The Constraints Inherited from Locality on Quantum Optimization

A dissertation

submitted by

Sos Oganessian,

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

August 2025

ADVISOR: Prof. Peter Love

The Constraints Inherited from Locality on Quantum Optimization

Sos Oganessian

ADVIS0R: Prof. Peter Love

When introduced, the Quantum Approximate Optimization Algorithm (QAOA) was a leading candidate for possible quantum advantage on noisy intermediate-scale quantum (NISQ) devices. The QAOA is a local algorithm whose approximation quality depends on its ansatz depth p. For $p = 1$, Farhi *et al.* established a problem-size independent worst-case approximation bound for 3-regular MaxCut, and advocates hypothesize the existence of a fixed p_{crit} beyond which the QAOA outperforms classical approximations on some NP-hard problem. In the decade since, extensive analysis has been unable to resolve this question. No p parameterized worst-case approximation guarantee has been established, and fixed-p worst-case approximation ratios remain below classical counterparts. Further complicating matters is the existence of Overlap Gap Property (OGP) clustering in the solution space of some NP-hard problems, which yields provable problem-size-dependent obstructions that prevent local algorithms from accessing some near-optimal solutions.

This work begins by investigating the practical relevance of asymptotic OGP obstructions as on random Max-k-SAT and show that $n > 10^{12}$ before we see a $p = 2$ restriction, implying a looseness in the OGP obstructions for random Max-k-SAT. We also introduce a tighter measure of the locality required for an algorithm to "see the whole

graph." Simulations demonstrate that the QAOA circumvents OGP obstruction once its ansatz depth spans this diameter. Beyond this threshold, the QAOA achieves parity on OGP and non-OGP instances, suggesting no further performance penalty once its locality demands are met. Finally, we characterize how the minimum ansatz depth for circumventing OGP obstructions scales with variable (n), clause (m), and arity (k) counts.

We then turn to mechanisms that relax locality constraints and introduce non-local mixers that embed long-range correlations into each QAOA layer without increasing p. We also propose Ballistic Optimization (BO), a "single-layer" continuous-time evolution under the combined Hamiltonian $\gamma H_{\mathrm{C}} + \beta H_{\mathrm{B}}$, without Trotterization. On a set of non-isomorphic 3-regular graphs with $n = 12$ BO attains a mean approximation ratio exceeding that of depth-one QAOA by ≈ 0.13.

Finally, we provide a formal account of BO's operator growth to explain its performance advantage over QAOA. We analyze BO's evolution in the Heisenberg picture on the "ring of disagrees" problem via the Baker-Campbell-Hausdorff (BCH) expansion and show that the amplitude of the single-qubit σ^x term after $k = 2q$ commutators grows as $(8\beta\gamma)^q \, 2^{q-1} \mathrm{Cat}_q$ where Cat_q is the q-th Catalan number. We then show that BO explores a much richer space of local configurations than depth-1 QAOA, yet it still needs evolution times that scale linearly with problem size before global support is guaranteed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Early quantum algorithms, such as Shor's factoring, made a strong case for a new computational model that could address problems long considered intractable [Sho94]. However, in the three intervening decades since the introduction of Shor's algorithm, due in no small part to the complexity associated with developing a quantum algorithm, much of that fanfare has tempered. Quantum amplitude-amplification techniques demonstrate a strict quadratic speed-up for unstructured global search and thereby motivate optimization-oriented quantum strategies [Gro96, BHMT02, FGG$^+$01].

Optimization is a type of structure search, namely extremum finding. The ultimate goal is to find the maximum or minimum of a known cost function $f\colon X \to \mathbb{R}$. Here $x \in X$ is a finite set of solutions that comprise the domain of the objective function f. Each optimization problem has an explicit set C of constraints (example constraint: an answer must be even). A feasible solution x^* is any element of X that satisfies every constraint in C. Three major quantum algorithms for optimization have been proposed in the thirty years since Shor: Adiabatic Quantum Computing (AQC) [FGG$^+$01], the Quantum Approximate Optimization Algorithm (QAOA) [FGG14], and Decoded Quantum Interference (DQI)

[JSW+25, JSW+24].

AQC, introduced in 2000, was the first quantum-optimization protocol. AQC is an exact algorithm that evolves a system under a time-dependent Hamiltonian from an easily prepared ground state to one that encodes the solution. The adiabatic theorem bounds from below the runtime of AQC as function of minimum instantaneous gap between the ground and first excited states along $H(s)$.

$$T \;\geq\; \Delta_{\min}^{-2}, \qquad \Delta_{\min} \;=\; \min_{s\in[0,1]} \big[E_1(s) - E_0(s)\big]$$

where $E_0(s) \leq E_1(s)$ are the lowest two eigenvalues of $H(s)$ [JRS07]. If $\Delta_{\min} = O(2^{-n})$, then $T = \Omega(2^{2n})$, so worst-case instances remain intractable.

At the other end, Decoded Quantum Interferometry (DQI) is a recently proposed quantum approximation algorithm that uses a quantum Fourier transform followed by classical decoding of low-density parity-check codes to concentrate amplitude on bit strings with large objective value [JSW+25, JSW+24]. Query-complexity bounds and numerical studies indicate exponential speedups on bounded-degree quadratic unconstrained binary optimization instances, but no tight worst-case or average-case runtime guarantees are known.

Perhaps the most studied quantum optimization algorithm is the Quantum Approximate Optimization Algorithm (QAOA). The QAOA is a variational quantum approximation algorithm [FGG14] applicable to a wide range of NP-hard optimization problems [Luc14]. As with other variational quantum algorithms, the QAOA is a hybrid quantum-classical algorithm [PMS+14, TCC+22]. It employs a parameterized ansatz (a quantum circuit with

adjustable parameters that generates candidate solutions)

$$|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle = \prod_{j=1}^{p} e^{-i\beta_j H_{\mathrm{B}}} e^{-i\gamma_j H_{\mathrm{C}}} |+\rangle^{\otimes n}$$

and a classical optimizer that updates these parameters ($(\beta.\gamma)$) based on measurement outcomes. While the QAOA looks like an alternating Trotterization of the AQC Hamiltonian, the QAOA is an approximation algorithm. Its approximation quality is measured by per-instance a worst-case approximation ratio between its candidate solution and the true optimum

$$\alpha \;=\; \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} \frac{\langle \psi_p(\boldsymbol{\gamma},\boldsymbol{\beta}) \,\big|\, H_{\mathrm{C}} \,\big|\, \psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle}{c_{\mathrm{opt}}}$$

where H_{C} is the problem Hamiltonian and c_{opt} is the optimal eigenvalue. The structural clarity of the QAOA, inherited from its two-operator decomposition, renders the protocol tractable for rigorous study: whereas AQC is benchmarked by worst-case runtime derived from spectral-gap, the relevant metric for an approximation algorithm is the approximation ratio.

An algorithm is a c-approximation if it always yields a solution at least fraction c of the optimal value for any input. The QAOA's approximation quality is controlled by a tunable ansatz depth p, which monotonically improves approximation quality. As p grows, the approximation ratio improves or remains the same, and in the limit $p \to \infty$ the QAOA converges to the optimal solution. Increasing p also increases the overall circuit depth and the number of parameters to be optimized. As p grows, the optimization becomes prohibitively difficult [BK21] and is hindered by the curse of dimensionality [WHJR18, BBC57].

A substantial share of the QAOA literature is devoted to bounding ρ_p, mapping how it scales with p, and delineating regimes in which QAOA either matches or fails to exceed classical approximation guarantees for random and worst-case instances. Part of

what enables such analysis of QAOA's performance as a function of p is its intrinsically local structure, which constrains each layer to act nontrivally only within a bounded-radius "causal cone" around each qubit.

In quantum terms, a k-local operator on n qubits acts non-trivially on at most $k \ll n$ of those qubits at a time [NC11]. Algorithmically, locality refers to the limited "neighborhood of influence" that an algorithm's decision rule can access. The QAOA at depth p is a local algorithm in this sense: each additional alternation of problem and mixer Hamiltonians can only propagate correlations one edge farther out in the problem graph. Equivalently, one QAOA layer extends any qubit's influence by at most one hop along graph edges, so a depth-p ansatz confines each output bit's dependence to the radius-p neighborhood of the corresponding input bit [WL21]. This inherent locality of shallow QAOA circuits underpins both their tractability for analysis and their potential limitations.

As noted above, the QAOA's performance is non-decreasing with p. If $\langle C \rangle_p$ denotes the maximum expected objective value attained at depth p, then $\langle C \rangle_{p+1} \geq \langle C \rangle_p$, since any depth-$p$ schedule embeds in depth $p+1$ by setting the extra angle parameters to zero [FGG14]. This nested structure invited early attempts to relate p directly to approximation quality. Most notably, Farhi showed that at $p = 1$ the QAOA achieves an approximation ratio of 0.6924 on every 3-regular Max-Cut instance [FGG14]. This value lies above the random-guess baseline of 0.5 yet below the 0.8786 approximation guarantee of the classical Goemans–Williamson algorithm [FGG14, GW95]. Seven years later, Wurtz and Love improved the worst-case guarantee to 0.7559 at *et al.*, albeit under the additional assumption that the input graph has girth greater than 5 [WL21]. While high girth is a mild constraint, no further work has extend the *et al.* result to arbitrary graphs (recall that Farhi *et al.'s* $p = 1$ guarantee was likewise limited to 3-regular graphs). To date, no general worst-case approximation guarantee is known for QAOA at any fixed p.

Even affording the above constraints, progress in understanding QAOA's worst-case approximation quality beyond *et al.* has proven difficult. Basso introduced a recursive formula that characterizes a measure of expected performance of the QAOA on large-girth regular graphs for arbitrary depth p [BFM$^+$22]. Using this approach, numerical evidence up to $p = 11$ suggests that beyond some small p, the QAOA's average-case performance might eventually surpass that of comparable classical local algorithms [BFM$^+$22]. Building on these techniques, Farhi *et al.'s* have recently show a higher-depth performance guarantee: on 3-regular Max-Cut instances with girth ≥ 16, a depth-7 QAOA can produce a cut covering at least 79.0% of edges [FGRV25]. While notable as the first rigorously certified quantum approximation above 75.59% on Max-Cut, this falls short of the 87.86% achieved by the Goemans–Williamson SDP and underscores the challenges in analytically bounding QAOA performance as a function of the depth p (even when p grows polynomially with the problem size). Nevertheless, some have conjectured the existence of a critical polynomially or fixed depth $p_{\text{crit}} > 1$ at which QAOA's minimum worst-case approximation ratio would surpass that of the best known classical algorithm on an NP-hard problem [FH19]. However, any circuit whose depth p is independent of the problem size exploits problem structure only within an $O(1)$-radius neighborhood of each input element, so it is unclear why such an advantage would arise at fixed depth. Indeed, long-standing results on classical greedy and local heuristics show that once an algorithm is confined to a fixed neighborhood size, its approximation quality is capped and cannot asymptotically exceed certain thresholds [SG76, Lin92].

