BOLT-DA: A SHOT-EFFICIENT DEPTH-AWARE QAOA OPTIMIZER FOR MAXCUT

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimizing the Quantum Approximate Optimization Algorithm (QAOA) under a limited measurement budget requires allocating samples to both parameter search and candidate comparison. We propose BOLT-DA, a depth-aware MaxCut optimizer that separates branch-specific search continuation from reuse of previously selected solutions. Target-graph and reduced-donor searches supply candidates, while interpolation and identity padding retain information from the preceding depth. Fresh target-side measurements select from this constant-size pool within an explicitly accounted raw-shot budget. On held-out AIDS, Linux, and IMDb graphs, BOLT-DA achieves higher mean approximation ratios (AR) and lower normalized-gap mean squared errors (MSE) than Red-QAOA, Neural QAOA^2, and an MG-inspired multi-angle control in all 15 dataset–depth settings at $q = 32$. At $p = 5$, MSE is 32.33%–81.00% lower than Neural QAOA^2. Equal-budget ablations identify identity padding as the principal source of candidate-reuse gains. A second target search improves quality on two datasets, whereas the reduced donor lowers logical circuit cost. A fixed-pool diagnostic detects no additional quality gain from the empirical selection guard. Experiments on 12-, 14-, and 16-node graphs retain leading native-solution means across 15 size–depth settings, and a five-node hardware case study demonstrates the method's practical execution behavior.

1 INTRODUCTION

MaxCut is a canonical NP-hard graph problem with mature classical approximation theory and a direct Ising representation (Goemans & Williamson, 1995; Lucas, 2014). QAOA addresses such problems by alternating problem and mixing evolutions with classically optimized angles (Zhou et al., 2020). Its shallow, problem-structured circuits motivate near-term quantum optimization, where trainability and measurement cost matter alongside expressivity (Preskill, 2018; Cerezo et al., 2021a).

Greater QAOA depth can improve attainable objectives but adds variational dimensions, two-qubit gates, and noisy objective evaluations. The challenge is not only depth-p expressivity, but reliable parameter search under a fixed execution budget despite reachability limits, flat or deceptive landscapes, hardware noise, and finite-shot fluctuations (Akshay et al., 2020; McClean et al., 2018; Cerezo et al., 2021b; Wang et al., 2021; Bravyi et al., 2020; Kuebler et al., 2020; Arrasmith et al., 2021; Scriva et al., 2024). Low-shot optimization must therefore decide both where to search and how to allocate measurements.

Parameter transfer and depth-wise reuse exploit regularities across instances and depths; learning-assisted methods provide graph-dependent initializations or circuit structures; and graph reduction lowers the quantum footprint by solving or exploring smaller surrogate problems. These signals are complementary but imperfect: sampling noise can mis-rank transferred or interpolated points, and reduced-graph rankings need not transfer to the original objective. Direct target search, reduced-graph transfer, and depth-wise reuse must therefore be reconciled within a limited measurement budget.

BOLT-DA (Budgeted Original-Objective Layerwise Triage with Depth Awareness) spends most shots on the original graph, uses a reduced graph only as a donor, and reserves a small budget

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.

for fresh original-graph arbitration over a constant-size candidate pool. This auditable finite-shot depth ladder integrates direct optimization, reduced-graph transfer, and depth-wise reuse without dense landscape reconstruction. Figure 1 summarizes the problem setting, finite-shot challenges, and workflow.

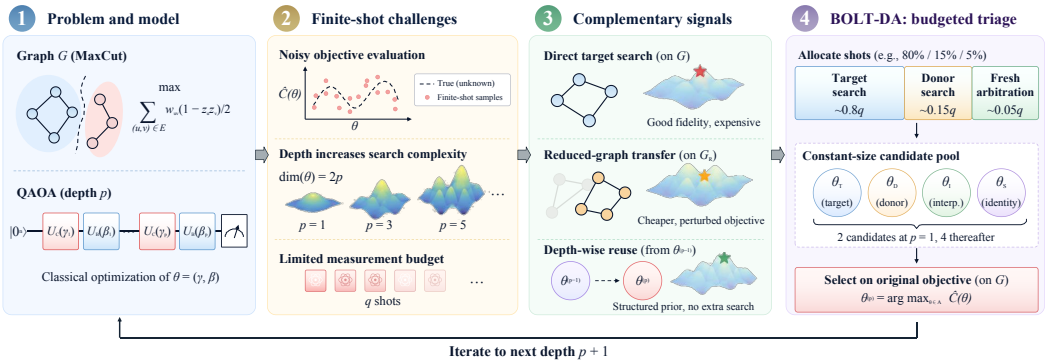

Figure 1: Overview of BOLT-DA: branch-specific search continuation, structured reuse candidates, and fresh target-side selection under a per-depth measurement budget.

Our main contributions are:

- **Separate continuation and reuse.** Our depth-aware framework warm-starts target and donor branches from their own previous outputs; the previously selected solution supplies interpolation and identity-padding candidates. The constant-size pool retains an equal-value candidate in the ideal circuit model without forcing either branch to follow the selected trajectory.
- **Joint search–selection budgeting.** A prescribed per-depth raw-shot budget covers target search, reduced-donor search, and fresh original-graph candidate comparison. Selection uses original-graph measurements, not donor rankings or cached search estimates.
- **Evaluation and component attribution.** We compare BOLT-DA with Red-QAOA, Neural QAOA^2, and an MG-inspired multi-angle control on three held-out datasets. Equal-budget ablations, graph-paired intervals, and post-hoc diagnostics identify identity padding as the main reuse mechanism and characterize the donor's quality–logical-cost trade-off; further tests cover 12-, 14-, and 16-node graphs and a hardware case study.

2 Related Work

QAOA for MaxCut and finite-shot optimization. Analytical, numerical, and hardware studies characterize QAOA performance and depth-dependent costs (Wang et al., 2018; Guerreschi & Matsuura, 2019; Willsch et al., 2020; Zhou et al., 2020; Pagano et al., 2020; Harrigan et al., 2021; Lotshaw et al., 2021; Basso et al., 2022). Shot-noise analysis and measurement-frugal optimization address sampling efficiency (Kuebler et al., 2020; Scriva et al., 2024); related variational methods reuse observations for uncertainty-aware gradients (Pedrielli et al., 2026), predict parameter updates (Huang et al., 2025), or freeze inactive parameters (Peng et al., 2025). Efficient updates alone do not budget final candidate comparison; BOLT-DA budgets both search and fresh target-side selection.

Parameter structure and transfer. Fixed angles, transfer, initialization, multistart, symmetry, and layerwise training exploit QAOA structure (Akshay et al., 2021; Wurtz & Lykov, 2021; Galda et al., 2021; Shaydulin et al., 2023; Galda et al., 2023; Falla et al., 2024; Sack & Serbyn, 2021; Egger et al., 2021; Shaydulin et al., 2019; Shaydulin & Wild, 2021; Campos et al., 2021; Lee et al., 2024). Recursive greedy initialization establishes ideal depth-wise improvement (Sack et al., 2023). These benefits do not ensure correct finite-shot rankings; BOLT-DA separates branch-specific warm starts

from freshly compared interpolation and identity-padding candidates derived from the previously selected solution.

Learning and circuit adaptation. Learning-based methods predict parameters or support graph solvers (Alam et al., 2020; Jain et al., 2022; Moussa et al., 2022; Amosy et al., 2024; Liang et al., 2024; Bach et al., 2024). MG-Net generates task- and depth-dependent mixers (Qian et al., 2024); Neural QAOA^2 jointly generates partitions and initial parameters (Zheng et al., 2026); multi-angle and adaptive-bias methods modify the ansatz (Herrman et al., 2022; Yu et al., 2022). ASAP separates learned action proposal and selection in classical combinatorial optimization (Fang et al., 2026), unlike our shot-budgeted QAOA candidate comparison. Better proposals or expressivity alone do not resolve noisy rankings; fixed-pool diagnostics detect no additional quality gain from our empirical anchor guard.

Graph reduction and decomposition. Reduction, decomposition, and circuit optimization address quantum-resource costs (Li et al., 2023; Zhou et al., 2023; Ponce et al., 2025; Bae & Lee, 2024; Bechtold et al., 2023). Red-QAOA transfers reduced-graph parameters before target refinement (Wang et al., 2024), which may not overcome poor donor initialization under limited shots. BOLT-DA instead retains a separate target trajectory and treats the donor output as an optional candidate, not a required initializer.

3 Problem Statement and Background

Let $G = (V, E)$ be an undirected, unweighted graph with vertex set $V = \{1, \dots, n\}$ and $m = |E| > 0$ edges. For each vertex $v \in V$, let $z_v \in \{-1, +1\}$ encode its membership in one of the two parts, and collect these variables in $z = (z_1, \dots, z_n)^\top$. For an edge $(u, v) \in E$, z_u and z_v are the variables associated with its endpoints. MaxCut is the binary quadratic program (Goemans & Williamson, 1995)

$$\max_{z} \quad \frac{1}{2} \sum_{(u,v)\in E} (1 - z_u z_v) \tag{1a}$$

$$\text{s.t.} \quad z_v \in \{-1, +1\}, \quad \forall v \in V. \tag{1b}$$

The sum counts each undirected edge once. The objective in Eq. (1a) is denoted by $C_G(z)$, and the optimal value of problem (1) by $C_G^\star$. Associate each vertex with a qubit. Let Z_v denote the Pauli-Z operator on qubit v and I the identity operator. The cost Hamiltonian is $\widehat{C}_G = \frac{1}{2} \sum_{(u,v)\in E} (I - Z_u Z_v)$. Computational-basis outcomes $|0\rangle$ and $|1\rangle$ encode spins $+1$ and -1, respectively.

At integer depth $p \geq 1$, QAOA is parameterized by $\theta_p = (\boldsymbol{\beta}, \boldsymbol{\gamma})$, where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$ and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)$ are the mixing and cost angles. The prescribed search domain is $\boldsymbol{\Theta}_p = [0, \pi)^{2p}$. With the rotation conventions $\mathrm{RX}(\alpha) = e^{-\mathrm{i}\alpha\sigma_x/2}$ and $\mathrm{RZZ}_{uv}(\alpha) = e^{-\mathrm{i}\alpha Z_u Z_v/2}$, where σ_x is the single-qubit Pauli-X matrix, the layer operators and the resulting state are

$$U_C^G(\gamma) = \prod_{(u,v)\in E} \mathrm{RZZ}_{uv}(2\gamma), \tag{2a}$$

$$U_B(\beta) = \prod_{v\in V} \mathrm{RX}_v(\beta), \tag{2b}$$

$$|\psi_G^{(p)}(\theta_p)\rangle = U_B(\beta_p) U_C^G(\gamma_p) \cdots U_B(\beta_1) U_C^G(\gamma_1) |+\rangle^{\otimes n}. \tag{2c}$$

Here, $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, and operators act from right to left, with cost evolution preceding mixing in each layer (Farhi et al., 2014; Zhou et al., 2020). At fixed depth p, parameter optimization maximizes the expected cut value over $\theta_p \in \boldsymbol{\Theta}_p$. The population objective is

$$J_G(\theta_p) = \langle \psi_G^{(p)}(\theta_p) | \widehat{C}_G | \psi_G^{(p)}(\theta_p) \rangle. \tag{3}$$

During training, the optimizer accesses sampled cut values rather than J_G itself. At a parameter setting $\theta \in \boldsymbol{\Theta}_p$, each computational-basis measurement yields a spin assignment whose cut value is a sample $X \in [0, m]$. A batch of s shots provides the estimate

$$\widehat{J}_G(\theta; s) = s^{-1} \sum_{r=1}^{s} X_r, \tag{4}$$

where X_r is the cut value from the rth measurement and s is the shot count. Neither $C_G^\star$ nor an exact statevector expectation is available to the optimizer.

4 THE BOLT-DA METHOD

4.1 POLICY FORMULATION

Let $B_p = cp$ denote the nominal number of target-search calls at depth p, where c is the calls factor, and let q denote the nominal number of shots per call. The incremental online-shot cap is

$$S_p = B_p q = cpq. \tag{5}$$

The task is to return $\widehat{\theta}_p \in \mathbf{\Theta}_p$ using at most S_p fresh training samples. Parameter information is obtained from three imperfect sources: direct search on G, search on a reduced donor graph $\widetilde{G}$, and the selected trajectory from depth $p-1$. The donor may identify a useful parameter basin, but its ranking need not transfer to the target graph:

$$J_{\widetilde{G}}(\theta_a) \geq J_{\widetilde{G}}(\theta_b) \quad \not\Rightarrow \quad J_G(\theta_a) \geq J_G(\theta_b). \tag{6}$$

Final selection must therefore rely on fresh samples from the original target graph rather than on a reconstructed parameter landscape.

