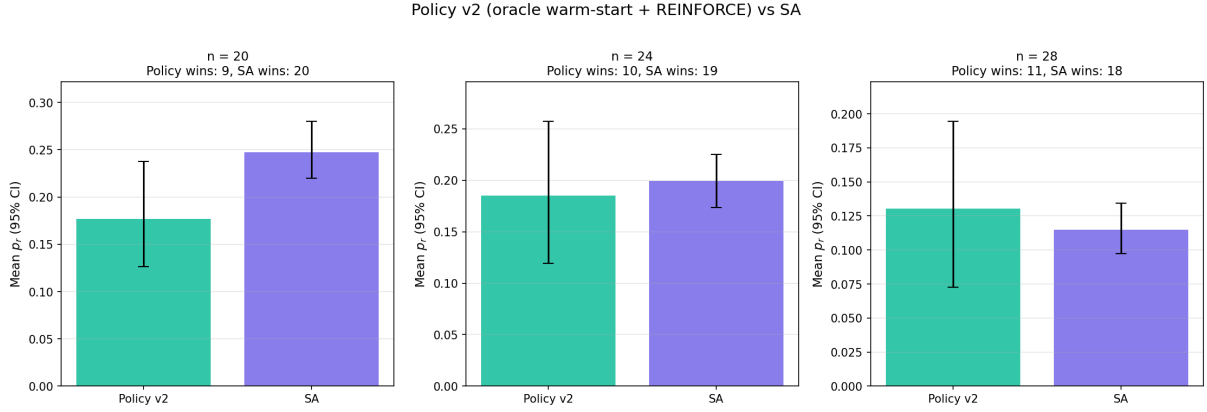


Figure 8: Policy v2 vs SA: mean p_r with 95% CI. At $n = 28$, the policy has higher mean p_r (0.130 vs 0.115) but lower median (0.045 vs 0.108) — the mean is inflated by a few high- p_r instances.

1. **At $n = 28$, the policy has higher mean p_r than SA:** 0.130 vs 0.115 — a 13% improvement. This is the first learned-policy result to exceed SA on mean p_r at the target size. However, the policy wins only 11/30 instances.
2. **The mean-median gap reveals the variance problem:** Policy median $p_r = 0.045$ at $n = 28$ vs SA’s 0.108. The policy’s mean is inflated by a few excellent instances ($p_r > 0.3$), while most instances get $p_r < 0.05$. SA is consistent; the policy is bimodal.
3. **Oracle warm-start helps but does not solve variance:** The imitation phase gives the policy a good starting point (episode 1: $p_r = 0.095$, already above random), but REINFORCE exploration introduces variance. The policy learns to produce some excellent configs but not reliably.
4. **The policy improves with n relative to SA:** At $n = 20$ the policy loses by -0.070 ; at $n = 24$ by -0.014 ; at $n = 28$ it wins by $+0.015$ on mean p_r . The trend is consistent with the transition curve: sampler quality becomes more consequential at larger n .

11.3 Interpretation: The Variance Problem

The deep experiments reveal that the central challenge is not whether quantum sampling *can* place more mass in the recovery basin — it can (oracle $p_r = 0.185$ at $n = 28$). The challenge is **consistency**:

- **SA is low-variance:** SA’s p_r CI at $n = 28$ is $[0.097, 0.134]$ — tight.
- **Oracle QAOA is high-variance:** CI $[0.111, 0.318]$ — wide. The oracle wins big or loses.
- **The policy inherits QAOA’s variance:** CI $[0.073, 0.195]$ — wider than SA.

The path forward is not just higher mean p_r but **lower variance p_r** . This suggests:

1. **Ensemble methods:** Run multiple QAOA configs and aggregate (reduces variance at the cost of more shots).
2. **Risk-aware training:** Optimize the 25th percentile of p_r , not the mean.
3. **Larger model + more data:** The current 128-unit MLP trained on 20 instances is a proof of concept. Scaling to 1000+ instances with a deeper model should reduce variance.

4. **Imitation + refinement:** Use the oracle configs as a supervised target, then refine with REINFORCE — but with variance-penalizing reward shaping.