Even if such a p_{crit} exists, achieving it in practice would face a second obstacle: the complexity of optimizing a large number of variational parameters. Each QAOA layer introduces two new angles, so a depth-p circuit has $2p$ free parameters that must be tuned via a difficult non-convex optimization process. This "curse of dimensionality" is evident

even at moderate p [BK21] and only worsens with increasing depth. Moreover, certain QAOA variants have been proven to require unfeasibly large depth to attain nontrivial guarantees: for instance, a QAOA-like algorithm employing Grover-style mixers (instead of the standard transverse-field mixer) needs $p = \Omega(n^\gamma)$ layers (for some $\gamma > 0$) to achieve any constant-factor approximation in the worst case [BBL$^+$25]. On the other hand, there are encouraging signs that optimizing the QAOA may be easier on average than in the worst case. Empirically, for several random problem ensembles, the optimal QAOA angles appear to concentrate around instance-independent values as n grows [ARCB21, FGGZ22, BFM$^+$22]. In other words, as the problem size increases, one can use "universal" angle schedules that yield near-optimal expected performance across almost all instances of a given type. This parameter concentration phenomenon, first observed in numerical studies, has since garnered analytical support in certain models [CLSS22a, Cro18, GM19, ARCB21, BFM$^+$22].

While the above worst-case analyses have not definitively ruled out a quantum advantage at fixed p, a different line of inquiry on random instances has identified more concrete obstacles. Chief among these is the Overlap Gap Property (OGP), a structural feature of the solution space in many constraint satisfaction problems. An instance is said to exhibit the OGP when every pair of its near-optimal solutions are either almost identical or almost completely dissimilar, with no moderate-overlap pairs [GS13]. More formally, there exist constants $0 < \beta < \delta < 1$ such that all η-optimal solutions (those achieving at least fraction η of the optimum) have Hamming overlap either above $1 - \beta$ or below $1 - \delta$ [GS13]. This creates a forbidden band of "medium" overlaps in $[\,1-\delta,\ 1-\beta\,]$ that separates distinct clusters of high-quality solutions.

The OGP poses a barrier to local optimization algorithms: to transform one cluster of near-optimal solution into another, an algorithm confined to local moves would have to pass through intermediate states lying in the forbidden overlap band. For the depth-p

output state of the QAOA that only encodes only local correlations this means a shallow circuit cannot smoothly interpolate between distant optimal solution clusters.

Indeed, recent works have leveraged the OGP to prove rigorous performance limitations on constant-depth QAOA. Chou [CLSS22b] and Jones [JMSS23] showed that for broad classes of random Max-CSP instances (including random Max-k-SAT), any QAOA with $p = O(1)$ will, with high probability, fail to produce solutions beyond a certain approximation ratio threshold. In other words, achieving near-optimal solutions on these problems with high probability would require the ansatz depth p to grow at least poly-logarithmically with n [CLSS22b, JMSS23]. Notably, this condition is necessary but not sufficient. Surpassing the OGP's locality threshold is only a prerequisite for success, and even once that threshold is crossed, other factors may still limit an algorithm's performance.

It is important to note that the OGP-based obstructions are asymptotic in nature and current proofs imply that they manifested at extremely large problem sizes. For example, analysis detailed in Chapter 3 (Section3.1) indicates that a depth-2 QAOA on random Max-k-SAT would require on the order of $n \sim 10^{12}$ variables for the OGP's "forbidden region" to decisively thwart the algorithm. More generally, the locality thresholds proven in Refs. [CLSS22b, FGG20] impose only very shallow depths (e.g., $p < 7$) when translated to concrete problem sizes scales that far exceed what is possible on classical computers [SP23]. In fact, even at the modest circuit depths that today's NISQ devices can achieve, the input sizes needed for obstructions to apply exceed quantum utilty bounds. Thus, while OGP-centered analysis provides proven obstructions, in practice it leaves open a substantial regime in which a fixed-p QAOA might still operate without encountering serious obstacles.

At the same time, even in the average case, the QAOA has yet to empirically outperform the best classical algorithms on any NP-hard problem. So the conjecture that even at fixed moderate depth, the QAOA might surpass classical approximations on

specific problems once past a certain threshold p_{crit} [FH19] remains unrealized. And all evidence so far indicates that at the small depths accessible on near-term hardware, QAOA's approximation ratios remain below those achieved by state-of-the-art classical heuristics or relaxations.

Nevertheless, some researchers remain optimistic about the QAOA's prospects. One avenue to boost QAOA performance is to relax the algorithm's locality by incorporating nonlocal operations. As mentioned, the standard QAOA uses cost and mixer Hamiltonians that are sums of local terms. However, this locality is not a fundamental requirement of the QAOA. By introducing long-range interactions, one might extend the QAOA's reach beyond a fixed neighborhood. For example, adding auxiliary couplings between distant qubits in H_C (so-called "phantom" ZZ-edges that do not change the cost function's value) or using a complete-graph XY mixer that connects every pair of qubits transforms the QAOA ansatz into a nonlocal circuit [LTE25, CBE20]. Such modifications allow a single QAOA layer to propagate correlations across the entire graph. In principle, a suitably designed nonlocal QAOA could evade the limitations that afflict strictly local (constant-depth) algorithms, potentially bypassing OGP barriers even at shallow circuit depths.

The considerations above motivate the research presented in this dissertation. Chapter 3 investigates how the QAOA's locality interacts with the Overlap Gap Property (OGP) on random Max-k-SAT formulae and considers how the OGP constrains shallow QAOA in practice. Random Max-k-SAT formulae were generated well above the satisfiability threshold, OGP instances were isolated with a brute-force detector, and depth-p circuits were simulated up to $p = 7$. We consider the analytic lower bounds on ansatz depth that any local algorithm must satisfy to overcome OGP-induced obstructions, then compare those bounds with numerical simulations at $p \leq 7$. Across $k = 3$–6 and $n \leq 26$ the ansatz clears the forbidden-overlap region whenever $2p + 1 \geq \operatorname{diam}\big(H_{\phi_k(n,m)}\big)$, i.e. when the QAOA's

support spans the hypergraph diameter.

We also show that, for Max-k-SAT, the asymptotic obstruction $p \geq \Omega\big(\log n\big)$ predicted by coupled-OGP theory appears vacuous until $n > 10^{12}$. Within the accessible regime ($n \leq 26$) once locality requirements are met, QAOA's approximation quality improves roughly linearly with depth and is indistinguishable on OGP and non-OGP instances. Scaling sweeps show that the minimum depth rises with variable count and clause arity (variable per clause) but remains insensitive to further increases in clause density, suggesting that hardness is driven by geometric—not combinatorial—growth.

Chapter 4 turns from limits of locality and instead asks whether locality can be sidestepped rather than overcome. Three non-local mixer families are introduced and that embed long-range correlations into each QAOA layer without increasing p. While none outperform the standard QAOA, an ensemble comparison on 3-regular graphs with $n = 14$ shows tha $X \otimes X$ couplings and distance-weighted graph mixers conserve mean performance at $p = 1$ while traversal mixers suppress outlier mis-cuts by incorporating frustration topology.

We then propose Ballistic Optimization (BO), which defines the optimization as a continuous-time evolution. Ballistic Optimization prepares the single "layer" state

$$\exp\big[-it\,(\beta H_B + \gamma H_C)\,t\big]\;\;|+\rangle^{\otimes n}$$

We show that on every non-isomorphic 3-regular graph with $n = 12$, ballistic circuits surpass depth-one QAOA by ≈ 0.13 in mean approximation ratio and approach the worst Goemans–Williamson cuts.

Chapter 5 supplies an analytic account of BO's advantage. For the antiferromagnetic cycle ("ring of disagrees") the Baker–Campbell–Hausdorff expansion shows that the amplitude of any single-qubit σ^x term after $k = 2q$ commutators is well approximated

by $(8\beta\gamma)^q\, 2^{q-1}\mathrm{Cat}_q$, with odd orders vanishing. Operator locality inside the expansion grows to full $N-1$-qubit support at $k = 2N-2$. Competing Catalan-number growth and factorial suppression yield a turnover at $q_* \approx 4\sqrt{\beta\gamma}\, t$; global support is therefore guaranteed only when $t \gtrsim N/\sqrt{\beta\gamma}$. Inside the BCH expansion, for commutator order $k \leq 2N-2$, the operators that contribute to the expectation are 1-local. During the intermediate algebra, the commutator chain temporarily visits Pauli words that touch at most $\lfloor k/4 \rfloor + 2$ qubits before the support shrinks back to one qubit. When the depth exceeds $2N-2$, a word that acts on $N-1$ qubits, namely $I_j X^{\otimes(N-1)}$, appears.

We close the thesis with some discussion and direction for future work. These include investigating QAOA's unexpected insensitivity to increases in clause count, analyzing ballistic optimization beyond the ring of disagrees, and clarifying the relative scalability of QAOA and ballistic optimization in terms of approximation performance and problem size.

1.1 Distance Measures

Throughout this dissertation, multiple distance notions are used; these measures have other corresponding measures when discussing the same class of problem or problem instance in different contexts. When we discuss raw graphs, graph distance, the number of edges along the shortest path between any pair of vertices u, v in the graph, is used. We use this measure when introducing the hypergraph diameter associated with a k-SAT instance in Chapter 3, and again in Chapter 4, when the unfrustrated subgraphs are encoded in the construction of the graph-distance mixers.

When we are considering an optimization problem on a graph (or one that has its structure modeled by a graph), we switch from discussing the graph to discussing the bitstring that represents a candidate solution to the optimization problem. In the case of MaxCut, the bitstring represents the partition assignments. Here, when discussing the

distance between assignments, we consider the Hamming distance (the number of bits that differ in a pair of assignments). This is a distance in the solution space of optimization. Since the Hamming distance is between assignments, it does not depend on the graph structure, just on the number of vertices. For a pair of n-vertex graphs G_1 and G_2, the elements of the solution space (or feasible set) are all $s_i \in \{0,1\}^n$, and the Hamming distance between any pair of candidate solutions s_i and s_j is the same. When discussing a property of the solution space, the Overlap Gap Property introduced in Section 2.6 and discussed throughout this dissertation, we are speaking in terms of Hamming distance (or its inverse, the overlap).