Finite-shot depth-ladder problem. Let $\mathcal{O}_H(\theta, s)$ denote an oracle that returns s independent cut values obtained by measuring the depth-p QAOA state for graph H at parameters θ. An admissible policy Π may adaptively query $\mathcal{O}_G$ and $\mathcal{O}_{\widetilde{G}}$, use its preceding outputs $\widehat{\theta}_{1:p-1}$, and perform arbitrary classical computation. At depth p, its oracle transcript must satisfy

$$\sum_{(H,\theta,s)\in\mathcal{T}_p(\Pi)} s \leq S_p. \tag{7}$$

Here, $\mathcal{T}_p(\Pi)$ denotes the oracle transcript generated by Π at depth p. The policy returns $\widehat{\theta}_p = \Pi(G, \widetilde{G}, \mathcal{T}_p, \widehat{\theta}_{1:p-1})$. For a graph distribution $\mathcal{D}$ and optimizer randomness ξ, the policy-design objective is

$$\min_{\Pi \text{ admissible}} \mathbb{E}_{G\sim\mathcal{D},\xi}\left[\left(1 - \frac{J_G(\widehat{\theta}_p)}{C_G^\star}\right)^2\right]. \tag{8}$$

Equation (8) specifies the evaluation target, not an oracle available during training. The quantities J_G and $C_G^\star$ are withheld until the returned parameters are frozen. The policy must therefore allocate its limited samples among exploration, transfer, and target-side verification.

Evaluation after optimization. For the expected-value experiments, let $P_{i,s,p}^{(a)}$ denote method a's final output distribution over assignments on the original graph G_i, and write

$$\mathcal{J}_{i,s,p}^{(a)} = \mathbb{E}_{z\sim P_{i,s,p}^{(a)}}[C_{G_i}(z)]. \tag{9}$$

For BOLT-DA, this equals $J_{G_i}(\widehat{\theta}_{i,s,p}^{(a)})$. The reporting notation accommodates the compared methods' respective outputs without imposing a shared parameterization. After optimization has concluded, define

$$\begin{aligned} \mathrm{AR}_{i,s,p}^{(a)} &= \frac{\mathcal{J}_{i,s,p}^{(a)}}{C_{G_i}^\star}, & g_{i,s,p}^{(a)} &= 1 - \mathrm{AR}_{i,s,p}^{(a)}, \\ L_{i,p}^{(a)} &= \frac{1}{N_S}\sum_{s=1}^{N_S}\left(g_{i,s,p}^{(a)}\right)^2, & \mathrm{MSE}_p^{(a)} &= \frac{1}{N_G}\sum_{i=1}^{N_G} L_{i,p}^{(a)}. \end{aligned} \tag{10}$$

For comparator b, the relative normalized-gap MSE improvement is

$$\Delta_p(a;b) = 1 - \frac{\mathrm{MSE}_p^{(a)}}{\mathrm{MSE}_p^{(b)}}. \tag{11}$$

A positive $\Delta_p(a;b)$ indicates lower MSE for a. AR and MSE measure returned-solution quality, not discrepancies between parameter landscapes. Graph instances are the independent units of analysis; optimizer seeds are repeated runs on each graph. The native-solution extension in Section 5.5 uses separately defined solution-level metrics.

4.2 RAW-SHOT ALLOCATION

Let α_T, α_D, and α_R be positive allocation coefficients for target search, donor search, and target-side selection, respectively, with $\alpha_T + \alpha_D + \alpha_R = 1$. The BOLT-DA policy divides the per-depth budget in Eq. (5) as $S_p^T = \lfloor \alpha_T S_p \rfloor$, $S_p^D = \lfloor \alpha_D S_p \rfloor$ and $S_p^R = S_p - S_p^T - S_p^D$. Here, S_p^T and S_p^D are the search budgets, and S_p^R is reserved for fresh candidate evaluation on the original graph. Assigning the rounding remainder to selection preserves exact shot conservation. Thus, α_R specifies the nominal selection share, while S_p^R is its integer-valued allocation.

A seeded reducer constructs a single donor graph $\widetilde{G} = \mathcal{R}_\tau(G; s)$, which is reused across depths. Here, s is the reduction seed, and τ specifies the average-degree retention criterion rather than a fixed vertex-retention ratio. Let $\mathrm{ShotDE}(H, p, X, b)$ denote fixed-budget noisy differential evolution on graph H at depth p, with X total samples and b nominal calls. Let $\rho \in (0, 1]$ control the donor call budget relative to B_p. The search outputs are $\theta_p^T = \mathrm{ShotDE}(G, p, S_p^T, B_p)$ and $\theta_p^D = \mathrm{ShotDE}(\widetilde{G}, p, S_p^D, \lfloor \rho B_p \rfloor)$.

The donor shot budget is controlled by α_D, whereas ρ controls its nominal call count. Ignoring integer rounding, the nominal allocations correspond to $\alpha_T q$ shots per target call and $(\alpha_D/\rho)q$ shots per donor call.

The allocation coefficients, donor call ratio, and reduction parameter are chosen empirically for the problem setting and held fixed throughout the depth ladder. Their values must provide valid shot and call budgets for both ShotDE branches at every depth; in particular, integer rounding must not eliminate either search branch. The residual budget S_p^R is passed to the candidate-selection procedure defined below.

For $p > 1$, the target and donor branches are warm-started from interpolations of their respective preceding outputs θ_{p-1}^T and θ_{p-1}^D. The previously selected solution $\widehat{\theta}_{p-1}$ is used separately to construct the interpolation and identity-padding candidates in the next subsection.

4.3 A CONSTANT-SIZE STRUCTURED CANDIDATE POOL

At $p = 1$, the candidate pool is

$$\mathcal{A}_1 = [\theta_1^T, \theta_1^D]. \tag{12}$$

For $p > 1$, construct two candidates from $\widehat{\theta}_{p-1}$. For each parameter block $x = (x_1, \ldots, x_{p-1})$, set $x_0 = x_p = 0$. The zero-endpoint interpolation is (Zhou et al., 2020)

$$x_j^I = \frac{j-1}{p-1} x_{j-1} + \frac{p-j}{p-1} x_j, \qquad j = 1, \ldots, p. \tag{13}$$

Applying Eq. (13) to both blocks gives $\mathrm{INTERP}_p(\widehat{\theta}_{p-1})$. Identity padding is

$$\mathrm{ZPAD}_p(\widehat{\theta}_{p-1}) = (\beta_1, \ldots, \beta_{p-1}, 0, \gamma_1, \ldots, \gamma_{p-1}, 0). \tag{14}$$

The complete pool is therefore

$$\mathcal{A}_p = [\theta_p^T, \theta_p^D, \mathrm{INTERP}_p(\widehat{\theta}_{p-1}), \mathrm{ZPAD}_p(\widehat{\theta}_{p-1})], \quad p > 1. \tag{15}$$

Its size is $K_1 = 2$ and $K_p = 4$ thereafter, independent of the dimension $2p$. No parameter grid or landscape reconstruction is performed.

4.4 FRESH TARGET-SIDE EMPIRICAL ANCHOR GUARD

Divide S_p^R evenly among the K_p candidates, assigning integer remainders in $\mathcal{A}_p$ order, and collect n_k fresh measurements on G for candidate k. Let a index the target-search anchor θ_p^T. If any $n_k < 2$, return the anchor. Otherwise, compute sample means $\widehat{\mu}_k$ and unbiased sample variances s_k^2, and let $j = \arg\max_{k \neq a} \widehat{\mu}_k$. Replace the anchor by j only if

$$\widehat{\mu}_j - \widehat{\mu}_a > \lambda \sqrt{\frac{s_a^2}{n_a} + \frac{s_j^2}{n_j}}. \tag{16}$$

The margin coefficient $\lambda > 0$ is not a calibrated confidence level; we use $\lambda = 1/4$. Challenger selection and variance estimation use the same samples, without within-race multiplicity correction. A fixed-pool comparison with direct fresh-sample argmax detects no additional quality gain from this guard (Section F.1).

4.5 Algorithm summary

Algorithm 1 (Appendix A) summarizes one-time graph reduction, branch-specific continuation, constant-size candidate construction, and fresh target-side selection, with fixed control parameters and deterministic ordering and tie-breaking. The appendix also proves raw-shot conservation and ideal-circuit identity embedding, bounds finite-shot selection loss relative to the target-search anchor, and analyzes implementation-specific resource costs.

5 Experiments

5.1 Experimental setting

We evaluate on connected graphs ($3 \leq n \leq 10$) from AIDS, Linux, and IMDb in the Red-QAOA repository. Disjoint SHA-256-ordered splits contain 4 development and 30 holdout graphs per dataset, with three seeds per graph. BOLT-DA uses an $80/15/5$ target/donor/selection shot split and $\rho = \lambda = 1/4$. Comparators are RED-QAOA (Wang et al., 2024), Neural QAOA^2 (Zheng et al., 2026) rerun with its native components, and MG-control (Qian et al., 2024). All methods match online raw-shot budgets, $S_p = 256pq$, per depth and cumulatively, but not circuit-batch, gate-shot, or classical costs. Optimization uses sampled cuts; exact expectations are evaluated only after outputs are frozen. AR and squared-gap losses are averaged over seeds within graphs, with pointwise graph-paired bootstrap intervals. Implementation details are in Appendix B.

5.2 Low-shot solution quality

Table 1 reports the small-graph expected-value results for $q = 32$ and $p = 1, \ldots, 5$. Among the four displayed methods, BOLT-DA has the highest mean AR and lowest normalized-gap MSE in all 15 dataset–depth settings; Neural QAOA^2 ranks second throughout. For BOLT-DA, dataset-averaged AR increases and MSE decreases with depth, whereas greater depth does not consistently improve Red-QAOA or MG-control. Strong internal alternatives are examined in Section 5.4.

Table 1: Mean expected-value results at $q = 32$. MG-control is the MG-inspired multi-angle comparison described in Appendix B. Bold marks the best displayed mean, not statistical significance.

Dataset	p	BOLT-DA		Red-QAOA		Neural QAOA^2		MG-control	
		AR↑	MSE↓	AR↑	MSE↓	AR↑	MSE↓	AR↑	MSE↓
AIDS	1	**0.772645**	**0.053747**	0.755273	0.062961	0.771772	0.053957	0.727218	0.081183
	2	**0.823020**	**0.033421**	0.767251	0.061418	0.814210	0.037512	0.770064	0.057101
	3	**0.841664**	**0.027122**	0.766044	0.061402	0.825710	0.033020	0.718153	0.084527
	4	**0.856464**	**0.022725**	0.740348	0.085025	0.833039	0.031337	0.746170	0.071010
	5	**0.861631**	**0.021499**	0.731929	0.086012	0.828221	0.033021	0.691195	0.099613
Linux	1	**0.740726**	**0.067574**	0.725068	0.076625	0.739689	0.068053	0.713063	0.084981
	2	**0.788032**	**0.046273**	0.731913	0.078247	0.783536	0.048359	0.752139	0.062818
	3	**0.812859**	**0.036594**	0.710598	0.092321	0.800260	0.041721	0.709070	0.087414
	4	**0.825542**	**0.032239**	0.687172	0.110807	0.807193	0.039624	0.730284	0.076949
	5	**0.835532**	**0.027986**	0.688940	0.108722	0.805292	0.041354	0.680503	0.105476
IMDb	1	**0.959842**	**0.002558**	0.821624	0.052528	0.958006	0.002649	0.893779	0.012761
	2	**0.967448**	**0.001794**	0.781250	0.077103	0.964639	0.002069	0.922421	0.007508
	3	**0.971791**	**0.001421**	0.712304	0.117700	0.954775	0.003123	0.911742	0.009168
	4	**0.972672**	**0.001342**	0.694031	0.126371	0.939079	0.005045	0.905427	0.009981
	5	**0.974011**	**0.001205**	0.630480	0.172191	0.931135	0.006343	0.903880	0.010384

At $p = 5$, BOLT-DA reduces MSE by 75.00%, 74.26%, and 99.30% versus RED-QAOA, and by 34.89%, 32.33%, and 81.00% versus Neural QAOA^2, on AIDS, Linux, and IMDb, respectively. AR gains over Neural QAOA^2 are 3.34, 3.02, and 4.29 percentage points; the two metrics describe the

same solutions, not independent evidence. Through $p = 5$, BOLT-DA uses 5,188 circuit batches versus 3,840 for RED-QAOA (1.3510×); the depth-five-only ratio is $1{,}724/1{,}280 = 1.346875$. Logical one- and two-qubit gate-shot ratios to RED-QAOA are 1.159–1.763 and 1.311–2.054, respectively. These quality–cost comparisons do not establish hardware speedup; see Appendix E.