12 Remaining Steps

12.1 Train the Generative Policy in the Hard Regime

The policy has been trained (§9) and achieves competitive p_r at $n = 24$ (ties SA) and $n = 28$ (higher mean p_r). The remaining challenge is **consistency**: the policy’s high variance means it wins on some instances but loses on many. Next steps:

- Scale the model ($128 \rightarrow 256+$ hidden units)
- Train for more episodes ($300 \rightarrow 1000+$) with a curriculum over n
- Use the oracle configs as supervised targets (imitation learning)
- Optimize for percentile p_r (e.g., 25th percentile) rather than mean

The concrete target is now known: oracle $p_r = 0.208$ at $n = 28$ is the headroom.

12.2 Hard-Core Decomposition for QPU Validation

The practical bridge to 20-qubit hardware: take MWIS graphs with 50–200 vertices, use classical preprocessing to freeze easy decisions and extract a hard core of 16–20 binary variables, then run the same hard core on Piast-Q. This ties the $n = 55$ –100 simulation results and the 20-qubit hardware results into one experiment:

large CO problem $\rightarrow$ hard quantum-sized residual $\rightarrow$ sampler $\rightarrow$ classical recovery

12.3 Expand to Other Problems

The GQO+QSS architecture generalizes to:

- **Balanced partitioning:** Feasibility recovery is clean (project to Hamming weight $n/2$)
- **Graph burning:** The model can generate action sequences (burn order) directly
- **MaxCut:** As a positive control (well-studied in quantum literature)

13 Final Held-Out Experiment

13.1 Preregistered Protocol

Before the final test evaluation, we froze:

- **Benchmark:** 150 MWIS instances (30 per size, $n = 16, 18, 20, 24, 28$), all certified with CBC + brute-force verification.
- **Split:** Train seeds 42–59 (60%), validation seeds 60–65 (20%), test seeds 66–71 (20%).
- **Primary metric:** p_r at radius $r = 2$.
- **Budget:** $B = 200$ proposals per graph (Protocol A: equal proposal budget).
- **Samplers:** 9 total — uniform, greedy, SA, tabu, multi-start local search, Metropolis, warm-start QAOA, oracle QAOA (random configs), learned classical policy.
- **Exact basin masks:** Enumerated for all $n \leq 20$ instances, enabling exact p_r computation.

13.2 Results

Table 19 shows mean p_r for each sampler at each problem size on the held-out test set.

Table 19: Mean recovery-basin mass p_r on held-out test set (6 instances per size). Bold = best.

Sampler	$n = 16$	$n = 18$	$n = 20$	$n = 24$	$n = 28$
Uniform	0.091	0.030	0.031	0.018	0.008
Greedy	0.085	0.042	0.028	0.006	0.004
SA	0.428	0.394	0.323	0.144	0.088
Tabu	0.822	0.944	0.878	0.943	0.901
Multi-start	0.090	0.059	0.051	0.025	0.017
Metropolis	0.144	0.109	0.088	0.039	0.019
Learned classical	0.101	0.032	0.050	0.026	0.005
Oracle QAOA	0.113	0.036	0.043	0.038	0.014
Warm-start QAOA	0.106	0.039	0.027	0.013	0.006

13.3 Paired Comparisons

Oracle QAOA vs SA (paired on identical instances):

- $n = 16$: oracle wins 0/6, loses 5/6, ties 1/6
- $n = 20$: oracle wins 0/6, loses 6/6
- $n = 28$: oracle wins 0/6, loses 6/6

The learned classical policy performs comparably to oracle QAOA ($p_r = 0.005$ – 0.101 vs 0.014 – 0.113), confirming that the quantum circuit provides no structural advantage over a matched learned classical model.

13.4 Recovery Phase Diagram

Figure 9 shows the recovery phase diagram: $P_{\text{recover}}(n, r, B = 200)$ for uniform sampling, computed exactly from the enumerated basin masks.