Finally, after we encode a problem instance to a Pauli word P, and discuss the quantum algorithm, we discuss the operator locality or support size $\ell(P)$. This is the number of qubits on which P acts non-trivially. Locality or support is used in Chapters 3 and 4 when discussing the locality of the QAOA and again in Chapter 5, which tracks the growth of ℓ inside the Baker–Campbell–Hausdorff expansion.

These measures are often interconnected. A depth p QAOA ansatz is comprised of operators with $2p$ support, and, in the case of MaxCut, this is equivalent to a radius-r ball of graph distance around each vertex. Similarly, overlap-gap analysis discusses the forbidden region between two bitstrings $x, y \in \{\pm 1\}^n$ that differ on $c\,n$ coordinates for some constant $c > 0$, yet assign the same labeled tree to every radius-r ball in the graph. If the forbidden region were absent, one could connect x and y by a chain of near-optimal strings differing in $o(n)$ bits each. Since it exists, the local view presented to a local quantum algorithm with $2p$ support is identical under x and y and is indistinguishable to any rule whose decision depends solely on those neighborhoods in the graph, the distance in the bitstring, and the support in the operator. As such, optimization procedures that adjust based on radius r receive no directional signal that distinguishes the two.

1.2 Summary of QAOA Research

1.2.1 Approximation Guarantees and Worst-Case Performance

In their original proposal for the QAOA, Farhi *et al.* proved that a depth-1 circuit achieves an approximation ratio of at least 0.6924 for MaxCut on every 3-regular graph, a value above the random baseline 0.5 yet below the 0.8786 guarantee of the Goemans–Williamson semidefinite program [FGG14, GW95]. Subsequent work extended the bound on 2-regular graphs: when the girth g (the length of a shortest cycle) satisfies $g > 5$, depth-2 QAOA reaches 0.7559, and under the high-girth conjecture depth-3 attains cut fraction 0.7924 [WL21].

Basso *et al.* derived a recursive tree formula valid for arbitrary depth on large-girth D-regular graphs (where D is the vertex degree). Numerical evaluation of the formula shows a non-decreasing cut fraction c_p; when $p \approx 11$, c_p rises above 0.782, the best proven cut-fraction bound supplied by Marwaha's 2-local classical algorithm on the same family of graphs [BFM$^+$22, BM22a]. Basso's result is numerical, so $0.782\,|E|$ is still the best provable cut-fraction lower bound for all classical algorithms whose decision at each vertex depends only on its radius-two neighborhood. Large-girth arguments do not uniformly apply to random regular graphs. A random d-regular graph's girth is inversely proportional to d. Wybo and Leib show that for fixed QAOA depth p the asymptotic approximation ratio rises with d for MaxCut but falls with d for Maximum Independent Set (finding the largest set of vertices no two of which share an edge) [WL24].

Farhi *et al.* recently proved that on every 3-regular graph with girth $g \geq 16$ a depth-7 QAOA produces a cut of at least $0.790\,|E|$ edges [FGRV25]. This analytic constant exceeds the $0.782\,|E|$ benchmark from Marwaha, giving the first closed-form guarantee above that classical locality barrier at fixed depth $p \leq 7$. These absolute bounds are not

directly comparable with ratio-based guarantees such as Goemans–Williamson's 0.8786 or the 0.9326 approximation ratio proved for cubic graphs by Halperin–Livnat–Zwick [HLZ04]: no universal lower bound is known for $\mathrm{OPT}(G)/|E|$, where $\mathrm{OPT}(G)$ is the number of edges in a maximum cut of G; ratio results therefore do not translate into explicit cut fractions.

Outside MaxCut, depth lower bounds indicate that a QAOA circuit of fixed constant depth cannot, in the worst case, achieve any constant-factor approximation. Campos, Rabinovich, and Uvarov convert continuous-time unstructured search into an alternating QAOA circuit and prove that achieving success probability at least $1 - \varepsilon$ demands a depth in the window

$$\sqrt{N} \;<\; p \;<\; N^{\frac{1}{2}+c} \quad (\forall c > 0).$$

with N the search-space size; this is the first analytic upper bound showing QAOA must be super-constant yet sub-linear for an oracle task [CRU24]. For random Max-k-XOR, Müller *et al.* report that a classical mean-field optimizer matches or outperforms QAOA unless the depth grows steeply with the clause locality—rising empirically from $p \approx 50$ at $k = 3$ to $p \approx 770$ at $k = 10$ to reach 99 % of the ground-state energy [MSFB25]. These empirical and analytic upper bounds are mirrored by the worst-case lower bound of Benchasattabuse *et al.*, who show that any QAOA employing a Grover-style permutation-invariant mixer needs

$$p = \Omega(n^{\gamma})$$

layers, for some constant $\gamma > 0$, to guarantee a fixed approximation ratio on NP-hard families; their method yields only a trivial bound for the conventional transverse-field mixer [BBL^{+}25].

Several algorithmic variants aim to raise shallow-depth performance, although—apart from a few recent exceptions—none yet comes with worst-case performance guarantees.

- *Warm-Start QAOA (WS-QAOA)* prepares an initial state from a high-quality classical solution and then executes a short QAOA circuit; Egger, Gambella and Woerner report empirical gains on portfolio-optimization and Max-2-SAT, but the quantum stage inherits rather than improves the classical guarantee [EMW21].

- *ADAPT-QAOA* grows the circuit layer by layer, adding those mixer or phase-separator terms whose gradients are largest; Zhu *et al.* demonstrate faster convergence and lower gate counts on Max-Cut, yet no analytic approximation ratio is proved [ZTB+22].

- *Recursive QAOA (R-QAOA)* iteratively measures spin–spin correlations, fixes the most strongly correlated qubit pair and recurses on the reduced instance, achieving deeper-circuit quality with substantially smaller physical depth on 3-regular Max-Cut [BKKT20].

- The *Quantum Alternating Operator Ansatz* generalizes QAOA by designing problem-tailored *mixers* that preserve feasibility constraints; Hadfield *et al.* furnish the framework and give proof-of-principle numerics on Max-k-SAT, but no uniform ratio bound [HWO+19].

Two recent proposals supply partial analytic guarantees. *Phantom-QAOA* adds a single global ZZ coupling—effectively a "phantom" edge between every qubit pair—which lets each qubit interact beyond the radius set by the circuit depth; Langfitt, Tate and Eidenbenz prove an approximation ratio of 0.7925 for even-length cycle graphs at $p=1$ (versus 0.75 for standard QAOA) and observe average gains of 4% on random cubic graphs for $p=1$ and 2% for $p=2$ [LTE25].

Wang, Su and Li present a one-round 'light-cone variational algorithm' that employs a locality-expanded cost operator; they establish an approximation ratio of 0.7926 on 3-regular graphs and raise it to 0.8333 via angle relaxation, the highest rigorously certified

constant for any depth-1 ansatz to date [WSL25]. Small-size numerics in the same work indicate that increasing p under their modified cost operator offers only marginal additional benefit, so simply raising the depth is unlikely to overtake the 0.8786 Goemans–Williamson or the 0.9326 Halperin–Livnat–Zwick semidefinite-program bounds.

To date, beyond the $p=1$ bound of 0.6924 3-regular graph, rigorous constant-factor guarantees for depth-p QAOA exist only in the restricted regimes described above: the analytic values 0.7559 ($p=2$) and 0.7924 ($p=3$) under girth assumptions, the 0.7925–0.8333 constants obtained with modified cost operators at $p=1$, and the cut-fraction bound 0.79 for $p=7$ on graphs with girth $g \geq 16$.

As yet no results exist that shows a fixed depth $p > 1$ securing a constant approximation ratio on arbitrary instances. The tightest available lower bound that grows in n applies only to QAOA variants that employ a Grover-style mixer, requiring $p = \Omega(n^\gamma)$ rounds to guarantee any fixed ratio on NP-hard families [BBL$^+$25]. For the conventional transverse-field mixer the question remains open: analyses have so far neither fully ruled out nor demonstrated a constant-factor guarantee at fixed $p > 1$.

Furthermore, long-standing classical work on greedy or constant-radius ("local") heuristics shows that upper-bound approximation quality is capped once an algorithm is confined to a fixed neighborhood: a single-vertex ("1-flip") local optimum never exceeds a 1/2 approximation for Max-Cut [SG76]; constant-radius algorithms obtain at most $(1-\epsilon)\log d/d$ for the maximum independent set [GS13]; and deterministic distributed algorithms require $\Omega(\log^* n)$ rounds merely to 3-color a cycle [Lin92]. Apart from the radius-one upper bound that applies equally to classical and quantum circuits [BM22a], similar limitations for fixed-depth quantum algorithms remain open. As such, whether a constant depth p can guarantee a universal approximation ratio or if some growth with problem size is inevitable remains an open question.

1.2.2 Parameter Optimization and Concentration Phenomena

A challenge for QAOA is the optimization of its $2p$ variational parameters (the rotation angles $\boldsymbol{\gamma}, \boldsymbol{\beta}$ in each layer). The choice of parameters influences performance, and poor parameter optimization renders a high-depth circuit ineffective. Recent research has focused on characterizing the parameter landscape and devising efficient search methods.

One theoretical insight is parameter concentration: for several ensembles the optimal QAOA parameters become instance-independent in the large-n limit [ARCB21, FGGZ22, BFM$^+$22]. In other words, as the problem size grows, "universal" angle schedules yield near-optimal expected performance across almost all instances of a given type. The phenomenon was first observed empirically and has since received analytical support.

For the SK model [FGGZ22] and for random D-regular MaxCut [BFM$^+$22], analytic and numerical studies show that the optimal $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ converge to fixed limits as $n \to \infty$. Moreover, applying these infinite-size parameters to finite but large instances yields objective values close to the true optimum for those instances [FGGZ22]. Farhi *et al.* demonstrated that, with probability tending to 1 as $n \to \infty$, the energy of a QAOA state with fixed angles concentrates sharply around its mean—a property sometimes called landscape independence [FGGZ22]. Angle optimization on a single representative instance (or via mean-field analysis) can therefore supply parameters that transfer to other instances, avoiding a costly per-instance search.