5.3 PERFORMANCE ACROSS NOMINAL-SHOT SETTINGS

We report the $p = 5$ results for the nominal-shot configurations $q \in \{32, 128, 512\}$, keeping the small-graph holdout and optimizer seeds fixed. Each of the 36 method–dataset–configuration entries averages 30 graphs and three seeds per graph. For all four methods, q indexes the same online raw measurement-shot budget, rather than the literal shot count of every circuit call. The common incremental and cumulative caps are $S_p = 256pq$ and $S_{\le p} = 128q\,p(p+1)$. Thus, at $q = 32$ every method receives 40,960 shots at depth five and 122,880 cumulatively through depth five; at $q = 512$, the corresponding common caps are 655,360 and 1,966,080. Across the displayed configurations, BOLT-DA attains the lowest mean MSE and highest mean AR on all three datasets (Figure 2). Comparing its $q = 32$ results with the best comparison-method means displayed in the $q = 512$ column, MSE is lower by 17.48%, 12.63%, and 65.26% on AIDS, Linux, and IMDb, respectively; mean AR is higher by 1.34, 0.81, and 2.58 percentage points. These are observed quality differences between the reported experimental configurations. Appendix C gives the complete numerical values.

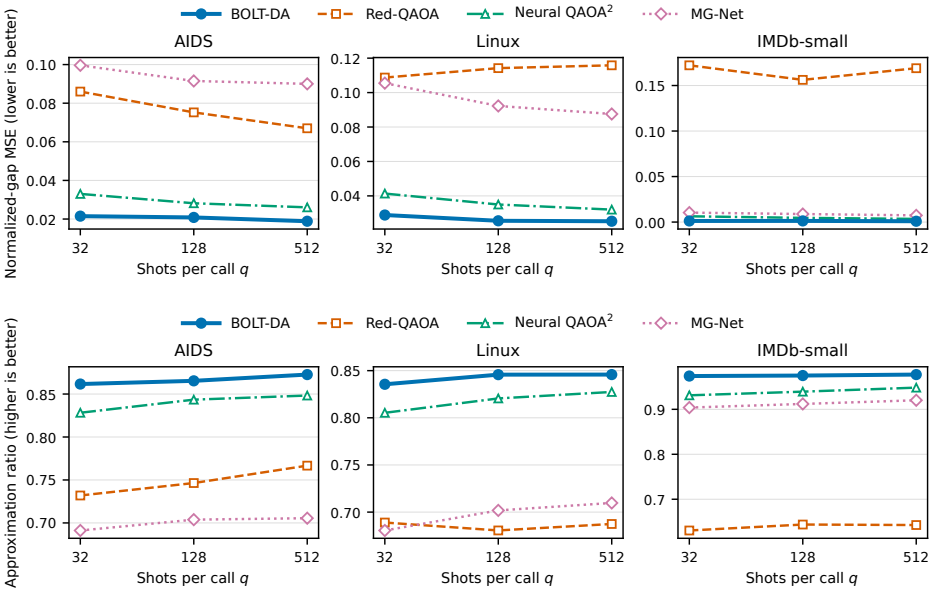

Figure 2: Performance at $p = 5$ across the reported nominal-shot settings: normalized-gap MSE (top) and mean AR (bottom).

5.4 COMPONENT ABLATIONS

At $q = 32$, $p = 5$, all variants use the same 30 holdout graphs per dataset and three shared seeds, with matched depth-ladder budgets: $S_5 = 40{,}960$ incremental and $S_{\le 5} = 122{,}880$ cumulative shots per graph–seed run. Full (BOLT-DA) uses an $80/15/5$ target/donor/selection split. Target spends all shots on cold target search; T+I adds branch-wise INTERP warm starts. NoD uses $95/0/5$, retaining reuse candidates and fresh selection; 2ndT replaces the donor with an independent target branch at Full's split. −ReuseCand. removes both reuse candidates, while INTERP and ZPAD retain only the named candidate; all three retain branch-wise warm starts.

Table 2: Equal-raw-shot ablations at $q = 32$, $p = 5$: mean normalized-gap MSE (↓).

Variant	AIDS	Linux	IMDb
Full	**0.021499**	**0.027986**	**0.001205**
Target	0.042390	0.050701	0.005921
T+I	0.037286	0.041291	0.006687
NoD	0.026629	0.033650	0.001271
2ndT	0.025992	0.029869	0.001236
−ReuseCand.	0.029898	0.036950	0.004576
INTERP	0.027445	0.032861	0.004380
ZPAD	0.025580	0.031198	0.001870

Full achieves the lowest pooled MSE among the displayed variants on all three datasets (Table 2). Removing both reuse candidates substantially increases MSE, while ZPAD consistently yields lower MSE than INTERP, supporting identity padding as a major source of reuse gains, especially on IMDb. Full further reduces MSE relative to ZPAD by 16.0%, 10.3%, and 35.6% on AIDS, Linux, and IMDb, respectively.These point estimates do not establish statistical superiority. Table 11 provides separate graph-paired AR intervals, not MSE significance tests. At fixed selection budget, changing the pool size also changes the fresh shots allocated per candidate.

Donor and second-trajectory effects. Full lowers pooled MSE relative to NoD by 19.3%, 16.8%, and 5.2% on AIDS, Linux, and IMDb, respectively, with larger donor-associated gains on AIDS and Linux. Relative to 2ndT, the reductions are 17.3%, 6.3%, and 2.5%. These point estimates do not establish statistical superiority; Table 10 reports separate graph-paired AR intervals. 2ndT lowers pooled MSE relative to NoD by 2.4%, 11.2%, and 2.8%, suggesting benefits from a second target-search trajectory, especially on Linux. At equal raw-shot budgets and circuit-batch counts, Full uses fewer logical gate-shots than 2ndT (Table 13), offering a quality–logical-cost trade-off through reduced-graph search.

Selection and reuse diagnostics. Fixed-pool comparison with fresh-sample argmax detects no additional guard benefit, with only 3.3%–5.6% disagreement (Table 14). The target anchor is selected most often on AIDS and Linux, while ZPAD accounts for 66.9% of selections on IMDb. At $p = 5$, selection is exact-best within the pool in 88.9%–97.8% of runs, with mean AR regret below 0.06 percentage points. Regressions occur in 5.8%–6.7% of adjacent-depth transitions, so finite-shot monotonicity is not implied. All diagnostics are post-hoc and do not inform training (Appendix F).

5.5 Evaluation on Medium-Sized Graphs

For each $n \in \{12, 14, 16\}$, we evaluate ten connected 3-regular graphs and ten connected Erdős–Rényi $G(n, m)$ graphs with $m = 3n/2$. The two families have the same node count, edge count, and mean degree at each size, but different local topology. Each size–depth–method cell contains 20 graphs and three seeds, evaluated at the $q = 32$ configuration and $p = 1, \ldots, 5$.

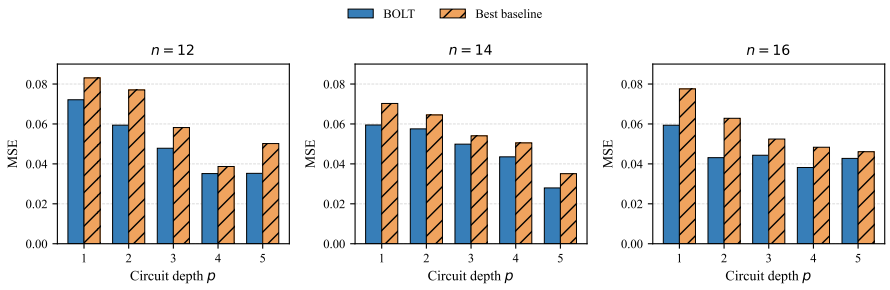

Figure 3: Native-solution MSE on medium-sized graphs at $q = 32$.

This extension uses one final cut solution per run as the common output unit, rather than the exact expected cut used in the small-graph panel. The resulting solution-level AR and squared-gap loss are reported separately in Appendix G.

Across all 15 size–depth settings, BOLT-DA has the highest mean native-solution AR and lowest native-solution MSE among the displayed methods (Table 17). Figure 3 shows the MSE comparison with the best reported comparator at each setting. The smallest AR lead is 0.96 percentage points and the smallest MSE reduction is 7.24%, both at $n = 16, p = 5$. These observations extend the quality comparison to medium-sized graphs and two topological families. They are comparisons of displayed means rather than simultaneous significance claims or a guarantee of monotone depth-wise improvement.

5.6 Hardware case study

We compare BOLT-DA, Red-QAOA, Neural QAOA^2, and MG-control on the AGate-100 processor using the five-node, four-edge Linux instance at $p \in \{2, 3, 4\}$ and three shared seeds. For BOLT-DA, each circuit readout uses 8,192 shots, and hardware measurements evaluate the optimized solution rather than update its search trajectory. Table 3 reports mean AR and MSE computed from the raw measurement counts.

Table 3: Hardware solution quality on the five-node, four-edge Linux instance. Entries are means over three seeds; bold marks the best displayed mean for each metric and depth.

p	BOLT-DA		Red-QAOA		Neural QAOA^2		MG-control	
	AR↑	MSE↓	AR↑	MSE↓	AR↑	MSE↓	AR↑	MSE↓
2	**0.748769**	**0.063664**	0.690450	0.099327	0.711446	0.085294	0.709564	0.087682
3	**0.703776**	**0.091927**	0.591014	0.172728	0.687195	0.101003	0.639455	0.132952
4	**0.676961**	**0.107371**	0.585968	0.180962	0.598022	0.170631	0.605011	0.157208

BOLT-DA achieves the highest mean AR and lowest MSE at every tested depth. Relative to the best comparison-method mean at each depth, MSE is lower by 25.36%, 8.99%, and 31.70% at $p = 2, 3, 4$, respectively. At $p = 4$, the relative AR gain is 11.89%. These results concern the tested instance and device executions. A bind-first compilation audit of BOLT-DA, Red-QAOA, and MG-control shows comparable CZ counts at $p = 2$ and lower mean BOLT-DA circuit cost at $p = 3, 4$. Relative to these two resource comparators, compiled depth is lower by 8.75%–10.43% at $p = 3$ and 14.49%–14.90% at $p = 4$; CZ count is lower by 11.11% and 16.67%, respectively. Retaining a shallower solution can therefore translate into lower compiled cost in this case. Appendix H gives the compilation protocol and per-method resource table. A single instance does not establish a universal hardware advantage.

6 Conclusion

We presented BOLT-DA, a finite-shot MaxCut QAOA optimizer that maintains branch-specific search trajectories while retaining structured candidates from previously selected solutions. Target search, donor search, and fresh candidate comparison share an explicit measurement allocation. The small-graph experiments improve mean AR and normalized-gap MSE over the reported comparison methods, and the native-solution extension retains leading means on 12-, 14-, and 16-node graphs. Ablations and post-hoc diagnostics identify identity padding as the principal reuse mechanism. The reduced donor offers a quality–logical-cost trade-off, while the fixed-pool comparison detects no additional quality gain from the empirical guard. The hardware case study demonstrates solution execution and the compiled-cost benefit of retaining shallower candidates on one instance. Together, these results support budgeted solution retention and fresh comparison during depth growth. Evaluation beyond the tested graph sizes and closed-loop hardware optimization of BOLT-DA remain directions for further work; the present results do not establish end-to-end hardware acceleration or quantum advantage.