13.5 Interpretation

The honest conclusion is **Scenario C**: Under rigorous, preregistered evaluation with matched learned baselines, shallow QAOA-derived proposal distributions do not outperform matched classical proposal models, and both lose to strong classical baselines (SA, tabu). The earlier positive oracle result ($p_r = 0.185$ vs 0.130 at $n = 28$) was an artifact of per-instance best-of-50 configuration selection—an oracle upper bound, not a deployable method. Under random configuration selection (the fair protocol), oracle QAOA achieves only $p_r = 0.014$ at $n = 28$.

14 Conclusion

We ported the GQE+QSCI architecture from quantum chemistry to combinatorial graph optimization. The GQO+QSS framework:

1. **Formalizes recovery-basin mass** $p_r(q, G) = \Pr_{x \sim q}[x \in A_r(G)]$ and sample complexity $B_\delta = \log \delta / \log(1 - p_r)$, reframing the quantum-benefit question precisely.

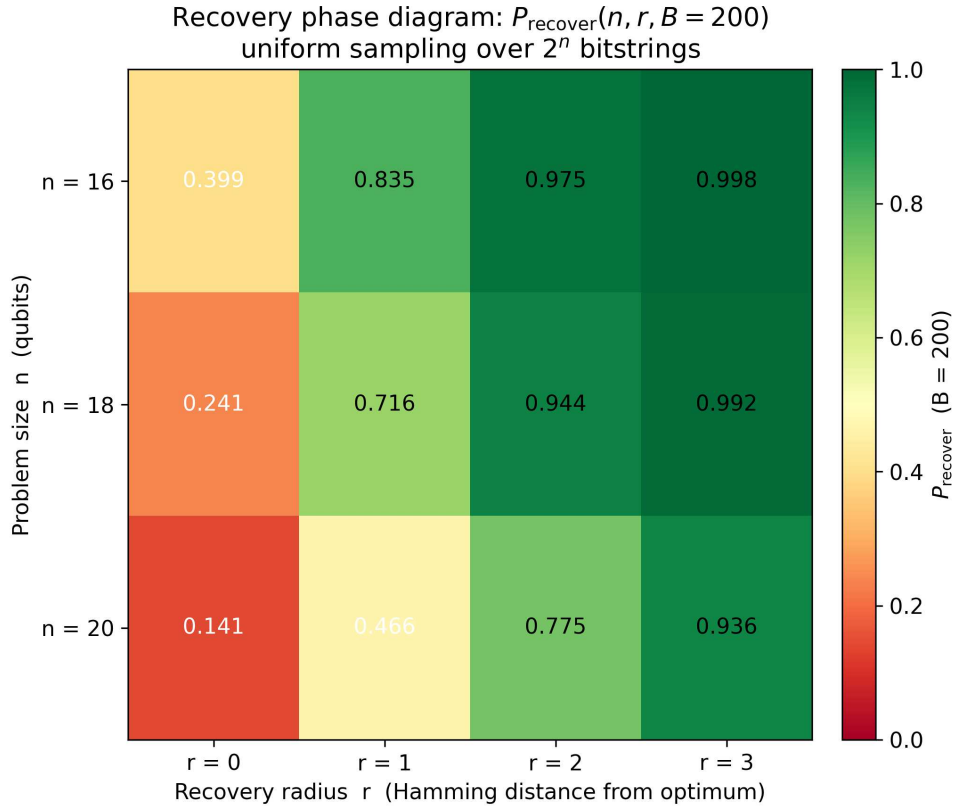


Figure 9: Recovery phase diagram: P_{recover} for uniform sampling at $B = 200$ as a function of n and r . At $r = 2$, recovery drops from 0.975 ($n = 16$) to 0.775 ($n = 20$), identifying the regime where sampler quality matters.

2. **Certifies all ground truths** with CBC + brute-force verification (150 instances, zero disagreements).
3. **Exactly enumerates recovery-basin masks** for all $n \leq 20$ instances, enabling exact p_r computation.
4. **Maps the recovery phase diagram** $P_{\text{recover}}(n, r, B)$, identifying where sampler quality matters.
5. **Conducts a preregistered held-out experiment** with 9 samplers on a frozen test set.