Recent results strengthen this picture. Sureshbabu *et al.* (2024) studied weighted MaxCut and derived parameter-setting rules [SHS$^+$24]. In the average-case setting for $p = 1$, the first local optimum near $\boldsymbol{\gamma} = \mathbf{0}$ coincides with the global optimum. For any fixed depth p, the QAOA energy landscape for a weighted instance converges to that of an unweighted instance under a simple parameter rescaling. Consequently, near-optimal angles obtained for unweighted MaxCut transfer to the weighted case via a single scaling

rule. Numerical validation showed that the fixed-parameter strategy achieved, on average, an objective value within 1.1 percentage points of the value produced by full optimization on each instance [SHS+24]. The data confirm that near-optimal performance is attainable without per-instance optimization in the large-size regime.

Beyond MaxCut, transferability appears in other tasks; similarity-based heuristics move parameters between instances with loss below 2 percentage points [GGF+23]. Such strategies reduce the classical computational overhead of running QAOA because an accurate initial guess narrows the search space.

Analytic variance bounds show that finite-depth QAOA avoids barren plateaus [ZLZ24, LTW+24]; gradients remain significant for moderate p, especially when using informed initial points. Researchers have observed structure including many local optima, symmetries, and periodicities in angle space that can be exploited. Because $e^{-i\beta H_M}$ and $e^{-i\gamma H_C}$ are 2π-periodic, the search may be restricted to $\beta_i, \gamma_i \in [0, 2\pi)$ [PBGE24]. Optimal β angles often decrease as layers progress, resembling bang-bang control that mirrors an adiabatic schedule [LLL20].

All these efforts aim to overcome the otherwise exponential cost of naive parameter tuning.

1.2.3 Ansatz Variants and Custom Mixers

QAOA research investigates alternative ansätze that employ customized mixing operators to extend the algorithm's applicability and improve performance on structured instances. The standard QAOA uses a transverse-field X mixer, which generates a uniform superposition over 2^n bit strings. The transverse-field mixer suits unconstrained problems; combinatorial tasks with hard constraints (e.g. exact-cover or budget-constrained portfolio selection) require feasibility preservation. In such cases it is wasteful for the quantum state to explore

infeasible configurations.

Hadfield *et al.* introduced the *Quantum Alternating Operator Ansatz*, generalizing QAOA by replacing the transverse-field mixer with problem-specific unitaries $U_M(\beta)$ that preserve feasibility [HWO+19]. In this framework one alternates a phase-separation unitary $U_C(\gamma)$ with $U_M(\beta)$, ensuring that the state remains inside the valid subspace during the evolution [HWO+19]. For problems with fixed Hamming weight (selecting exactly k items from n) a global XY mixer preserves excitation number and confines the search to the correct subspace [HWO+19]. The resulting circuit remains inside the feasible subspace; empirical studies report higher objective values than penalty-based encodings on small constrained instances.

Maximum Independent Set provides an example: a specialized mixer toggles a vertex conditioned on neighbor occupancy and thereby explores the independent-set space efficiently. Such mixers improve success probabilities on small graphs and prevent the infeasible-region trapping that affects standard QAOA.

The framework permits formulation of nearly any constrained combinatorial problem as a QAOA-like circuit [HWO+19]. With suitable mixers one can reproduce known quantum algorithms. A Grover-search mixer that selectively rotates marked states realizes unstructured search within the alternating-operator paradigm [HWO+19]. Bärtschi and Eidenbenz proposed Grover-Mixer QAOA (GM-QAOA), providing a route to quadratic speed-ups for unstructured search [BE20]. Because its operations are gate-exact rather than trotterized, GM-QAOA avoids certain errors and yields equal-amplitude superpositions in the feasible subspace [BE20].

Other mixers enforce degree, partitioning, or permutation constraints [HWO+19]. Each mixer or phase operator tailors QAOA to a specific structure. Greater expressiveness raises circuit complexity: non-local mixers demand additional qubits or long-range gates. A

complete-graph XY mixer samples the Hamming-weight-k manifold more ergodically than a ring mixer but needs all-to-all connectivity. Cook *et al.* report higher mean objective values and an exponential contraction of outcome variance with depth when using the complete-graph mixer [CBE20]. Angle schedules that maximize performance align with a discretized adiabatic trajectory [CBE20].

Recursive QAOA runs a shallow circuit, fixes variables, and recurses; depth-1 RQAOA solves MaxCut exactly on complete graphs [BL24]. Warm-start QAOA initializes from a classical relaxation solution and inherits its guarantee, boosting shallow-depth performance [EMW21]. In summary, customized mixers enlarge the set of problems QAOA addresses and can reduce the depth needed for a target approximation, while leaving an open trade-off between per-round gate cost and depth.

Chapter 2

Background

In this chapter we fix notation and collect results used in through the rest of the dissertation. Content includes graph and hypergraph preliminaries, Catalan numbers and Dyck paths, formal definitions of Max-k-SAT and Max-Cut, elements of quantum computation and Hamiltonian dynamics, the QAOA framework and its locality, cost-Hamiltonian mappings for Max-Cut and SAT, the Goemans–Williamson approximation, and hardness via satisfiability and algorithmic thresholds and the Overlap Gap Property.

2.1 Math Preliminaries

2.1.1 Graphs

A **graph** $G = (V, E)$ consists of a set of vertices V and a set of edges E, where each edge connects two vertices in V. A **hypergraph** $H = (V, E)$ is a generalization of a graph where V is a set of vertices, and E is a set of non-empty subsets of V, called hyperedges. Unlike graphs, a hypergraph permits hyperedges that connect any number of vertices. A **k-uniform hypergraph** is a hypergraph where every hyperedge connects exactly k vertices,

i.e., for all $e \in E$, $|e| = k$.

2.1.1.1 Diameter of a Hypergraph

The **distance** between two vertices $u, v \in V$ in a hypergraph H is defined by the minimum number of hyperedges that must be traversed to get from u to v. More formally, a **path** in a hypergraph is a sequence of vertices $(v_0, v_1, \ldots, v_l)$ such that for each i, there exists a hyperedge $e_i \in E$ containing both v_{i-1} and v_i.

The **diameter** of a hypergraph H, denoted $\mathrm{diam}(H)$, is the maximum distance between any pair of vertices in H. That is:

$$\mathrm{diam}(H) = \max_{u,v \in V} d(u, v), \tag{2.1}$$

where $d(u, v)$ is the distance between u and v.

2.1.2 Catalan Numbers and Dyck Paths

The Catalan numbers C_n are a fundamental integer sequence in combinatorics, defined for $n \geq 0$ by the closed formula

$$C_n = \frac{1}{n+1}\binom{2n}{n}, \tag{2.2}$$

with the first few values $C_0 = 1, C_1 = 1, C_2 = 2, C_3 = 5, C_4 = 14, \ldots$. Equivalently, one sets $C_0 = 1$ and for $n \geq 0$ imposes the recurrence

$$C_{n+1} = \sum_{i=0}^{n} C_i \, C_{n-i} \tag{2.3}$$

with $C_0 = 1$. This convolution-type recurrence mirrors the canonical decomposition of Catalan structures.

Introducing the ordinary generating function

$$C(x) = \sum_{n \geq 0} C_n \, x^n, \tag{2.4}$$

the recurrence translates into the quadratic equation $C(x) = 1 + x\,C(x)^2$, whose unique solution (with $C(0) = 1$) is

$$C(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

Expanding this by the binomial theorem recovers the closed form in Eq. (2.2) [Ric11, FS09]. Thus the Catalan numbers grow roughly like $4^n/(n^{3/2}\sqrt{\pi})$ asymptotically, but exactly satisfy the closed form.

2.1.2.1 Dyck Paths

A Dyck path of semilength n is a lattice path from $(0, 0)$ to $(2n, 0)$ taking up-steps $U = (1, 1)$ and down-steps $D = (1, -1)$ and never passing below the x-axis [Ric11]. Equivalently, Dyck paths are in bijection with balanced sequences of n up-steps and n down-steps in which every prefix has at least as many U's as D's. A classical result is that the number of Dyck paths of semilength n equals the nth Catalan number C_n [Ric11].

2.1.2.2 Combinatorial Interpretations

Catalan numbers enumerate a wide variety of combinatorial structures. Notable examples include:

- **Balanced Parentheses (Dyck words)**: The number of well-formed strings of n pairs of parentheses is C_n.

- **Full Binary Trees**: The number of plane binary trees with n internal nodes (and $n+1$ leaves) is C_n.
- **Polygon Triangulations**: The number of ways to triangulate a convex polygon with $n+2$ vertices is C_n.
- **Monotonic Lattice Paths**: The number of lattice paths from $(0,0)$ to (n,n) that never cross above the diagonal $y=x$ is C_n.

2.1.2.3 Bijections between Catalan Structures and Dyck Paths

Many Catalan objects can be encoded by Dyck paths, explaining why the Catalan numbers recur. For example, any balanced parentheses string (Dyck word) corresponds to a Dyck path by reading each "(" as U and each ")" as D. Likewise, a plane binary tree with n internal nodes corresponds to a Dyck path of semilength n via its balanced-parentheses (contour) encoding, mapping opens to U and closes to D [FS09]. Other examples include triangulations and noncrossing partitions. The recursive decomposition of the Dyck path mirrors that of the Catalan structure, underpinning the convolution recurrence defined above.

2.2 Classical Problems

This section defines the two classical problems used throughout: Max-k-SAT and Max-Cut, together with notation for variables, clauses, and cut value.

2.2.1 Max-k-SAT

A random instance of a k-SAT problem is characterized by a 3-tuple (n,k,m). Here n is the number of binary variables $\{x_i\}_{i=1}^{n}$ and m is the number of clauses w_i in the formula

$\phi_k(n, m)$. An individual clause is a disjunction of k literals, where a literal is either a variable x_i or its negation $\bar{x}_i$. A k-SAT clause w_i is defined by:

$$w_i = \bigvee_{j=1}^{k} l_j \tag{2.5}$$

where each literal l_j is either a variable x_i or its negation $\bar{x}_i$.

Random k-SAT instances are generated by sampling m clauses from the non-contradictory subset of the $\binom{2n}{k}$ possible clauses. The $2n$ in the binomial coefficient represents the total number of possible literals, accounting for both the variables and their negations. Contradictory clauses—those that include both a variable and its negation—are excluded from this set. The m clauses are sampled uniformly and without replacement from this subset [Ach09]. The k-SAT formula ϕ is the conjunction of these m clauses:

$$\phi = \bigwedge_{i=1}^{m} w_i \tag{2.6}$$

where each w_i is a k-literal clause.