REPRODUCIBILITY STATEMENT

The accompanying artifact records source code, environment information, protocol manifests, graph and result hashes, retained run-level and aggregated data, hardware counts, and reporting commands. Graph selection, random seeds, and BOLT-DA control settings are fixed before aggregation. Exact reporting and post-hoc diagnostics are separated from the sampled BOLT-DA training oracle. All four reported methods consume identical online raw measurement-shot budgets at each depth and cumulatively through the depth ladder. Red-QAOA results use the complete vendored workflow, including original-graph refinement after reduced-graph transfer. Neural QAOA^2 results come from newly executed full-method runs. The supplementary reporting script records input SHA-256 hashes and retains row-level candidate frequencies, streaks, regressions, oracle regrets, guard contrasts, direct Red-QAOA costs, and scale results in `results/supplementary_evidence_v1/`. Reporting scripts do not delete or overwrite the raw experimental records.

ETHICS STATEMENT

This study uses graph benchmarks and quantum-computing infrastructure; it involves no human participants, personal data, or consequential automated decisions. We report shot, batch, gate, simulator, and hardware scope so that performance claims are not detached from their computational cost. Hardware experiments consume shared energy and device time, motivating the finite-shot focus and the release of raw counts to avoid unnecessary repetition.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, wording refinement and consistency checks. The research design, algorithm development, experiments, analysis, and conclusions were carried out and verified by the authors. All AI-assisted edits were reviewed by the authors, who remain fully responsible for the manuscript.

REFERENCES

V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte. Reachability deficits in quantum approximate optimization. *Physical Review Letters*, 124:090504, 2020. doi: 10.1103/PhysRevLett.124.090504.

V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte. Parameter concentrations in quantum approximate optimization. *Physical Review A*, 104:L010401, 2021. doi: 10.1103/PhysRevA.104.L010401.

Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh. Accelerating quantum approximate optimization algorithm using machine learning. In *Design, Automation and Test in Europe Conference and Exhibition*, pp. 686–689, 2020. doi: 10.23919/DATE48585.2020.9116348.

Ohad Amosy, Tamuz Danzig, Ohad Lev, Ely Porat, Gal Chechik, and Adi Makmal. Iteration-free quantum approximate optimization algorithm using neural networks. *Quantum Machine Intelligence*, 6:38, 2024. doi: 10.1007/s42484-024-00159-y.

Andrew Arrasmith, Zoë Holmes, M. Cerezo, and Patrick J. Coles. Effect of barren plateaus on gradient-free optimization. *Quantum*, 5:558, 2021. doi: 10.22331/q-2021-10-05-558.

Bao Bach, Jose Falla, and Ilya Safro. MLQAOA: Graph learning accelerated hybrid quantum-classical multilevel QAOA. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2024. doi: 10.1109/QCE60285.2024.00072. URL https://doi.org/10.1109/QCE60285.2024.00072.

Eunok Bae and Soojoon Lee. Recursive QAOA outperforms the original QAOA for the MAX-CUT problem on complete graphs. *Quantum Information Processing*, 23(3):78, 2024. doi: 10.1007/s11128-024-04286-0. URL https://link.springer.com/article/10.1007/s11128-024-04286-0.

Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou. The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington–Kirkpatrick model. In *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 7:1–7:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.TQC.2022.7.

Marvin Bechtold, Johanna Barzen, Frank Leymann, Alexander Mandl, Julian Obst, Felix Truger, and Benjamin Weder. Investigating the effect of circuit cutting in QAOA for the MaxCut problem on NISQ devices. *Quantum Science and Technology*, 8(4):045022, 2023. doi: 10.1088/2058-9565/acf59c. URL https://doi.org/10.1088/2058-9565/acf59c.

Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters*, 125:260505, 2020. doi: 10.1103/PhysRevLett.125.260505.

E. Campos, D. Rabinovich, V. Akshay, and J. Biamonte. Training saturation in layerwise quantum approximate optimization. *Physical Review A*, 104:L030401, 2021. doi: 10.1103/PhysRevA.104.L030401.

M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3:625–644, 2021a. doi: 10.1038/s42254-021-00348-9.

M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12:1791, 2021b. doi: 10.1038/s41467-021-21728-w.

Daniel J. Egger, Jakub Marecek, and Stefan Woerner. Warm-starting quantum optimization. *Quantum*, 5:479, 2021. doi: 10.22331/q-2021-06-17-479.

Jose Falla, Quinn Langfitt, Yuri Alexeev, and Ilya Safro. Graph representation learning for parameter transferability in quantum approximate optimization algorithm. *Quantum Machine Intelligence*, 6:46, 2024. doi: 10.1007/s42484-024-00178-9.

Han Fang, Paul Weng, and Yutong Ban. ASAP: Exploiting the satisficing generalization edge in neural combinatorial optimization. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/Downloads/2026.

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

Alexey Galda, Xiaoyuan Liu, Danylo Lykov, Yuri Alexeev, and Ilya Safro. Transferability of optimal QAOA parameters between random graphs. In *2021 IEEE International Conference on Quantum Computing and Engineering*, pp. 171–180, 2021. doi: 10.1109/QCE52317.2021.00034.

Alexey Galda, Eesh Gupta, Jose Falla, Xiaoyuan Liu, Danylo Lykov, Yuri Alexeev, and Ilya Safro. Similarity-based parameter transferability in the quantum approximate optimization algorithm. *Frontiers in Quantum Science and Technology*, 2, 2023. doi: 10.3389/frqst.2023.1200975.

Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. doi: 10.1145/227683.227684.

Gian Giacomo Guerreschi and Anne Y. Matsuura. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Scientific Reports*, 9:6903, 2019. doi: 10.1038/s41598-019-43176-9.

Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics*, 17:332–336, 2021. doi: 10.1038/s41567-020-01105-y.

Rebekah Herrman, Phillip C. Lotshaw, James Ostrowski, Travis S. Humble, and George Siopsis. Multi-angle quantum approximate optimization algorithm. *Scientific Reports*, 12:6781, 2022. doi: 10.1038/s41598-022-10555-8.

Yiming Huang, Yajie Hao, Jing Zhou, Xiao Yuan, Xiaoting Wang, and Yuxuan Du. PALQO: Physics-informed model for accelerating large-scale quantum optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1968. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/54f82cdae821aad5c2888d61a6515170-Paper-Conference.pdf.

Nishant Jain, Brian Coyle, Elham Kashefi, and Niraj Kumar. Graph neural network initialisation of quantum approximate optimisation. *Quantum*, 6:861, 2022. doi: 10.22331/q-2022-11-17-861.

Jonas M. Kuebler, Andrew Arrasmith, Lukasz Cincio, and Patrick J. Coles. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum*, 4:263, 2020. doi: 10.22331/q-2020-05-11-263.

Xinwei Lee, Xinjian Yan, Ningyi Xie, Dongsheng Cai, Yoshiyuki Saito, and Nobuyoshi Asai. Iterative layerwise training for the quantum approximate optimization algorithm. *Physical Review A*, 109:052406, 2024. doi: 10.1103/PhysRevA.109.052406.

Junde Li, Mahabubul Alam, and Swaroop Ghosh. Large-scale quantum approximate optimization via divide-and-conquer. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(6):1852–1860, 2023. doi: 10.1109/TCAD.2022.3212196.

Zhiding Liang et al. Graph learning for parameter prediction of quantum approximate optimization algorithm. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024. doi: 10.1145/3649329.3663523.

Phillip C. Lotshaw, Travis S. Humble, Rebekah Herrman, James Ostrowski, and George Siopsis. Empirical performance bounds for quantum approximate optimization. *Quantum Information Processing*, 20:403, 2021. doi: 10.1007/s11128-021-03342-3.

Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2:5, 2014. doi: 10.3389/fphy.2014.00005.

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

Charles Moussa, Hao Wang, Thomas Back, and Vedran Dunjko. Unsupervised strategies for identifying optimal parameters in quantum approximate optimization algorithm. *EPJ Quantum Technology*, 9:11, 2022. doi: 10.1140/epjqt/s40507-022-00131-4.

Guido Pagano, Aniruddha Bapat, Patrick Becker, Kelsey S. Collins, Arinjoy De, Paul W. Hess, Harvey B. Kaplan, Antonis Kyprianidis, Wen Lin Tan, Christopher Baldwin, Lucas T. Brady, Amey Deshpande, Fangli Liu, Alexey V. Gorshkov, and Christopher Monroe. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proceedings of the National Academy of Sciences*, 117(41):25396–25401, 2020. doi: 10.1073/pnas.2006373117.

Samuele Pedrielli, Christopher J. Anders, Lena Funcke, Karl Jansen, Kim A. Nicoli, and Shinichi Nakajima. Bayesian parameter shift rules in variational quantum eigensolvers. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10008504.

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*, volume 38, 2025. doi: 10.52202/085713-1576. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/438d8b727410a558c4266d1e1a4f8616-Abstract-Conference.html.

Moises Ponce, Rebekah Herrman, Phillip C. Lotshaw, Sarah Powers, George Siopsis, Travis Humble, and James Ostrowski. Graph decomposition techniques for solving combinatorial optimization problems with variational quantum algorithms. *Quantum Information Processing*, 24(2):60, 2025. doi: 10.1007/s11128-025-04675-z.

John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. doi: 10.22331/q-2018-08-06-79.

Yang Qian, Xinbiao Wang, Yuxuan Du, Yong Luo, and Dacheng Tao. MG-Net: Learn to customize QAOA with circuit depth awareness. In *Advances in Neural Information Processing Systems*, volume 37, pp. 33691–33725, 2024. doi: 10.52202/079017-1061.

Stefan H. Sack and Maksym Serbyn. Quantum annealing initialization of the quantum approximate optimization algorithm. *Quantum*, 5:491, 2021. doi: 10.22331/q-2021-07-01-491.

Stefan H. Sack, Raimel A. Medina, Richard Kueng, and Maksym Serbyn. Recursive greedy initialization of the quantum approximate optimization algorithm with guaranteed improvement. *Physical Review A*, 107:062404, 2023. doi: 10.1103/PhysRevA.107.062404.

Giuseppe Scriva, Nikita Astrakhantsev, Sebastiano Pilati, and Guglielmo Mazzola. Challenges of variational quantum optimization with measurement shot noise. *Physical Review A*, 109:032408, 2024. doi: 10.1103/PhysRevA.109.032408.

Ruslan Shaydulin and Stefan M. Wild. Exploiting symmetry reduces the cost of training QAOA. *IEEE Transactions on Quantum Engineering*, 2, 2021. doi: 10.1109/TQE.2021.3140190.

Ruslan Shaydulin, Ilya Safro, and Jeffrey Larson. Multistart methods for quantum approximate optimization. In *2019 IEEE High Performance Extreme Computing Conference*, 2019. doi: 10.1109/HPEC.2019.8916288.

Ruslan Shaydulin, Phillip C. Lotshaw, Jeffrey Larson, James Ostrowski, and Travis S. Humble. Parameter transfer for quantum approximate optimization of weighted MaxCut. *ACM Transactions on Quantum Computing*, 2023. doi: 10.1145/3584706.

Meng Wang, Bo Fang, Ang Li, and Prashant J. Nair. Red-QAOA: Efficient variational optimization through circuit reduction. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, pp. 980–998. Association for Computing Machinery, 2024. doi: 10.1145/3620665.3640363.

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*, 12:6961, 2021. doi: 10.1038/s41467-021-27045-6.

Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A*, 97:022304, 2018. doi: 10.1103/PhysRevA.97.022304.

Madita Willsch, Dennis Willsch, Fengping Jin, Hans De Raedt, and Kristel Michielsen. Benchmarking the quantum approximate optimization algorithm. *Quantum Information Processing*, 19:197, 2020. doi: 10.1007/s11128-020-02692-8.

Jonathan Wurtz and Danylo Lykov. Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs. *Physical Review A*, 104:052419, 2021. doi: 10.1103/PhysRevA.104.052419.

Yunlong Yu, Chenfeng Cao, Carter Dewey, Xiang-Bin Wang, Nic Shannon, and Robert Joynt. Quantum approximate optimization algorithm with adaptive bias fields. *Physical Review Research*, 4:023249, 2022. doi: 10.1103/PhysRevResearch.4.023249.

Zubin Zheng, Jiahao Wu, and Shengcai Liu. Neural QAOA^2: Differentiable joint graph partitioning and parameter initialization for quantum combinatorial optimization. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://icml.cc/Downloads/2026`.

Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10(2):021067, 2020. doi: 10.1103/PhysRevX.10.021067.

Zeqiao Zhou, Yuxuan Du, Xinmei Tian, and Dacheng Tao. QAOA-in-QAOA: Solving large-scale MaxCut problems on small quantum machines. *Physical Review Applied*, 19:024027, 2023. doi: 10.1103/PhysRevApplied.19.024027.

A METHOD ANALYSIS

This appendix gives the full algorithm, supporting properties, and implementation-specific resource analyses. It does not introduce additional algorithmic components.

A.1 FINITE-SHOT DEPTH-LADDER PSEUDOCODE

Algorithm 1 Finite-shot BOLT-DA depth ladder

Require: G, depth P, calls factor c, nominal shots q, seed s; $\alpha_T, \alpha_D, \alpha_R, \rho, \tau, \lambda$
Ensure: $\{\widehat{\theta}_p\}_{p=1}^{P}$

$\widetilde{G} \leftarrow \mathcal{R}_\tau(G; s)$
for $p = 1, \dots, P$ **do**
  $B_p \leftarrow cp$; $S_p \leftarrow B_p q$; $S_p^T \leftarrow \lfloor \alpha_T S_p \rfloor$; $S_p^D \leftarrow \lfloor \alpha_D S_p \rfloor$; $S_p^R \leftarrow S_p - S_p^T - S_p^D$
  if $p > 1$ **then**
    Warm-start target/donor from $\mathrm{INTERP}_p(\theta_{p-1}^T)$ / $\mathrm{INTERP}_p(\theta_{p-1}^D)$
  end if
  $\theta_p^T \leftarrow \mathrm{ShotDE}(G, p, S_p^T, B_p)$; $\theta_p^D \leftarrow \mathrm{ShotDE}(\widetilde{G}, p, S_p^D, \lfloor \rho B_p \rfloor)$
  $\mathcal{A}_p \leftarrow [\theta_p^T, \theta_p^D]$
  if $p > 1$ **then**
    $\mathcal{A}_p \leftarrow \mathrm{append}\Big(\mathcal{A}_p, [\mathrm{INTERP}_p(\widehat{\theta}_{p-1}), \mathrm{ZPAD}_p(\widehat{\theta}_{p-1})]\Big)$
  end if
  $K_p \leftarrow |\mathcal{A}_p|$; distribute S_p^R shots as evenly as possible to obtain $\{n_k\}_{k=1}^{K_p}$
  Measure each $\mathcal{A}_p[k]$ on G with n_k fresh shots
  $a \leftarrow 1$; $\widehat{\theta}_p \leftarrow \mathcal{A}_p[a]$
  if $n_k \geq 2,\ \forall k$ **then**
    Compute $\{\widehat{\mu}_k, s_k^2\}_{k=1}^{K_p}$; $j \leftarrow \arg\max_{k \neq a} \widehat{\mu}_k$ using fixed tie-breaking
    if Eq. (16) holds strictly **then**
      $\widehat{\theta}_p \leftarrow \mathcal{A}_p[j]$
    end if
  end if
end for
return $\{\widehat{\theta}_p\}_{p=1}^{P}$

A.2 RAW-SHOT ACCOUNTING AND IDENTITY EMBEDDING

Proposition 1 (Raw-shot conservation)**.** *At depth p,* BOLT-DA *consumes exactly cpq training samples. Through depth P, it consumes exactly* $cqP(P+1)/2$ *samples.*

Proof. The target search, donor search, and fresh selection stages exhaust their assigned budgets. Hence

$$S_p^T + S_p^D + S_p^R = S_p = cpq.$$

Summing over $p = 1, \dots, P$ yields

$$\sum_{p=1}^{P} S_p = cq \sum_{p=1}^{P} p = cqP(P+1)/2.$$

□

This equality concerns raw training samples only. It does not imply equal circuit submissions, logical gate-shots, routed gates, noise exposure, or wall-clock time.

Proposition 2 (Identity embedding)**.** *For* $p > 1$*,*

$$J_G\Big(\mathrm{ZPAD}_p(\widehat{\theta}_{p-1})\Big) = J_G(\widehat{\theta}_{p-1}). \tag{17}$$

Proof. The added mixing and cost angles are zero. The new layer is therefore $U_B(0)U_C^G(0) = I$, leaving both the state and its expected cut unchanged. □

Equation (17) guarantees that an equal-value candidate remains available in the ideal circuit model. It does not imply finite-shot monotonicity, because noisy selection can still return a candidate with lower true expectation.

A.3 FINITE-SHOT SELECTION BOUND

Condition on the candidate pool before fresh selection. Let μ_k denote the true expected cut value of candidate k, let a index the target-search anchor, and let μ_{out} denote the true expected cut value of the returned candidate. If any candidate receives fewer than two samples, the selection rule returns the anchor and $\mu_{\text{out}} = \mu_a$.

Otherwise, for $\delta \in (0, 1)$ define

$$\epsilon_k = m\sqrt{\frac{\log(2K_p/\delta)}{2n_k}}. \tag{18}$$

Because each fresh cut sample lies in $[0, m]$, Hoeffding's inequality and a union bound imply

$$|\widehat{\mu}_k - \mu_k| \le \epsilon_k$$

simultaneously for all candidates with probability at least $1 - \delta$.

On this event, retaining the anchor satisfies the following bound immediately. If a challenger j is accepted, then $\widehat{\mu}_j > \widehat{\mu}_a$, and therefore

$$\mu_j \ge \widehat{\mu}_j - \epsilon_j \ge \widehat{\mu}_a - \epsilon_j \ge \mu_a - \epsilon_a - \epsilon_j.$$

Hence any rule that either retains the anchor or switches only to a candidate with a larger empirical mean satisfies

$$\mu_{\text{out}} \ge \mu_a - \epsilon_a - \max_k \epsilon_k \tag{19}$$

with probability at least $1 - \delta$.

For $\lambda > 0$, the empirical guard in Eq. (16) satisfies the required switching condition. This generic bound does not calibrate λ as a confidence level, certify improvement over the anchor, or imply dominance over an independently optimized baseline.

A.4 SHOT, BATCH, AND LOGICAL GATE-SHOT COMPLEXITY

Let $\widetilde{n} = |V_{\widetilde{G}}|$ and $\widetilde{m} = |E_{\widetilde{G}}|$. Proposition 1 gives cumulative raw-shot complexity $\Theta(cqP^2)$.

ShotDE instance. The implementation analyzed here uses population size

$$r(b, p) = \max\{4, \min[\max(8, 4p), \max(4, \lfloor b/4 \rfloor)]\}. \tag{20}$$

The mutation and crossover factors are fixed at 0.72 and 0.88. Each trial–incumbent comparison remeasures both points with equal fresh shots. The final population is remeasured, and integer shot remainders are fully spent within the assigned branch budget.

Circuit batches. Let $C(X, b, p)$ be the actual number of circuit batches used by a ShotDE branch with X shots and b nominal calls. This count includes fresh trial–incumbent comparisons, final-population evaluation, and remainder redistribution. When each candidate receives at least one selection shot,

$$C_p = C(S_p^T, B_p, p) + C(S_p^D, \lfloor \rho B_p \rfloor, p) + K_p. \tag{21}$$

If some candidate receives zero selection shots, the final term is replaced by the number of positively sampled candidates.

For fixed q and fixed control parameters, the specified ShotDE implementation satisfies

$$C_p = O(cp + p\log(1 + cq)),$$

and therefore

$$C_{\le P} = O(cP^2 + P^2\log(1 + cq)).$$

The constant-size selection stage does not change this order.

Logical gate-shots. Using the logical depth-p circuit template in Eq. (2), including initial state preparation, each target-graph shot contributes $n(p+1)$ one-qubit gates and mp two-qubit gates. Donor shots use the corresponding reduced-graph sizes. Before zero-angle elimination or hardware compilation,

$$G_{1,p} = (S_p^T + S_p^R)n(p+1) + S_p^D\widetilde{n}(p+1), \tag{22}$$

and

$$G_{2,p} = (S_p^T + S_p^R)mp + S_p^D\widetilde{m}p. \tag{23}$$

Both accumulate as $O(cqP^3)$ times the corresponding weighted graph size.

When the donor graph is smaller than the target graph, the reduced branch can lower logical gate-shots relative to using the same shots exclusively on the target graph, while the two-branch optimizer can still use more circuit batches. These ledgers exclude routing, SWAP insertion, compilation, queueing, calibration drift, and error mitigation, and should not be interpreted as calibrated QPU time.

A.5 Classical simulator time and space

The statevector implementation computes an ideal output distribution and draws multinomial training samples from it. For a graph with r vertices and e edges, cut/phase preprocessing costs $O(e2^r)$ time and $O(2^r)$ memory. After preprocessing, one depth-p state evolution costs $O(pr2^r)$ time.

Up to logarithmic factors from remainder batches, state evolution through depth P requires

$$O\Big(cP^3[n2^n + \widetilde{n}2^{\widetilde{n}}]\Big). \tag{24}$$

This expression accounts for state evolution rather than total execution time; graph reduction, preprocessing, sample generation, and optimizer management are additional costs.

Mutation and crossover accumulate as $O(cP^3)$, while duplicate-checked population initialization admits the conservative bound $O(P^4)$. Peak classical space is

$$O\Big(2^n + 2^{\widetilde{n}} + n + m + \widetilde{n} + \widetilde{m} + P^2\Big). \tag{25}$$

The exponential terms arise from classical statevector simulation and do not describe QPU memory requirements.

B Experimental protocol and implementation details

B.1 Graph eligibility, partitions, and independent units

The small-graph corpus is the unmodified AIDS, Linux, and IMDb collection distributed with the Red-QAOA repository. A graph is eligible if and only if it is connected, has at least one edge, and satisfies $3 \le n \le 10$. For graph index i in a dataset, the deterministic ordering key is

$$\begin{aligned} h(G_i) = \mathrm{SHA256}\big(&\texttt{BOLT|small-ideal-five-method|v1} \\ &\|\texttt{dataset}\|i\big). \end{aligned} \tag{26}$$

Eligible graphs are sorted by this key within each dataset. Policy development uses ranks 1–4 (12 graphs in total) with seeds $\{0, 1, 2\}$; the disjoint confirmatory holdout uses ranks 113–142 (30 graphs per dataset) with seeds $\{5, 6, 7\}$. Each holdout cell contains 90 runs. For uncertainty analysis, seed-level ARs and seed-level squared gaps are averaged separately within each graph, then paired graphs are resampled for 20,000 bootstrap replicates. The graph, not an optimizer seed, is the independent uncertainty unit.

Table 4: Deterministic graph selection. Graph IDs are retained in the machine-readable manifest; ranks refer to the within-dataset SHA-256 ordering in Eq. (26).

Dataset	Raw records	Eligible	Development ranks	Holdout ranks
AIDS	700	698	1–4	113–142
Linux	1,000	1,000	1–4	113–142
IMDb	1,498	812	1–4	113–142

B.2 BOLT-DA OPTIMIZER AND CONTROL SETTINGS

At depth p, BOLT-DA uses the incremental allocation $S_p = cpq$ with $c = 256$. We test $q \in \{32, 128, 512\}$; the primary setting is $q = 32$ and $p = 1, \ldots, 5$. Because optimization proceeds sequentially through depths, its cumulative allocation is $S_{\le p} = cq\,p(p+1)/2$. In particular, $S_5 = 40{,}960$ and $S_{\le 5} = 122{,}880$ at $q = 32$. The internal equal-shot ablations use the same per-depth and cumulative allocations.

Table 5: BOLT-DA control settings used in the experiments.