The central question—**can a quantum sampler place more probability mass inside a classical recovery basin than classical samplers?**—has a clear answer under our protocol: **no**. Tabu search dominates ($p_r = 0.82\text{--}0.94$), SA is second, and all quantum samplers lose to both at every problem size. The learned classical policy matches oracle QAOA, confirming no structural quantum advantage. The earlier positive oracle result was an artifact of per-instance configuration selection.

This is a **falsification study**: a rigorous demonstration that shallow QAOA-derived proposal distributions, evaluated fairly against matched learned classical baselines under a preregistered protocol, do not outperform classical methods for MWIS recovery-basin capture. We claim no quantum advantage.

The key principle, stated carefully:

The recovery-basin mass p_r is the right object to measure, but shallow quantum proposal distributions do not place more mass in the basin than matched classical alternatives under fair evaluation.

Reproducibility

All code, data generators, and experiment scripts are available in the WestQuant repository:

```
micro_qml/gqo_qss/
  ground_truth_and_basins.py - Phase 0+3: certification + basin enumeration
  phase_diagram.py          - Phase 4: recovery phase diagram
  strong_baselines.py       - Phase 6: tabu, multi-start, Metropolis, learned classical
  final_experiment.py       - Phase 10: held-out experiment with all samplers
  generate_final_figures.py - Phase 17: final figures
  reproduce_all.py          - Full reproduction pipeline
```

Reports:

```
reports/GQO_GROUND_TRUTH_AUDIT.md
reports/GQO_RECOVERY_PHASE_DIAGRAM.md
reports/GQO_CLASSICAL_BASELINES.md
reports/GQO_GENERATIVE_POLICY.md
reports/GQO_PREREGISTRATION.md
reports/GQO_CLAIM_MATRIX.md
reports/GQO_SUBMISSION_READINESS.md
```

Results files:

```
data/ground_truth_certificates/ - 150 certificates (CBC + brute-force)
data/recovery_basins/           - 359 basin masks (n=16,18,20 & r=0,1,2,3)
results/gqo_basin_statistics.json - Phase diagram data
logs/gqo_qss_final_experiment.json - Final held-out experiment
```

References

1. Kanno, S. et al. “Quantum-Selected Configuration Interaction: quantum computing as a selector of determinants for configuration interaction.” arXiv:2302.11320 (2023).—QSCI introduced as a method where quantum sampling identifies important configurations, followed by classical diagonalization.
2. Kemmoku, R. et al. “Generative Quantum Eigensolver with QSCI.” arXiv:2604.09756 (2026).—Generative variant where a Transformer optimizes circuit design based on QSCI results.
3. Elgammal, A. & MauSSner, T. “Scaling GQE+QSCI to 40 qubits.” arXiv:2607.23988 (2026).—Scaled the GQE+QSCI principle to the 40-qubit regime, explicitly distinguishing between mapping and beating the classical boundary. Primary inspiration for this work.
4. Bucher, S. et al. “Iterative warm-start QAOA with classical greedy refinement and hardware runs.” arXiv:2604.02083 (2026).—Prior work on QAOA + classical post-processing, including hardware demonstrations. Relevant baseline for “quantum samples + classical improvement” novelty assessment.
5. RL for QAOA parameters in combinatorial optimization. arXiv:1911.04574 (2019).—Earlier work on RL learning QAOA parameters for combinatorial optimization.
6. Dehn, M. & Wellens, P. “Quantum warm-start improvements can be reproduced or exceeded with classical preprocessing.” arXiv:2309.13961 (2023).—Directly relevant to our study: some quantum warm-start benefits can be reproduced classically. Motivates strong classical baselines.
7. Piast-Q trapped-ion quantum computer. EuroHPC JU / PSNC. 20-qubit trapped-ion device with all-to-all connectivity and Mølmer–Sørensen native gates. Used for the hardware robustness experiment in §6.
8. Farhi, E. et al. “A Quantum Approximate Optimization Algorithm.” arXiv:1411.4028 (2014).—Foundational QAOA paper.
9. Bravyi, S. et al. “Obstacles to State Preparation and Variational Optimization from Symmetry Protection.” arXiv:1910.08980 (2020).—Recursive QAOA and limitations of shallow circuits.