The k-SAT problem is a decision problem that determines whether there exists an assignment $\mathbf{x} \in \{0, 1\}^n$ that satisfies all m clauses in the formula ϕ. It checks if there exists a binary vector assignment $\mathbf{x} \in \{0, 1\}^n$ such that:

$$\phi(\mathbf{x}) = 1 \tag{2.7}$$

Max-k-SAT is the associated optimization problem that seeks to find the assignment $\mathbf{x}$ of truth values to the n variables that maximizes the number of satisfied clauses in the

formula ϕ. A solution to Max-k-SAT maximizes the following objective function:

$$C(\mathbf{x}) = \sum_{i=1}^{m} w_i(\mathbf{x}), \tag{2.8}$$

where $w_i(\mathbf{x}) = 1$ if the i-th clause is satisfied by the assignment $\mathbf{x}$, and 0 otherwise. The goal is to find:

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in \{0,1\}^n} C(\mathbf{x}) \tag{2.9}$$

which represents the assignment that maximizes the number of satisfied clauses.

2.2.2 Max-Cut

Given a graph $G = (V, E)$ with a set of n vertices $V = \{v_1, v_2, \ldots, v_n\}$ and a set of m edges $E = \{e_1, e_2, \ldots, e_m\}$. A cut is a partitioning of V into two subsets $(S, \bar{S})$, where $S \cup \bar{S} = V$ and $S \cap \bar{S} = \emptyset$. The size of the cut is determined by the number of edges spanning across these two partitions:

$$C_{P_1,P_2} = \sum_{i \in P_1, j \in P_2, (ij) \in E} w_{ij} \tag{2.10}$$

where $w_{ij} = 1$ when $(ij) \in E$ and 0 otherwise. The objective of the Max-Cut problem is to find a cut that maximizes this sum, effectively maximizing the number of edges between P_1 and P_2. Solutions can be represented using a bit string $x \in \{0,1\}^{|V|}$, where $x_i = 1$ indicates $i \in S$ and $x_i = 0$ indicates $i \in \bar{S}$ [Kar72].

We can formulate the Max-Cut problem as an optimization problem with the objective function [GW95]:

$$C(x) = \sum_{(i,j) \in E} \frac{1}{2}(1 - (-1)^{x_i \oplus x_j}) \tag{2.11}$$

where $x_i \oplus x_j$ denotes the bitwise XOR operation between x_i and x_j. In this formulation, x is the bit string representing the partition of vertices. The $\oplus$ operation checks if two connected vertices belong to different subsets. If they are in the same subset, $x_i \oplus x_j$ equals 0, resulting in the term contributing nothing to the sum. Conversely, if they are in different subsets, $x_i \oplus x_j$ equals 1, and the term contributes to the sum and counts the edge in the cut. The factor of $\frac{1}{2}$ is used to normalize the count since each edge is considered twice in the summation. Thus, maximizing $C(x)$ corresponds to finding the cut that maximizes the number of edges between the two subsets.

2.3 Quantum Computing

Quantum computation generalizes classical computation by utilizing two-level quantum systems, called *qubits*, and replacing deterministic logic with unitary evolution governed by quantum mechanics [Deu85, NC11].

2.3.1 Qubits and Quantum States

A *qubit* is a normalized vector $|\psi\rangle$ in the two-dimensional complex Hilbert space $\mathcal{H}_2 \cong \mathbb{C}^2$. Selecting an orthonormal computational basis $\{|0\rangle, |1\rangle\}$, any single-qubit pure state admits the expansion

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1.$$

An n-qubit system (or register) inhabits the tensor product Hilbert space $\mathcal{H}_2^{\otimes n} \cong \mathbb{C}^{2^n}$. Basis states are labeled by bitstrings $x \in \{0,1\}^n$, denoted

$$|x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle\,, \quad x_j \in \{0,1\}.$$

A general pure state $|\Psi\rangle \in \mathcal{H}_2^{\otimes n}$ is thus

$$|\Psi\rangle = \sum_{x\in\{0,1\}^n} \psi_x |x\rangle, \quad \psi_x \in \mathbb{C}, \quad \sum_x |\psi_x|^2 = 1.$$

The exponential growth of the Hilbert space dimension with n underlies quantum computational power and complexity. For example, a 50-qubit system's state vector has $2^{50} \approx 1.12 \times 10^{15}$ complex components, far too many to store or process on existing classical hardware [NC11].

Non-factorizable multi-qubit states exhibit *entanglement*. Entanglement is a uniquely quantum form of correlation with no classical analog: for instance, two qubits can be prepared in the entangled Bell state

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}},$$

Here neither qubit has a definite value on its own, yet if one is measured the other is found in a perfectly correlated state [NC11, JL03].

Entangled states cannot be factored into independent subsystems and have stronger-than-classical correlations that are essential to quantum algorithms and protocols. Indeed, many quantum algorithms rely on creating entanglement among qubits to achieve a computational speedup [Joz98]. At the same time, the no-cloning theorem prohibits making independent copies of an unknown quantum state [WZ82], and measurement disturbs the state, so extracting the advantages of superposition and entanglement requires careful algorithmic design.

2.3.2 Quantum Gates and Circuits

To perform computation, qubits are manipulated by applying sequences of quantum logic gates. A *quantum gate* is a reversible evolution of the quantum state that corresponds to a unitary operator $U : \mathcal{H}_2^{\otimes n} \to \mathcal{H}_2^{\otimes n}$ satisfying $U^\dagger U = I$, which preserves the norm (total probability).

In the simplest case, a single-qubit gate is a 2×2 unitary matrix that transforms the state of a single qubit. Single-qubit gates are represented by 2×2 unitary matrices, which form the group U(2). Canonical examples include the Pauli gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and the Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

which produces an equal superposition from a basis state: $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ [NC11].

Multi-qubit gates enact entangling operations. A two-qubit gate is the controlled-NOT (CNOT), which flips the state of a target qubit conditional on the control qubit being in state $|1\rangle$:

$$\mathrm{CNOT}\,|x, y\rangle = |x, y \oplus x\rangle\,, \quad x, y \in \{0, 1\}.$$

While an n-qubit unitary is specified by an exponentially large number of parameters, it can be approximated by a sequence of gates from a small, fixed set. A set of gates is called *universal* if it can be used to construct a circuit that approximates any unitary operation to arbitrary precision. A common universal set consists of all single-qubit gates combined

with any two-qubit entangling gate, such as the CNOT gate [BBC$^+$95]. For practical implementation, discrete universal sets are often used; an example is $\{H, T, \mathrm{CNOT}\}$, where $T = \mathrm{diag}(1, e^{i\pi/4})$ is the $\pi/8$ phase gate. The Solovay–Kitaev theorem guarantees that any arbitrary unitary can be approximated to a precision ε with a sequence of gates from such a set whose length grows only polylogarithmically with $1/\varepsilon$ [DN05].

A quantum algorithm in the circuit model is described by a sequence of such gate operations acting on an initial state (usually a simple product state like $|0\rangle^{\otimes n}$) and followed by a measurement of one or more qubits at the end [NC11]. A *quantum circuit* is an ordered sequence of gates acting on designated qubit lines, often depicted as a diagram where time progresses from left to right. The circuit's *depth* is the number of gate layers that cannot be executed in parallel, its *width* is the number of qubits, and its *size* is the total number of gates. For an n-qubit circuit with depth d and at most k-local gates, the unitary realized is $U = U_d \cdots U_2 U_1$, where each U_j factorizes into at most $\lceil n/k \rceil$ commuting blocks [ABO96]. Complexity measures such as T-count refine the cost model by isolating fault-tolerant overhead [AMM14].

2.3.3 Quantum Measurement

The exponential state space of an n-qubit system is not directly accessible, however, because any measurement yields only an n-bit outcome. The potential computational advantage arises not from storing an exponential amount of classical information per se, but from processing information in a high-dimensional quantum space and using interference to bias measurement outcomes toward desired results.

Upon measuring a single qubit in state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ in the $\{|0\rangle, |1\rangle\}$ computational basis, the outcome is 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$ [NC11]. Unlike a probabilistic bit (which is in either state but unknown), prior to measurement the

qubit is *not* in a definite classical state but rather in a quantum superposition of both.

Measurement extracts classical information through a set of measurement operators $\{M_m\}$ satisfying the completeness relation $\sum_m M_m^\dagger M_m = I$. The standard computational-basis measurement applies projectors $M_x = |x\rangle\langle x|$ and outputs the bit string x with probability given by the Born rule:

$$\Pr(x) = |\langle x|\psi\rangle|^2 = \langle\psi|M_x^\dagger M_x|\psi\rangle\,.$$

Following the measurement yielding outcome x, the state of the system collapses:

$$|\psi\rangle \to \frac{M_x\,|\psi\rangle}{\sqrt{\Pr(x)}}.$$

This measurement process constitutes the interface between quantum and classical processing [NC11].

2.3.4 Hamiltonian Dynamics and Continuous-Time Computing

The circuit model described above is often called the *digital* model of quantum computing. An alternate but computationally equivalent model is the *analog* or Hamiltonian model, where computation is implemented through continuous-time evolution of a quantum state under a carefully designed Hamiltonian [AvDK+04].

Quantum state evolution for a closed system follows the Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle\,, \quad |\psi(0)\rangle = |\psi_0\rangle\,,$$

where the Hamiltonian $H(t)$ is a Hermitian operator. The formal solution is $|\Psi(t)\rangle = U(t)\,|\Psi(0)\rangle$ with $U(t) = \mathcal{T}\exp\big(-\frac{i}{\hbar}\int_0^t H(t')\,dt'\big)$ being the time-ordered exponential of the

Hamiltonian. In the special case of a time-independent Hamiltonian H, this reduces to $U(t) = e^{-iHt/\hbar}$.

Quantum gates in a circuit can thus be viewed as discrete fragments of Hamiltonian evolution. Any gate U is in principle realizable by having some Hamiltonian H act for a fixed time [NC11]. Conversely, a time-dependent Hamiltonian evolution can be approximated by a sequence of short gate intervals, a process known as Trotterization.

2.3.5 Tangent Space of Unitary Operators

Time evolution $U = e^{-iHt}$ is the exponential map that describes the relation between the $U \in U(2^n)$ Lie group of unitary operators and the $H' \in \mathfrak{u}(2^n)$ Lie algebra of skew-Hermitian ($H'_j = -iH_j$) generators. The identity element of $U(2^n)$ corresponds to the $t = 0$ case of the exponential map:

$$U(0) = e^{-iH\cdot 0} = e^0 = I \tag{2.12}$$

We can expand U's exponential map using the Taylor series of the exponential function:

$$U = e^{-iHt} = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!} \tag{2.13}$$

Taking the derivative with respect to t, we have:

$$\frac{d}{dt} e^{-iHt} = \sum_{n=1}^{\infty} \frac{d}{dt} \left(\frac{(-iHt)^n}{n!} \right) = \sum_{n=1}^{\infty} \frac{n(-iH)^n \cdot t^{n-1}}{n!} \tag{2.14}$$

Finally, at $t = 0$ the expression simplifies:

$$\left. \frac{dU}{dt} \right|_{t=0} = \sum_{n=1}^{\infty} \frac{n(-iH)^n \cdot 0^{n-1}}{n!} = \frac{1(-iH)^1 \cdot 0^0}{1!} = -iH \tag{2.15}$$

This relation connects the tangent space of the group with the Hermitian generators, representing the infinitesimal action of the group on the tangent space at the identity. Specifically, the tangent space $T_I U(2^n)$ at the identity element of the group $U(2^n)$ is the set of all skew-Hermitian operators in $\mathfrak{u}(2^n)$.