Item	Setting
Shot split	Target/donor/selection $= 80/15/5$ percent
Nominal search calls	Target: $256p$; donor: $64p$; $\rho = 1/4$
Parameter dimension	$2p$ for shared-angle QAOA
DE population	$\max(8, 4p)$ for the reported branch call counts; see Eq. (20) for the general rule
Mutation / crossover	$F = 0.72$, $CR = 0.88$; one forced crossover coordinate
Angle domain	Every coordinate is reduced modulo π
DE comparisons	Trial and incumbent are evaluated with fresh samples
Final DE choice	Fresh evaluation of the final population
Candidate pool	Target anchor and reduced donor; INTERP and ZPAD from the previous selected solution are added for $p > 1$
Selection allocation	Uniform over candidates, with integer remainders assigned in candidate-list order
Guard	$\lambda = 1/4$ in Eq. (16)
Training observations	Multinomial cut samples drawn from the ideal output distribution
Expected-value reporting	Exact expected cut and the AR/MSE definitions in Eq. (10), evaluated after optimization

The initial population includes, when dimensionally valid, the primary initializer, previous-depth INTERP, a deterministic seed-specific point, the fixed angles $\beta = 3\pi/4$, $\gamma = \pi/6$, and random fill. Each trial–incumbent comparison remeasures both points. Final-population evaluations and integer-shot remainders are included in the branch budget. A counter-keyed multinomial random-number generator isolates measurement batches: changing a branch allocation does not advance the random stream of later batches. Exact expected cuts are computed for reporting after outputs and finite-shot choices have been finalized, not as optimizer observations.

The donor is constructed once per graph–seed run with the stochastic Red-QAOA reducer and reused throughout the depth ladder. The reduction criterion τ is the average-degree retention criterion defined in the method, not a vertex-retention fraction; the reducer configuration is held fixed across depths.

B.3 COMPARISON-METHOD IMPLEMENTATIONS

Table 6: Implemented comparison methods and internal controls.

Method	Implementation
Red-QAOA	The full vendored Red-QAOA workflow is executed. Its stochastic reducer constructs the reduced graph, reduced-graph optimization supplies transferred parameters, and the method then performs its target-side refinement on the original graph before the final original-graph output is reported. The Red-QAOA shot/batch audit charges the complete finite-shot workflow, including the target-side refinement.
Neural QAOA^2	The complete method is rerun for the reported experiments, including its native graph-partitioning and parameter-generation components. The reported results are obtained from these new runs.
MG-control	A node-specific multi-angle X-mixer capacity control is trained with the finite-shot protocol. It is inspired by MG-Net, rather than a reproduction of the official pretrained MG-Net model.
Target / T+I	One original-graph ShotDE branch uses the internal ablation budget. Target is cold-started; T+I adds branch-wise INTERP from the preceding depth.
NoD	A $95/0/5$ target/donor/selection split removes the donor while retaining INTERP/ZPAD candidates and fresh target-side selection.
2ndT	Full's $80/15/5$ split and selection rule are retained; the reduced donor is replaced by a separately initialized original-graph search.

For all four reported methods, the online raw measurement-shot budget is exactly matched at every depth: $S_p = 256pq$, with cumulative budget $S_{\le p} = 128q\,p(p+1)$. This equality applies only to raw online measurement shots and does not assert equal circuit-batch counts, logical gate-shots, or classical computational cost.

C COMPLETE RESULTS ACROSS NOMINAL-SHOT SETTINGS

The tables give complete $p = 5$ quality results for the configurations shown in Section 5.3. Each entry averages 30 held-out graphs and three optimizer seeds per graph. Every method receives the same cumulative online raw measurement-shot caps through $p = 5$: 122,880, 491,520, and 1,966,080 shots for $q = 32, 128, 512$, respectively.

Table 7: Mean approximation ratio (↑) across nominal-shot settings at $p = 5$. Bold and underlining indicate the best and second-best displayed means, respectively, not statistical significance.

Dataset	Method	$q = 32$	$q = 128$	$q = 512$
AIDS	BOLT-DA	**0.861631**	**0.865359**	**0.872645**
	RED-QAOA	0.731929	0.746426	0.766664
	Neural QAOA^2	<u>0.828221</u>	<u>0.843460</u>	<u>0.848256</u>
	MG-control	0.691195	0.703816	0.705506
Linux	BOLT-DA	**0.835532**	**0.845753**	**0.845829**
	RED-QAOA	0.688940	0.680557	0.687416
	Neural QAOA^2	<u>0.805292</u>	<u>0.820534</u>	<u>0.827405</u>
	MG-control	0.680503	0.701802	0.709696
IMDb	BOLT-DA	**0.974011**	**0.975121**	**0.977281**
	RED-QAOA	0.630480	0.643731	0.642625
	Neural QAOA^2	<u>0.931135</u>	<u>0.939297</u>	<u>0.948235</u>
	MG-control	0.903880	0.911984	0.920116

Table 8: Mean normalized-gap MSE (↓) across nominal-shot settings at $p = 5$, using the same graphs and seeds as Table 7.

Dataset	Method	$q = 32$	$q = 128$	$q = 512$
AIDS	BOLT-DA	**0.021499**	**0.020835**	**0.018863**
	RED-QAOA	0.086012	0.075236	0.067003
	Neural QAOA^2	0.033021	0.028173	0.026053
	MG-control	0.099613	0.091507	0.090005
Linux	BOLT-DA	**0.027986**	**0.025539**	**0.025278**
	RED-QAOA	0.108722	0.114205	0.115836
	Neural QAOA^2	0.041354	0.034996	0.032032
	MG-control	0.105476	0.092234	0.087564
IMDb	BOLT-DA	**0.001205**	**0.001277**	**0.000911**
	RED-QAOA	0.172191	0.156218	0.169011
	Neural QAOA^2	0.006343	0.004607	0.003469
	MG-control	0.010384	0.008820	0.007363

D ADDITIONAL COMPONENT-ABLATION RESULTS

This appendix complements Table 2 with AR means, graph-paired intervals for the main internal contrasts, and post-hoc reuse comparisons. Logical training costs are reported separately in Appendix E.

D.1 APPROXIMATION-RATIO MEANS

Table 9: Mean approximation ratio (↑) for the equal-raw-shot ablations at $q = 32$, $p = 5$. Each entry averages 30 graphs and three seeds per graph. Bold marks the highest displayed mean per dataset.

Variant	AIDS	Linux	IMDb
Full	0.861631	0.835532	0.974011
Target	0.802207	0.779284	0.931457
T+I	0.817400	0.805361	0.929369
NoD	0.857945	0.828851	0.972643
2ndT	**0.865838**	**0.840328**	**0.974393**
−ReuseCand.	0.837304	0.815644	0.939715
INTERP	0.841709	0.818727	0.941158
ZPAD	0.861536	0.834326	0.973055

D.2 MAIN GRAPH-PAIRED CONTRASTS

Table 10 provides the primary comparisons with Target, T+I, NoD, 2ndT, and the variant without reuse candidates. The AR column reports graph-paired bootstrap intervals after averaging seeds within each graph. The MSE column is the relative reduction computed directly from the pooled MSE means in Table 2, using Eq. (11). The AR intervals support Full over Target, T+I, and −ReuseCand. across all datasets. Full–NoD AR intervals include zero throughout; Full–2ndT AR intervals favor 2ndT on AIDS and Linux, while the IMDb interval includes zero. Separately, the pooled MSE point estimates favor Full in all displayed contrasts. In particular, the Full–NoD MSE reductions are 19.3%, 16.8%, and 5.2%, and the Full–2ndT reductions are 17.3%, 6.3%, and 2.5% on AIDS, Linux, and IMDb, respectively. A higher mean AR need not imply a lower mean squared gap, so the two metric orderings must be distinguished. The AR intervals do not establish statistical separation in MSE, and no MSE interval is reconstructed from the aggregate means.

Table 10: Main equal-raw-shot contrasts at $q = 32$, $p = 5$. Positive values favor Full. AR entries contain pointwise 95% graph-paired bootstrap intervals. MSE reductions are deterministic ratios of the pooled MSE means in Table 2; no bootstrap interval is attached to that aggregate ratio here.

Dataset	Comparator	Full − comparator ΔAR (pp) [95% CI]	Full MSE reduction (%)
AIDS	Target	5.94 [4.91, 7.06]	49.3
	T+I	4.42 [3.12, 5.72]	42.3
	NoD	0.37 [-0.57, 1.31]	19.3
	2ndT	-0.42 [-0.81, -0.08]	17.3
	−ReuseCand.	2.43 [1.85, 3.06]	28.1
Linux	Target	5.62 [4.52, 6.74]	44.8
	T+I	3.02 [2.14, 3.89]	32.2
	NoD	0.67 [-0.18, 1.56]	16.8
	2ndT	-0.48 [-0.88, -0.12]	6.3
	−ReuseCand.	1.99 [1.49, 2.53]	24.3
IMDb	Target	4.26 [3.78, 4.74]	79.6
	T+I	4.46 [3.70, 5.26]	82.0
	NoD	0.14 [-0.11, 0.38]	5.2
	2ndT	-0.04 [-0.18, 0.11]	2.5
	−ReuseCand.	3.43 [2.90, 4.00]	73.7

D.3 Post-hoc paired contrasts

For method a relative to reference b, we report AR differences in percentage points and relative pooled-MSE reductions in percent. Positive values favor the first method. The AR intervals use 20,000 paired bootstrap replicates over graph units after averaging seeds within each graph. The MSE reductions are computed directly from the pooled means in Table 2 using Eq. (11); this avoids mixing a graph-wise percentage statistic with the manuscript's aggregate MSE definition.

Table 11: Post-hoc contrasts at $q = 32$, $p = 5$. AR entries contain pointwise 95% graph-bootstrap intervals. MSE reductions are ratios of the pooled MSE means in Table 2.

Dataset	Comparison	ΔAR (pp) 95% CI	MSE reduction (%)
AIDS	Full − ZPAD	0.01 [-0.14, 0.22]	16.0
	ZPAD − INTERP	1.98 [1.50, 2.46]	6.8
	2ndT − NoD	0.79 [-0.03, 1.62]	2.4
Linux	Full − ZPAD	0.12 [-0.06, 0.40]	10.3
	ZPAD − INTERP	1.56 [0.98, 2.15]	5.1
	2ndT − NoD	1.15 [0.39, 1.97]	11.2
IMDb	Full − ZPAD	0.10 [0.02, 0.20]	35.6
	ZPAD − INTERP	3.19 [2.62, 3.78]	57.3
	2ndT − NoD	0.18 [-0.06, 0.41]	2.8

For Full–ZPAD, the paired AR intervals contain zero on AIDS and Linux and exclude zero on IMDb; the corresponding pooled-MSE reductions are 16.0%, 10.3%, and 35.6%. ZPAD lowers pooled MSE relative to INTERP by 6.8%, 5.1%, and 57.3%, respectively, supporting identity-preserving padding as the principal source of the observed reuse benefit in this setting. For 2ndT–NoD, the AR interval favors 2ndT only on Linux, while the AIDS and IMDb intervals include zero. The pooled MSE point estimates favor 2ndT on all three datasets, with reductions of 2.4%, 11.2%, and 2.8%, respectively. These MSE ratios describe the displayed point estimates; the AR intervals are not MSE significance tests.

The 2ndT–NoD contrast changes the number of independently initialized original-graph search branches while reserving the same 5% selection share. It therefore measures the value of a second search relative to concentrating the search budget in one branch. It does not change how an

individual branch chooses between its own previous output and the previously selected solution for warm-starting.

E EXECUTION-COST DEFINITIONS AND DIRECT AUDIT

A raw shot is one measurement sample. A circuit batch is one execution of a parameterized circuit with one or more shots. Logical one- and two-qubit gate-shots multiply each executed circuit's logical gate count by its shot count. Equal raw-shot totals therefore do not imply equal batches or circuit work. The audited comparisons in this appendix concern BOLT-DA, Red-QAOA, and the internal ablation variants.

E.1 DIRECT BOLT-DA–RED-QAOA SHOT AND BATCH LEDGER

Table 12: Per-run BOLT-DA–Red-QAOA shot and batch ledger at $q = 32$. Each row applies to one graph–seed depth ladder; the counts are fixed by the reported protocols. "At p" is incremental, while "through p" is cumulative over depths $1, \ldots, p$.

p	S_p	Full batches	Red batches	Ratio	$S_{\leq p}$	Full cumulative	Red cumulative / ratio
1	8,192	359	256	1.4023	8,192	359	256 / 1.4023
2	16,384	690	512	1.3477	24,576	1,049	768 / 1.3659
3	24,576	1,035	768	1.3477	49,152	2,084	1,536 / 1.3568
4	32,768	1,380	1,024	1.3477	81,920	3,464	2,560 / 1.3531
5	40,960	1,724	1,280	1.3469	122,880	5,188	3,840 / **1.3510**

The cumulative ratio through $p = 5$ is $5{,}188/3{,}840 = 1.3510417$, whereas the incremental ratio at depth five is $1{,}724/1{,}280 = 1.346875$. This distinction separates the cost of obtaining the entire depth ladder from the additional work at its final depth.