2.4 Quantum Optimization

Quantum optimization encodes a discrete objective as a Hamiltonian and drives the state so that measurement concentrates on high-value configurations. This section fixes QAOA notation, relates depth to locality, and the Max-Cut and SAT encodings.

2.4.1 Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm for approximating optimization problems [FGG14]. For a general optimization problem P with a classical objective function $f_P(x) : \{0,1\}^n \to \mathbb{R}$, generating a corresponding QAOA instance requires encoding $f_P(x)$. To be suitable for encoding, the objective function $f_P(x)$ must have a discrete configuration domain, where each of the N configurations is specified by n bits. The QAOA encodes f_P into a Hamiltonian C_P such that $C_P |x_i\rangle = f_P(x_i) |x_i\rangle$.

For SAT problems, the objective function is the sum of the m clauses:

$$f_P(\mathbf{x}) = \sum_{i=1}^{m} w_i(\mathbf{x}), \tag{2.16}$$

where $w_i(\mathbf{x}) = 1$ if $\mathbf{x}$ is a satisfying assignment for clause w_i. A detailed review of the encoding of the SAT objective function is provided in Section 2.4.5. For a general

optimization problem, C_P is an encoded sum of the constraints in the objective function:

$$C_P = \sum_{a=1}^{m} C_a \tag{2.17}$$

The QAOA also requires a mixing Hamiltonian B_P that governs transitions between candidate solutions. The standard mixing Hamiltonian is the transverse field:

$$B = \sum_{j=1}^{n} \sigma_j^x \tag{2.18}$$

The QAOA ansatz is an alternating sequence of unitary operators generated by the problem Hamiltonian C_P and the mixing Hamiltonian B_P, with a uniform superposition as the initial state $|\psi\rangle$.

Each operator is parameterized by angles $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, where p denotes the number of layers of the ansatz:

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = U(\beta_p, B_P)U(\gamma_p, C_P) \cdots U(\beta_1, B_P)U(\gamma_1, C_P) \, |\psi\rangle \tag{2.19}$$

$U(\theta, H) = e^{-i\theta H}$ are the unitaries resulting from the time evolution of Hamiltonian H with parameter θ. Finally, the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are classically optimized to minimize the expectation value of the cost Hamiltonian:

$$f_{opt} = \min_{\boldsymbol{\gamma}, \boldsymbol{\beta}} \langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | C_P | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle. \tag{2.20}$$

2.4.2 Locality of the QAOA

The Max-Cut problem Hamiltonian C_P ((2.11)) is 2-local and the transverse-field mixing Hamiltonian B_P ((2.18)) is 1-local. However, since the depth of the QAOA ansatz is linear

in p further analysis is required. In the Heisenberg picture (operators evolve while the state is fixed), the expectation value of the QAOA Hamiltonian can be broken down into a sum of terms corresponding to individual edges, with each term calculated as an expectation value over a certain subgraph [FGG14]. The locality of QAOA can be analyzed by examining the propagation of Pauli operators during the evolution of the system under the action of the mixing Hamiltonian and the problem Hamiltonian.

Making explicit the expectation value calculation for a single edge of the Max-Cut QAOA at $p = 1$:

$$E = \frac{1}{2} - \frac{1}{2} \langle + | U(\gamma_1, C_P)^\dagger U(\beta_1, B_P)^\dagger \sigma_i^z \sigma_j^z \cdots U(\beta_1, B_P) U(\gamma_1, C_P) | + \rangle \quad (2.21)$$

The application of $U(\beta_1, B_P)$ of to $\sigma_i^z \sigma_j^z$ yields:

$$E = \frac{1}{2} - \frac{1}{2} \langle + | U(\gamma_1, C_P)^\dagger (\cos 2\beta_1 \sigma_i^z + \sin 2\beta_1 \sigma_i^y) \cdots (\cos 2\beta_1 \sigma_j^z + \sin 2\beta_1 \sigma_j^y) U(\gamma_1, C_P) | + \rangle \quad (2.22)$$

All the $\sigma_k^z \sigma_l^z$ terms in $U(\gamma_1, C_P)$ cancel unless (k, i), (k, j), (l, i), or (l, j) are elements of the edge set of the source graph. In this case where an edge exists, terms that have support spanning 3 sites like $\sigma_l^z \sigma_i^y \sigma_j^y$ are formed.

For a fixed p and graph degree ν, the operator will have support over a subgraph with vertices at most p edges away from the initial vertices. This observation indicates that the QAOA's locality is determined by the depth p. For fixed graph degree ν and particular value of p, the locality can be captured by truncating the graph to an induced subgraph that only includes vertices which are at most p edges away from the initial vertex pair. The

presence of the other vertices does not contribute to the expectation value. [FGG14, WL21].

2.4.3 Sparse Operators

Like locality, sparsity is a way of characterizing the structure of a Hamiltonian, but instead focusing on the maximum number of nonzero entries in any row or column of the Hamiltonian matrix. A d-sparse Hamiltonian is one in which each row or column has at most d nonzero entries. This notion of sparsity is distinct from the concept of locality in local operators, as not all sparse Hamiltonians are local, and many sparse Hamiltonians do not admit a Pauli decomposition with a polynomial number of terms. In the computational basis, every k-local spin Hamiltonian is sparse (each row has $d = O(n^k)$ nonzeros).

The study of sparse operators and their applications to quantum simulation is an active area of research[ATS03, BCC$^+$14]. For example, the Variational Quantum Eigensolver (VQE) has been deployed with sparse Hamiltonians [KL21]. This field has seen significant development over the years, resulting in optimal sparse operator simulation algorithms[LC19, BCS$^+$20].

2.4.4 Max-Cut Hamiltonian

To run the QAOA on Max-k-SAT (or any optimization problem), it is necessary to convert a corresponding classical cost function into a quantum Hamiltonian. The Max-Cut cost function 2.11 can be further reduced if the bit string x_i is mapped to spin s_i over $\{-1, 1\}$ [Luc14]:

$$C(\boldsymbol{x}) = \frac{1}{2} \sum_{(ij)\in E} (1 - s_i s_j) \tag{2.23}$$

Which can be directly mapped as the QAOA cost Hamiltonian H_C for the Max-Cut problem [FGG14]:

$$H_C = \sum_{(i,j)\in E} \frac{1}{2}(I - \sigma_i^z \sigma_j^z), \tag{2.24}$$

where σ_i^z and σ_j^z are Pauli-Z matrices acting on the qubits representing vertices i and j, respectively.

2.4.5 SAT Cost Hamiltonian

To apply QAOA to Max-k-SAT, we first convert the clause-count objective to a Hamiltonian. As a conversion convenience, we formalize this as a minimization. The cost function C_w for each single clause w in a SAT formula ϕ is:

$$C_w = \left(\prod_{x\in w} f_w(x) \right) - 1, \tag{2.25}$$

where $f_w(x)$ scores if each literal x or its negation $\bar{x}$ in the clause w is satisfied. Specifically:

$$f_w(x) = \begin{cases} 1 - x & \text{if } x \text{ is present in } w, \\ x & \text{if } \bar{x} \text{ is present in } w, \\ 1 & \text{otherwise.} \end{cases} \tag{2.26}$$

Note that, for convenience of conversion to a Hamiltonian, Equation 2.25 is formulated as a minimization problem. The total cost of a formula ϕ is the sum of its constituent clauses:

$$C_\phi = \sum_{w\in\phi} C_w \tag{2.27}$$

The cost Hamiltonian associated with a formula ϕ converts each binary variable x_i in ϕ (where $x_i \in \{0, 1\}$) to a spin variable $s_i \in \{-1, 1\}$ using the transformation $x_i = \frac{1-s_i}{2}$. This

transformation allows the polynomial expansion of the cost term in Equation 2.25.

For instance, consider the clause $x_i \vee x_j \vee \bar{x}_k$:

$$C_{w_i} = (1 - x_i)(1 - x_j)x_k$$

Substituting $x_i = \frac{1-s_i}{2}$, $x_j = \frac{1-s_j}{2}$, and $x_k = \frac{1-s_k}{2}$, we get:

$$C_{w_i} = \left(1 - \frac{1 - s_i}{2}\right)\left(1 - \frac{1 - s_j}{2}\right)\left(\frac{1 - s_k}{2}\right)$$

Expanding the product:

$$C_{w_i} = \frac{1}{8}(1 + s_i)(1 + s_j)(1 - s_k) = \frac{1}{8}\left(1 + s_i + s_j - s_k + s_i s_j - s_i s_k - s_j s_k - s_i s_j s_k\right) \quad (2.28)$$

The total Hamiltonian for the formula ϕ is then expressed as:

$$H_\phi = \sum_{w \in \phi} C_w(s) \quad (2.29)$$

where each $C_w(s)$ term is correspondingly derived by substituting the spin variables into the classical cost function, transforming it into a quantum Hamiltonian suitable for QAOA optimization.

2.5 Approximation

Approximation algorithms provide classical baselines for combinatorial optimization. For Max-Cut the Goemans–Williamson semidefinite relaxation with random-hyperplane rounding yields an expected ratio $\alpha \approx 0.87856$, which serves as the benchmark for comparisons in later chapters.

2.5.1 Goemans–Williamson Algorithm for Max-Cut

2.5.1.1 Problem Formulation and Semidefinite Relaxation

Given an undirected graph $G = (V, E)$ with nonnegative edge weights w_{ij}, Max-Cut asks for a bipartition of V into two sets $(S, \bar{S})$ that maximizes the total weight of edges crossing between S and $\bar{S}$. Equivalently, assign each vertex $i \in V$ a binary variable $x_i \in \{\pm 1\}$, where $x_i = +1$ indicates $i \in S$ and $x_i = -1$ indicates $i \in \bar{S}$. We adjust the Max-Cut objective for weighted edges as

$$\max_{x_i \in \{\pm 1\}} \frac{1}{2} \sum_{(i,j) \in E} w_{ij}\big(1 - x_i x_j\big),$$

where the term $(1 - x_i x_j)/2$ equals 1 if $x_i \neq x_j$ and 0 otherwise. Goemans and Williamson introduced a semidefinite programming relaxation that replaces each scalar x_i by a unit vector $v_i \in \mathbb{R}^n$, so that $v_i \cdot v_j$ serves as a continuous analogue of $x_i x_j$ [GW95]. Specifically, one solves

$$\max_{\substack{v_i \in \mathbb{R}^n \\ \|v_i\| = 1}} \quad \frac{1}{2} \sum_{(i,j) \in E} w_{ij}\big(1 - v_i \cdot v_j\big),$$

$$\text{subject to} \quad X_{ij} = v_i \cdot v_j, \quad X \succeq 0, \quad X_{ii} = 1 \quad \forall i \in V,$$

where X is the Gram matrix with entries $X_{ij} = v_i \cdot v_j$. The constraint $X \succeq 0$ enforces positive semidefiniteness, and $X_{ii} = 1$ ensures unit norm. Since any $\{\pm 1\}$ assignment yields $X_{ij} = x_i x_j$, this semidefinite program (SDP) is a relaxation of the original integer quadratic program.

2.5.1.2 Random Hyperplane Rounding Procedure

Let $\{v_i^*\}$ be an optimal solution of the semidefinite program (SDP). To obtain a feasible cut, sample a random vector $r \in \mathbb{R}^n$ uniformly from the unit sphere (e.g. each coordinate

$r_k \sim \mathcal{N}(0,1)$ independently). Define

$$x_i = \operatorname{sign}\big(v_i^* \cdot r\big) \in \{\pm 1\} \quad \text{and} \quad S = \{\, i : x_i = +1\}, \quad \bar{S} = V \setminus S.$$

This *random hyperplane* rounding ensures $x_i \in \{\pm 1\}$, so $(S, \bar{S})$ is a valid cut. For any edge $(i,j) \in E$, let $\theta_{ij} \in [0, \pi]$ be the angle between v_i^* and v_j^*, satisfying $v_i^* \cdot v_j^* = \cos\theta_{ij}$. The probability that the random hyperplane separates i and j is

$$\mathbb{P}[x_i \neq x_j] = \frac{\theta_{ij}}{\pi},$$

since a random line through the origin in the plane spanned by v_i^*, v_j^* separates the vectors precisely if it falls within their angular separation.

2.5.1.3 Approximation Guarantee and Expected Value Analysis

Let $W_{\text{cut}}(S) = \sum_{(i,j)\in E : x_i \neq x_j} w_{ij}$ denote the weight of the cut produced. By linearity of expectation,

$$\mathbb{E}[W_{\text{cut}}] = \sum_{(i,j)\in E} w_{ij}\, \mathbb{P}[x_i \neq x_j] = \sum_{(i,j)\in E} w_{ij}\, \frac{\theta_{ij}}{\pi}.$$

The SDP objective equals

$$\text{SDP}(G) = \frac{1}{2} \sum_{(i,j)\in E} w_{ij}\big(1 - \cos\theta_{ij}\big),$$

and satisfies $\text{SDP}(G) \geq \text{OPT}(G)$, where $\text{OPT}(G)$ is the true Max-Cut optimum. Consider the per-edge ratio

$$\frac{w_{ij}\, \dfrac{\theta_{ij}}{\pi}}{\frac{1}{2}\, w_{ij}\big(1 - \cos\theta_{ij}\big)} = \frac{2\theta_{ij}}{\pi\big(1 - \cos\theta_{ij}\big)}\,.$$

Define

$$f(\theta) = \frac{2\theta}{\pi\big(1-\cos\theta\big)}, \quad \theta \in [0,\pi].$$

The approximation ratio of the algorithm is

$$\alpha = \min_{0\le\theta\le\pi} f(\theta) \approx 0.87856.$$

It follows that for each edge,

$$\frac{\theta_{ij}}{\pi} \geq \alpha\,\frac{1-\cos\theta_{ij}}{2},$$

and summing yields

$$\mathbb{E}[W_{\text{cut}}] \geq \alpha\,\text{SDP}(G) \geq \alpha\,\text{OPT}(G).$$

Hence the algorithm achieves an expected approximation factor of $\alpha \approx 0.87856$. Repeating the rounding and selecting the best cut amplifies success probability to arbitrarily close to one.

2.6 Hardness

Hardness results calibrate achievable performance on random instances. Clause density controls satisfiability and algorithmic thresholds for random k-SAT, and the Overlap Gap Property formalizes geometric obstructions to local algorithms.

2.6.1 Satisfiability Threshold

The clause-to-variable density $\alpha = \frac{m}{n}$ of a random k-SAT formula parametrizes different regimes of hardness and satisfiability. For fixed n, fewer constraints yield more satisfiable formulas, while more constraints limit satisfying assignments and can render a formula unsatisfiable. Early numerical work by Mitchell, Selman, and Levesque [MSL92] demonstrated

that when ϕ is sampled uniformly at random, the median hardness of such instances is characterized by the clause-to-variable ratio α. Specifically, hardness, as measured by the number of iterations of the Davis-Putnam algorithm [DP60], peaks near the α at which sampled formulas transition from being likely satisfiable to likely unsatisfiable. The density at which transition occurs is the satisfiability critical point; a term borrowed from statistical physics.

This critical point has been extensively researched [Ach09, FB99, DB97, AP03, KKKS98] and it is widely believed that for every $k \geq 3$, there exists a constant α_k such that:

$$\lim_{n\to\infty} \Pr[\phi_k(n, \alpha n) \text{ is satisfiable}] = \begin{cases} 1, & \text{if } \alpha < \alpha_k \\ 0, & \text{if } \alpha > \alpha_k \end{cases} \tag{2.30}$$

While a complete proof of the satisfiability threshold conjecture remains elusive for general k, it is known that as $n \to \infty$, a random k-SAT formula is with high probability (w.h.p.) unsatisfiable for α greater than [MMZ05a, KKKS98]:

$$\alpha_{k,\text{crit}} = 2^k \ln 2 - \frac{1 + \ln 2}{2} + \epsilon_k \tag{2.31}$$

where ϵ_k tends to zero as $k \to \infty$ and the location of the threshold becomes more precise, converging to:

$$\alpha^*_{k,\text{crit}} = 2^k \ln 2 - \frac{1 + \ln 2}{2} \tag{2.32}$$

Characterizations of the limiting term have been further refined by Coja-Oghlan and Panagiotou [COP16], with recent work Ding, Sly, and Sun [DSS21] precisely identifying the ϵ_k for large k.

2.6.2 Algorithmic Threshold

We can also define an algorithmic threshold at which the expected hardness of solving Max-k-SAT for a random formula shifts from polynomial to exponential time. Similar to the satisfiability threshold, the algorithmic threshold $\alpha_{k,\text{alg}}$ is a critical clause density for random k-SAT formulas:

$$\lim_{n\to\infty} \Pr[\phi_k(n, \alpha n) \in P] = \begin{cases} 1, & \text{if } \alpha < \alpha_{k,\text{alg}} \\ 0, & \text{if } \alpha > \alpha_{k,\text{alg}} \end{cases} \tag{2.33}$$

where P is the polynomial-time complexity class.

As we are dealing with algorithmic hardness, one natural method for finding $\alpha_{k,\text{alg}}$ is to bound the performance of actual algorithms. As the algorithmic threshold occurs before the satisfiability threshold, research into characterizing $\alpha_{k,\text{alg}}$ focuses on the decision version of random k-SAT. Coja-Oghlan [CO10] presented the Fix algorithm, which w.h.p is able to solve random k-SAT instances with a clause density of less than:

$$(1 - \epsilon_k)2^k \frac{\log(k)}{k} \tag{2.34}$$

Fix improved on the prior best clause density bound of $1.817 \cdot \frac{2^k}{k}$ [FS96] and remains the best-known upper bound derived from the performance of an algorithm.

Shifting away from algorithmically realized bounds, Bresler and Huang [BH22] leverage the OGP, defined below in 2.6.4, to show hardness for α_k above:

$$(1 - \epsilon_k)2^k \frac{\log(k)}{k}\kappa \tag{2.35}$$

The Bresler and Huang bound is a constant factor $\kappa = 4.911$ higher than the Coja-Oghlan

upper bound. This difference κ between the upper bound 2.34 achieved by the best-performing polynomial algorithm and the tightest rigorous lower bound 2.35 for algorithmic hardness is known as the statistical-to-computational gap.

2.6.3 Overlap Gap Property

The study of algorithmic hardness as a function of solution geometry, or optimization landscapes, initially identified solution clustering beyond some threshold, or phase transition, for random instances of K-SAT [ACO08]. These clusters were defined in terms of pairwise hamming distance over bit-strings $x_1, x_2 \in \{0,1\}^n$:

$$d(x_1, x_2) = \sum_{i}^{n} |(x_1)_i - (x_2)_i| \tag{2.36}$$

All candidate solutions in a cluster are at most d apart from one another in hamming distance. Each cluster contains at most $e^{-\gamma n}$ solutions, where γ is some fraction greater than 0. The clusters are at least ζn apart from one another. Finally, the threshold is in terms of candidate solution quality; solutions above the threshold are closer to the optimal value. In short, there exists small pockets of solutions similar in both quality and structure that are larger that d distances apart. More importantly, the threshold at which this structure occurs is just above the largest concentration of solutions produced by the best polynomial-time algorithms.

This initial observation spurred a line of work that linked both quantum and classical algorithmic hardness to what would eventually be known as the Overlap Gap Property [GS13, ARNSS16, CGPR19, SW20, FGG20, CLSS22a, BGMZ22]. Here, the term overlap is borrowed from the physics literature, where an overlap of two spin vectors $s_1, s_2 \in \{-1,1\}^n$

is:

$$R(s_1, s_2) = \langle s_1, s_2 \rangle = \sum_{i}^{n} (s_1)_i (s_2)_i \tag{2.37}$$

Note that the overlap of two spin vectors $R(s_1, s_2) = n - d(x_1, x_2)$ is directly related to the hamming distance of the corresponding bit strings, and the OGP can be defined interchangeably between the two.