The direct comparison in Section 5.2 reports BOLT-DA/Red-QAOA logical gate-shot ratios of 1.159–1.763 for one-qubit gates and 1.311–2.054 for two-qubit gates. Thus, equal raw-shot budgets do not imply lower logical circuit work than Red-QAOA. These direct-method ratios use a different comparator from the Full–2ndT savings below.

E.2 INTERNAL LOGICAL TRAINING COSTS

Table 13: Mean cumulative training cost through $p = 5$ at $q = 32$. Every row uses $122{,}880$ raw shots. Gate-shot counts are in millions and represent logical circuit costs rather than measured wall-clock time.

Dataset	Method	Batches	1q gate-shots	2q gate-shots
AIDS	Full	5,188	4.64	3.54
	Target	3,840	4.93	3.81
	T+I	3,840	4.93	3.81
	NoD	4,086	4.93	3.81
	2ndT	5,188	4.93	3.81
Linux	Full	5,188	4.19	2.92
	Target	3,840	4.51	3.18
	T+I	3,840	4.51	3.18
	NoD	4,086	4.51	3.18
	2ndT	5,188	4.51	3.18
IMDb	Full	5,188	4.35	10.67
	Target	3,840	4.45	11.11
	T+I	3,840	4.45	11.11
	NoD	4,086	4.45	11.11
	2ndT	5,188	4.45	11.11

Full and 2ndT use the same raw-shot total and number of circuit batches, while Full uses fewer logical gate-shots because part of the search runs on the reduced graph. Ratios computed from the rounded means displayed in Table 13 give approximate Full–2ndT reductions of 2.25%–7.10% for one-qubit gate-shots and 3.96%–8.18% for two-qubit gate-shots. These percentages are approximate ratios of the printed gate-shot values, not additional precision in the underlying counts. Among NoD, Full, and 2ndT, NoD uses the fewest batches; Target and T+I use fewer batches still. Combined with the quality results, these observations describe alternative quality–logical-cost operating points, not measured end-to-end hardware acceleration.

F Post-hoc selection and reuse diagnostics

All diagnostics in this appendix are computed after the BOLT-DA search outputs, candidate pools, and finite-shot selection decisions have been finalized. They do not feed information back to optimization or candidate selection.

F.1 Empirical guard versus direct fresh-sample argmax

The no-guard comparison retains Full's search outputs, candidate pool, shot split, and fresh-sample streams, and selects the largest fresh sample mean directly. Table 14 compares the two rules on the same completed $p = 5$ records without rerunning optimization. It is a fixed-pool selector comparison, not an end-to-end rerun of the depth ladder with a different rule at every preceding depth. In Table 14, ΔAR is guard minus argmax in percentage points, whereas MSE reduction is $100(1 - \mathrm{MSE}_{\mathrm{guard}} / \mathrm{MSE}_{\mathrm{argmax}})$. Positive values favor the guard. As in Eq. (10), seed-level ARs and squared gaps are averaged separately within each graph before graph-level uncertainty is assessed.

Table 14: Full guard minus direct fresh-sample argmax at $p = 5$. The final column is the fraction of graph–seed runs on which the chosen candidate differs.

Dataset	ΔAR (pp) [95% CI]	MSE reduction (%) [95% CI]	Disagree
AIDS	-0.008 [-0.046, 0.021]	-0.23 [-0.94, 0.23]	4.4%
Linux	0.017 [-0.049, 0.085]	0.12 [-0.75, 0.96]	5.6%
IMDb	-0.005 [-0.085, 0.026]	-2.29 [-9.98, 1.35]	3.3%

All intervals cross zero, and the chosen candidate differs on only 3.3%–5.6% of runs. No additional quality benefit from the guard is detected for these fixed pools. The guard is retained as an empirical anchor-relative decision rule, not identified as a main performance source.

F.2 Candidate sources, retention streaks, and selection regret

Candidate-source frequencies identify which pool members are selected. A ZPAD streak is the maximum consecutive number of depths in $p = 2, \ldots, 5$ at which ZPAD is selected. A depth regression is a negative change in exact AR between adjacent completed depths. Oracle regret is the exact-AR difference between the best candidate in the same pool and the selected candidate. Exact candidate values used for this diagnostic are never exposed to training.

Table 15: Candidate source frequency over $p = 2, \ldots, 5$ (360 selections per dataset).

Dataset	Target	Donor	INTERP	ZPAD
AIDS	52.5%	6.7%	10.8%	30.0%
Linux	57.2%	6.9%	7.5%	28.3%
IMDb	29.4%	0.6%	3.1%	66.9%

Table 16: Post-freeze diagnostic summary. Oracle regret and maximum drops are reported in AR percentage points.

Dataset	Mean max ZPAD streak	Depth regressions	Max drop (AR pp)	$p = 5$ regret (AR pp) mean / 95th percentile	Oracle-best rate at $p = 5$
AIDS	1.07	21/360 (5.8%)	2.56	0.039 / 0.346	88.9%
Linux	1.00	24/360 (6.7%)	1.88	0.055 / 0.305	91.1%
IMDb	2.41	24/360 (6.7%)	2.16	0.006 / 0.000	97.8%

The target anchor is the modal choice on AIDS and Linux. IMDb relies more heavily on identity-preserving ZPAD and has longer ZPAD streaks. At $p = 5$, the selected candidate is the exact best within its pool in 88.9%–97.8% of runs, and mean oracle regret is below 0.06 AR percentage points. Adjacent-depth regressions occur in 5.8%–6.7% of transitions, with maximum drops of 1.88–2.56 points. These results are consistent with useful retention and fresh selection, while also showing that finite-shot selection can lose value at individual depths.

G Native-solution results on 12-, 14-, and 16-node graphs

G.1 Graph families and output metrics

For every $n \in \{12, 14, 16\}$, the experiment uses ten connected 3-regular graphs and ten connected Erdős–Rényi $G(n, m)$ graphs with $m = 3n/2$. The families share node count, edge count, and average degree within each size, but differ in topology. Each size–depth–method cell has 20 graphs and three optimizer seeds (60 runs), with the $q = 32$ configuration and $p = 1, \ldots, 5$.

The common output is one final cut solution $z^{(a)}_{i,s,p}$ per method–graph–seed–depth. Neural QAOA^2 supplies the native solution from a new run of its complete method. For BOLT-DA, Red-QAOA, and MG-control, the stored final output distributions are converted to one cut with a consistent deterministic post-optimization draw namespace across $n = 12, 14, 16$. This draw is a reporting step and does not affect training. Define the solution-level metrics by

$$
\begin{aligned}
R^{(a)}_{i,s,p} &= \frac{C_{G_i}(z^{(a)}_{i,s,p})}{C^{\star}_{G_i}}, \\
\mathrm{AR}^{(a)}_{\mathrm{sol},p} &= \frac{1}{N_G N_S} \sum_{i=1}^{N_G} \sum_{s=1}^{N_S} R^{(a)}_{i,s,p}, \\
\mathrm{MSE}^{(a)}_{\mathrm{sol},p} &= \frac{1}{N_G N_S} \sum_{i=1}^{N_G} \sum_{s=1}^{N_S} \big(1 - R^{(a)}_{i,s,p}\big)^2.
\end{aligned}
\tag{27}
$$

These solution-level quantities are reported separately from the expected-value metrics in Eq. (10); the former include variability in the returned cut itself.

G.2 COMPLETE POOLED RESULTS

Table 17: Native-solution results on the larger graph families. "Best" is the best displayed comparison-method mean, selected separately for AR and MSE among Red-QAOA, MG-control, and Neural QAOA^2. Each row averages 60 runs per method. MSE uses Eq. (27); AR leads are in percentage points. Displayed means are rounded; derived contrasts are retained at their reported precision.

n	p	BOLT AR	Best AR	AR lead (pp)	BOLT MSE	Best MSE	MSE reduction
12	1	0.759498	0.739739	1.98	0.072123	0.083061	13.17%
12	2	0.810090	0.766348	4.37	0.059400	0.077030	22.89%
12	3	0.821769	0.800225	2.15	0.047831	0.058195	17.81%
12	4	0.854187	0.829579	2.46	0.035143	0.038693	9.17%
12	5	0.874236	0.807006	6.72	0.035262	0.050173	29.72%
14	1	0.777795	0.763654	1.41	0.059488	0.070270	15.34%
14	2	0.795654	0.784299	1.14	0.057538	0.064509	10.81%
14	3	0.823945	0.796113	2.78	0.049884	0.054048	7.70%
14	4	0.831473	0.805705	2.58	0.043528	0.050500	13.81%
14	5	0.866042	0.838674	2.74	0.027952	0.035086	20.33%
16	1	0.773924	0.746602	2.73	0.059365	0.077558	23.46%
16	2	0.815396	0.786185	2.92	0.043114	0.062840	31.39%
16	3	0.831333	0.805559	2.58	0.044327	0.052446	15.48%
16	4	0.834825	0.804909	2.99	0.038186	0.048336	21.00%
16	5	0.824176	0.814536	0.96	0.042746	0.046085	7.24%

BOLT-DA has the highest mean native-solution AR and lowest native-solution MSE in all 15 size–depth cells. The minimum AR lead is 0.96 percentage points and the minimum MSE reduction is 7.24%, both at $n = 16, p = 5$. These results extend the observed quality comparison across graph sizes and the two matched-edge-count families. The displayed means do not by themselves establish simultaneous statistical significance or isolate the individual mechanisms at the larger sizes.

H HARDWARE COMPILATION PROTOCOL AND RESOURCE AUDIT

H.1 GRAPH SELECTION AND COMPILATION PROCEDURE

The resource audit in this appendix reports BOLT-DA, Red-QAOA, and MG-control. The graph-selection rule within the confirmatory holdout minimizes n subject to $5 \le n \le 10$, then minimizes the number of edges, then the selection rank. It selects AIDS index 240 ($n = 6, m = 6$), Linux index 332 ($n = 5, m = 4$), and IMDb index 1463 ($n = 7, m = 13$). The hardware-quality case in Section 5.6 uses the selected Linux graph.

The audited circuits target AGate-100 in the seven-qubit physical region $[6, 7, 8, 26, 27, 28, 29]$ with native basis $\{R_Z, SX, X, CZ\}$. Optimized circuit parameters are bound before transpilation. SABRE layout and routing are evaluated at optimization levels 2 and 3 with seeds 0–15. The chosen circuit minimizes

$$(N_{CZ},\ \text{depth},\ N_{1q},\ N_{\text{unitary}},\ \text{optimization level},\ \text{seed}) \tag{28}$$

lexicographically. BOLT-DA and Red-QAOA repeat each shared mixer angle β_ℓ across nodes; MG-control binds $\beta_{\ell,v}$ directly. The archived statevector checks bound the absolute expected-cut discrepancy by 1.74×10^{-13}. The resource-audit execution records use 8,192 shots per binding and retain the graph, layout, measurement map, circuit hash, and compilation choice.

H.2 LINUX PHYSICAL-CIRCUIT RESULTS

Table 18: Linux physical-circuit audit for BOLT-DA, Red-QAOA, and MG-control (index 332, seeds 5–7). Entries are means with seed ranges in brackets. This is a compilation-resource table, distinct from the solution-quality comparison in Table 3.

p	Method	Physical depth	CZ	1q gates	Native unitary total
2	BOLT-DA	38.67 [38,39]	16.00 [16,16]	69.00 [67,70]	85.00 [83,86]
	Red-QAOA	38.33 [38,39]	16.00 [16,16]	68.00 [67,70]	84.00 [83,86]
	MG-control	38.33 [38,39]	16.00 [16,16]	68.00 [67,69]	84.00 [83,85]
3	BOLT-DA	48.67 [39,54]	21.33 [16,24]	86.33 [70,96]	107.67 [86,120]
	Red-QAOA	53.33 [53,54]	24.00 [24,24]	94.33 [93,96]	118.33 [117,120]
	MG-control	54.33 [54,55]	24.00 [24,24]	96.00 [95,97]	120.00 [119,121]
4	BOLT-DA	59.00 [39,70]	26.67 [16,32]	104.00 [70,123]	130.67 [86,155]
	Red-QAOA	69.33 [68,70]	32.00 [32,32]	121.67 [119,123]	153.67 [151,155]
	MG-control	69.00 [68,70]	32.00 [32,32]	121.33 [120,123]	153.33 [152,155]

At $p = 3, 4$, the selected BOLT-DA circuits reduce mean CZ count and physical depth on this Linux instance, while $p = 2$ is essentially tied. This audit shows that retaining a shallower solution can lower compiled resources in the tested case. It is not a substitute for the multi-graph finite-shot analysis or evidence of a universal hardware advantage.

I ADDITIONAL PAIRED UNCERTAINTY FOR REPORTED COMPARATORS

This section complements the mean results in Sections 5.2 and 5.3 with paired uncertainty estimates computed from the same archived confirmatory runs. No optimizer trajectory, candidate decision, or reported mean is changed. For each graph, seed-level ARs and seed-level squared gaps are averaged separately over the three optimizer seeds, as in Eq. (10); squared-gap losses are not obtained by squaring a seed-averaged gap. Paired percentile intervals then resample graphs jointly across methods using 20,000 bootstrap replicates. Thus, the graph—rather than the optimizer seed—is the uncertainty unit.

I.1 FINAL-DEPTH CONTRASTS AT THE PRIMARY LOW-SHOT SETTING

Table 19 reports the final-depth contrasts at $q = 32$ against the three comparison methods already used in the main small-graph experiment. Positive ΔAR and positive MSE reduction favor BOLT-DA. The graph-paired AR intervals exclude zero for every comparison. MSE reductions are recomputed from the corrected pooled MSE means and are reported as point estimates here; no MSE interval is reconstructed from aggregate values.

Table 19: Final-depth contrasts at $q = 32, p = 5$. AR differences are in percentage points with pointwise 95% graph-paired bootstrap intervals after averaging the three optimizer seeds within each graph. MSE reductions are ratios of the corrected pooled MSE means and are shown without a reconstructed bootstrap interval.

Dataset	Comparator	ΔAR (pp) [95% CI]	MSE reduction (%)
AIDS	Red-QAOA	12.970 [9.364, 17.092]	75.00
AIDS	Neural QAOA^2	3.341 [2.125, 4.545]	34.89
AIDS	MG-control	17.044 [15.840, 18.283]	78.42
Linux	Red-QAOA	14.659 [11.955, 17.666]	74.26
Linux	Neural QAOA^2	3.024 [1.690, 4.397]	32.33
Linux	MG-control	15.503 [14.192, 16.864]	73.47
IMDb	Red-QAOA	34.353 [29.285, 39.519]	99.30
IMDb	Neural QAOA^2	4.288 [3.662, 4.919]	81.00
IMDb	MG-control	7.013 [6.509, 7.541]	88.40

I.2 Cross-budget paired robustness against Neural QAOA^2

The mean curves in Section 5.3 compare $q \in \{32, 128, 512\}$. To verify that the final-depth ordering is not an artifact of a single nominal-shot setting, Table 20 gives AR graph-paired contrasts and MSE point-estimate reductions against Neural QAOA^2 at every reported budget. Across all nine (dataset, q) cells, the AR intervals remain strictly positive and the corrected pooled-MSE reductions are positive.

Table 20: BOLT-DA minus Neural QAOA^2 contrasts at $p = 5$ across the three reported nominal-shot settings. AR entries contain pointwise 95% graph-paired bootstrap intervals. MSE reductions are ratios of pooled MSE means; the underlying graph and optimizer-seed sets are fixed across q.

Dataset	q	ΔAR (pp) [95% CI]	MSE reduction (%)
AIDS	32	3.341 [2.125, 4.545]	34.89
AIDS	128	2.190 [1.259, 3.145]	26.05
AIDS	512	2.439 [1.383, 3.469]	27.60
Linux	32	3.024 [1.690, 4.397]	32.33
Linux	128	2.522 [1.499, 3.579]	27.02
Linux	512	1.842 [0.745, 2.864]	21.08
IMDb	32	4.288 [3.662, 4.919]	81.00
IMDb	128	3.582 [3.166, 3.978]	72.29
IMDb	512	2.905 [2.441, 3.382]	73.74

The AR intervals are pointwise rather than simultaneous confidence bands. Their role is to quantify the stability of the already reported final-depth AR comparisons; the MSE columns report corrected aggregate effect sizes without reconstructing unavailable paired intervals from summary values.

J Additional BOLT-DA budget audit and reproducibility details

J.1 Exact integer shot allocation at $q = 32$

The main method specifies the raw-shot split algebraically. Table 21 expands that definition into the exact integer allocations used at the primary $q = 32$ setting. The S_p and $S_{\le p}$ columns are the common online raw measurement-shot budgets used by all four reported methods; the S_p^T, S_p^D, S_p^R split and candidate allocation are specific to BOLT-DA. The rounding remainder is assigned to fresh selection after flooring the target and donor shares, and the selection remainder is distributed in the fixed candidate-list order. This table is a deterministic audit of the stated budget rule, not an additional experiment.

Table 21: Exact incremental raw-shot allocation for BOLT-DA at $q = 32$. "Per candidate" gives the minimum–maximum number of fresh target-side selection shots when integer remainders are present.

p	S_p	S_p^T	S_p^D	S_p^R	K_p	Per candidate	$S_{\le p}$
1	8,192	6,553	1,228	411	2	205–206	8,192
2	16,384	13,107	2,457	820	4	205	24,576
3	24,576	19,660	3,686	1,230	4	307–308	49,152
4	32,768	26,214	4,915	1,639	4	409–410	81,920
5	40,960	32,768	6,144	2,048	4	512	122,880

At $p = 5$, for example, the fresh-selection stage receives 2,048 shots, so each of the four candidates is remeasured with exactly 512 original-graph shots. This clarifies that the nominal parameter q controls the overall budget in Eq. (5); it is not the literal shot count of every individual candidate evaluation.

J.2 Fixed optimizer and randomization conventions

For completeness, Table 22 collects the control settings that are fixed across the reported confirmatory depth ladders. These entries restate the archived execution protocol in one location and do not alter any reported result.

Table 22: BOLT-DA execution conventions for the reported finite-shot experiments.

Item	Fixed convention
Depths	$p = 1, \dots, 5$
Optimizer seeds	$\{5, 6, 7\}$ on the confirmatory graph set
Nominal-shot settings	$q \in \{32, 128, 512\}$; primary setting $q = 32$
Calls factor	$c = 256$, so $S_p = 256pq$
Budget split	target/donor/selection $= 80/15/5$
Nominal calls	target $256p$; donor $64p$
Differential evolution	mutation factor $F = 0.72$; crossover probability $CR = 0.88$
Trial comparisons	trial and incumbent are remeasured with fresh samples
Final branch choice	final population is remeasured with fresh samples
Donor reuse	one reduced graph is constructed per graph–seed run and reused across depth
Cross-depth continuation	target and donor branches retain their own preceding search outputs
Candidate tie handling	fixed list order; the earlier candidate wins exact ties
Guard coefficient	$\lambda = 1/4$ in Eq. (16)
Evaluation separation	exact expected values are computed only after the finite-shot output is frozen

K Pool-relative finite-shot retention guarantee

The anchor-relative guarantee in Appendix A controls how far the returned candidate can fall below the target-search anchor. The presence of identity padding yields an additional pool-relative statement that connects the selection step directly to depth-wise solution retention. The argument conditions on all search history before the fresh selection samples are drawn, so the candidates may themselves have been produced by adaptive noisy optimization.

Let $n_{\min} = \min_k n_k \ge 2$, and for a chosen $\delta_p \in (0, 1)$ define

$$\epsilon_p = m\sqrt{\frac{\log(2K_p/\delta_p)}{2n_{\min}}}, \qquad h_p = \frac{\lambda m}{\sqrt{2(n_{\min} - 1)}}. \tag{29}$$

Proposition 3 (Loss relative to the available candidate pool)**.** *Conditioned on the candidate pool and its fresh-shot allocation, the selection rule in Eq.* (16) *satisfies, with probability at least* $1 - \delta_p$*,*

$$\mu_{\text{out}} \ge \max_{1 \le k \le K_p} \mu_k - 2\epsilon_p - h_p. \tag{30}$$

For $p > 1$ in the ideal circuit model, because the candidate pool contains $\mathrm{ZPAD}_p(\widehat{\theta}_{p-1})$, this further implies

$$J_G(\widehat{\theta}_p) \geq J_G(\widehat{\theta}_{p-1}) - 2\epsilon_p - h_p. \tag{31}$$

Proof. For observations $x_1, \ldots, x_n \in [0, m]$, the empirical variance with normalization $1/n$ is at most $m^2/4$. Therefore the unbiased sample variance obeys

$$\frac{s^2}{n} \leq \frac{m^2}{4(n-1)}.$$

For any challenger j, the empirical replacement threshold in Eq. (16) is consequently bounded by

$$\lambda\sqrt{\frac{s_a^2}{n_a} + \frac{s_j^2}{n_j}} \leq \frac{\lambda m}{2}\sqrt{\frac{1}{n_a - 1} + \frac{1}{n_j - 1}} \leq h_p.$$

If the challenger is accepted, it has the largest non-anchor empirical mean and exceeds the anchor by more than the guard threshold. If the anchor is retained, its empirical mean is within h_p of the best empirical mean in the pool. Hence in either case

$$\widehat{\mu}_{\mathrm{out}} \geq \max_k \widehat{\mu}_k - h_p.$$

Hoeffding's inequality and a union bound over the K_p fresh candidate measurements give $|\widehat{\mu}_k - \mu_k| \leq \epsilon_p$ simultaneously for all candidates with probability at least $1 - \delta_p$. Subtracting one concentration radius for the selected candidate and one for a true pool maximizer gives Eq. (30). Proposition 2 then gives Eq. (31) because identity padding preserves the previous selected expectation in the ideal circuit model. □

For a finite depth ladder with at least two fresh samples per candidate at every depth $p = 2, \ldots, P$, choosing $\delta_2, \ldots, \delta_P$ with $\sum_{p=2}^{P} \delta_p \leq \delta$ and applying a union bound gives the depth-wise bounds jointly. Summing them yields

$$J_G(\widehat{\theta}_P) \geq J_G(\widehat{\theta}_1) - \sum_{p=2}^{P}(2\epsilon_p + h_p)$$

with probability at least $1 - \delta$. The result is a finite-shot retention bound: it formalizes that the equal-value identity-padded candidate limits how much value can be lost through noisy fresh selection, even though it does not require the optimizer to follow the previously selected trajectory. The additive slack is generally positive and can be large at low shot counts. Consequently, this bound does not guarantee nondecreasing realized solution quality, calibrate the empirical guard as a confidence test, or extend ideal-circuit identity preservation to noisy hardware executions.

L FAMILY-RESOLVED SCALE RESULTS FOR BOLT-DA

The pooled 12-, 14-, and 16-node results in Appendix G combine two graph families with identical node and edge counts at each size. Table 23 shows the corresponding BOLT-DA decomposition at $p = 5, q = 32$. These are subgroup summaries of the already reported scale runs, not additional training runs. Each family contains ten graphs and three optimizer seeds.

Table 23: Family-resolved BOLT-DA native-solution results at $p = 5, q = 32$. ER-like denotes connected $G(n, m)$ graphs with $m = 3n/2$; RRG3 denotes connected 3-regular graphs.

n	Family	Runs	Mean AR	Mean MSE
12	ER-like	30	0.875278	0.043900
12	RRG3	30	0.873194	0.026624
12	Pooled	60	0.874236	0.035262
14	ER-like	30	0.851307	0.033157
14	RRG3	30	0.880776	0.022747
14	Pooled	60	0.866042	0.027952
16	ER-like	30	0.811988	0.048498
16	RRG3	30	0.836364	0.036995
16	Pooled	60	0.824176	0.042746

At every tested size, the $p = 5$ mean AR exceeds 0.81 in both topology families. The 12-node AR means are close, while the 3-regular family has higher mean AR and lower MSE than the matched-edge-count ER-like family at 14 and 16 nodes. These BOLT-DA-only subgroup summaries provide a more granular view of the pooled experiment without changing its native-solution evaluation protocol. They do not by themselves establish superiority over other methods within either individual graph family.