If we consider a general optimization problem with objective P where:

1. $dom(P) = C$
2. $|C| = N = 2^n$
3. $F_{\max} = \max_{c \in \mathcal{C}} F(c)$
4. $\alpha_P = \alpha F_{\max}$
5. $C_\alpha = \{c \in C : P(c) \geq \alpha_P\}$

There exists an OGP, if there are two bounds on distance $0 < \beta < \delta$ such that:

$$\forall c_i, c_j \in C_a : d(c_i, c_j) < \beta \vee d(c_i, c_j) > \delta \tag{2.38}$$

with high probability. Equivalently, there exists an OGP if there are bounds $0 < \overline{\delta} < \overline{\beta}$ on the overlaps (where $\overline{x} = n - x$) such that:

$$\forall c_i, c_j \in C_a : R(c_i, c_j) > \overline{\beta} \vee R(c_i, c_j) < \overline{\delta} \tag{2.39}$$

Both definitions imply the existence of a region (β, δ) in distance where solutions in C_α cannot exist. As such, the OGP is more than a topological property of the solution space, it can obstruct algorithms that only process problem instances in fixed chunks as they optimize. [GS13].

Recent research has demonstrated that the OGP is present in various optimization problems, such as Max-Cut on $K \geq 4$ hypergraphs [CGPR19], spin glass models [JMSS23], and other problems on random graphs[CLSS22a]. This has led to an increased interest in understanding the barriers that the OGP imposes optimization algorithms [Gam21].

2.6.4 Algorithmic Hardness and the Overlap Gap Property

Regardless of its precise location, the question of what causes hardness beyond the algorithmic threshold has been widely considered. Recently, taking cues from statistical physics, a form of topological clustering in the solution space has been the focus of much research. Starting from a correspondence between spin glasses and SAT problems, the initially non-rigorous methods suggested that for lower clause densities, the set of satisfying assignments forms a single large cluster [MMZ05b]. However, at some critical density, this single cluster sharply separates into exponentially many much smaller disjoint clusters [KMRT$^+$07]. These clusters are randomly distributed in the total solution space, have relatively small diameters, and are "far apart" from all other clusters in terms of Hamming distance [ACO08].

This clustering phenomenon was eventually formalized by Gamarnik and Sudan [GS13] into the Overlap Gap Property (OGP) framework, which Bresler and Huang [BH22] used to prove the algorithmic threshold discussed above in Section 2.6.2. The OGP is a rigorous characterization of the disjoint solution space topology described at the beginning of this section. With it, Gamarnik and Sudan showed the non-existence of high-quality solutions at certain Hamming distances. These "forbidden regions" are defined in terms of pairwise Hamming distance:

$$d(x_1, x_2) = \sum_{i}^{n} |(x_1)_i - (x_2)_i| \tag{2.40}$$

A problem P with η-optimal solutions in the solution space C_η exhibits an OGP if the pairwise Hamming distances of all solutions in C_η are split into two distinct regions $0 < \beta < \delta$:

$$\forall c_i, c_j \in C_\eta : d(c_i, c_j) < \beta \vee d(c_i, c_j) > \delta \tag{2.41}$$

By η-optimal, we mean that given $c_{\max}$ is the optimal solution to P, then all solutions in C_η are at least $\eta c_{\max}$ for maximization.

These regions, in Hamming distance, greater than β but less than δ, where solutions in C_η cannot exist, obstruct optimization algorithms that process problem instances in fixed chunks as they optimize [GS13]. Gamarnik and Sudan used the OGP to show that local algorithms are obstructed from producing independent sets over sparse random graphs within a multiplicative factor the optimal even as the graph degree grows asymptotically.

Extensions of the OGP framework that consider the pairwise distances for η-optimal solutions over a set of instances have been used to show algorithmic hardness for NAE-k-SAT [GS17], specific and general random graph problems, and random constraint satisfaction problems [GJW22, JMSS23].

Chapter 3

Locality and SAT

Previous work using the OGP framework shows that local algorithms face a depth barrier $p = \Omega(\log n)$ on random Max-k-SAT for $k \geq 4$ and that this obstruction appears well before the satisfiability transition [CLSS22a, BH22]. These obstructions are theoretical and hold in the large-n limit. Because existing OGP-based bounds do not specify the onset approximation ratio or the gap width, empirical checks on how the presence of an OGP affects approximation quality were unavailable. Furthermore, no study has tested whether any additional barrier remains once the QAOA depth exceeds the obstruction bound. This chapter fills in those gaps: we generate random formulas with an OGP using a brute-force detector, and simulate depth-p circuits up to $p = 7$. The data show that, as soon as the circuit depth covers every variable, QAOA performs the same on OGP and non-OGP inputs, and the minimum effective depth rises roughly linearly with both variable count and clause arity.

Farhi et al. developed the QAOA with near-term gate-based quantum devices in mind [FGG14]. The first commercially accessible Noisy Intermediate-Scale Quantum (NISQ) device became available two years later [IBM21, Dev16]. Although NISQ hardware can

generate entanglement, vulnerability to uncorrectable noise-induced errors constrains circuit depth. Limits on circuit depth bound, from above, the number of gates that can sequentially be applied to a qubit or collection of qubits [Pre18]. These limitations, coupled with the availability of public NISQ devices prompted extensive investigation into the QAOA's practical and theoretical performance at both shallow-constant and super-constant ansatz depths [BGMZ22, BFM$^+$22, FGG15, FGG20, GM19, Has19, ZWC$^+$20].

Not long after it was proposed, the QAOA improved on the best-known approximation ratio for the bounded-occurrence optimization problem MAX-3-LIN-2 [FGG15]. MAX-3-LIN-2 is a maximization defined over n variables and m linear equations modulo 2, where each equation computes $(x_i + x_j + x_k) \mod 2$ of three variables. The QAOA achieved an approximation ratio of $\Omega(1/D^{3/4})$, where D is the maximum number of constraints (equations) involving any single variable x_i, surpassing prior classical algorithms for certain instances [FGG15]. Responding directly to the QAOA, the state-of-the-art classical bound was subsequently improved to $\Omega(1/D^{1/2})$ [BMO$^+$15, Has19].

Following that initial exchange, theoretical and practical research on the QAOA developed in multiple directions. Two lines are highlighted here. The first centers the Overlap Gap Property (OGP) [GS13], a pairwise clustering phenomenon in near-optimal solutions of certain optimization problems. A key consequence of the OGP clustering is the emergence of "forbidden regions" of lower-quality solutions that obstruct local algorithms, including the QAOA, from reaching the higher-quality solution clusters. Here, a local algorithm is one that processes the input in fixed-size sub-problem chunks regardless of input size. In QAOA parlance, such locality corresponds to a small fixed p. Chou et al. [CLSS22b] and Jones et al. [JMSS23] established a lower asymptotic bound on the QAOA's ansatz depth, showing that for a broad class of Max Constraint Satisfaction Problems (Max-CSPs), including Max-k-SAT, depth $p = \Omega\big(\frac{\log n}{\log(\mathrm{poly}(n))}\big)$ is necessary (though not

sufficient) to find near-optimal solutions. Below this threshold, the QAOA fails with high probability to reach such solutions. Beyond it, other factors may still limit approximation quality.

The second body of research views the QAOA as having a path to quantum utility [FH19, MS24, CRU24], both as an algorithm that operates in a global regime [FGG20], processing the input in total rather than in fixed segments to circumvent the OGP-related obstructions. Global in terms of ansatz depth means a p that grows, at least, logarithmically with problem size $O(log(n))$. Also possibly as a higher-depth local algorithm [BFM$^+$22] with the potential to outperform classical approximations when optimized beyond some fixed ansatz depth where it provably achieves a higher approximation ratio then is classically possible. Although work has been done to run the QAOA on near-term devices [HSN$^+$21] and attempts have been made to consider its feasibility on commercial problems [BBD$^+$22], further hardware development may be needed to realize quantum utility [GM19, HSN$^+$21].

These differing perspectives constitute a core contention regarding the potential and limitations of the QAOA. Although OGP-centered analysis rigorously establishes approximation limitations for many optimization problems [GS13, CGPR19, GJW22, CLSS22b, JMSS23], such results are average-case and hold in the large n limit [GS13]. Given a large enough number of variables n and an unsatisfiable collection of $m = \alpha n$ constraints, the near-optimal solutions of any random instance will have similar solution-space geometry [JMSS23, HS22]. As a result of this coupled geometry there exists some problem dependent fraction $\eta < 1$ that local approximation algorithms cannot, with high probability, do better than. However, for such obstructions to apply, an algorithm's locality must remain smaller than the forbidden region. Building on the existence of these geometries, the bounds in [CLSS22b, FGG20] specify a minimum locality, for any random instance, below which local algorithms cannot exceed η. Such bounds represent significant progress in understanding

local-algorithm limitations. However, in their current form, the bounds impose small limitations on locality for large problem sizes (for example, $p = 2 \to n = 10^{12}$, detailed further in Section 3.1). Consequently, the problem sizes needed for the obstructions to apply, even at the modest QAOA ansatz depth that can currently run on NISQ hardware [SP23] quickly begin to push against the capacity of what classical computers can manage.

Similarly, even though the QAOA applies broadly and offers a theoretical guarantee of better performance with increasing ansatz depth, it has yet to outperform any state-of-the-art classical algorithms at the depths studied. Nor has its performance in the global regime ($p > O(\log(n))$) been analyzed for any NP-hard problems. The latter likely stems from the greater complexity of both analyzing and optimizing the QAOA at higher depths, which further limits practical deployment.

Accordingly, this study evaluates the QAOA's approximation performance for random Max Boolean Satisfiability (Max-k-SAT) formulae exhibiting an OGP at ansatz depths ($p \leq 7$) that remain tractable to easily optimize. We identify the ansatz depths where the QAOA overcomes OGP obstructions and compare our empirical results with theoretical bounds. We also describe how the required ansatz depth for circumventing OGP obstructions changes with number of variables (n), clause count (m), and clause arity (k). We then compare the QAOA's approximation performance on formulae with an OGP to those lacking an OGP at ansatz depths that are theoretically unobstructed. For formulae exhibiting an OGP, we also compare the QAOA's approximation performance to a classical SAT solver.

The ansatz depths necessary overcomes OGP obstructions and how they relate to the theoretical minimums is of particular importance for interpreting the bound and assessing the QAOA's potential for quantum utility. On the theoretical side, in light of the large problem sizes, practical testing of the bounds remains elusory, and the tightness of the existing bounds is still undetermined. However, were the necessary ansatz depths to

significantly exceed the theoretical bound, the QAOA's practical utility may be further constrained.

As a foundational problem in computation theory, random Max-k-SAT is extensively studied, making it a suitable candidate for this study. The clause densities $\frac{m}{n}$ associated with algorithmic hardness [BH22] and satisfiability [DSS21, CO10], although not universally proven, are well-characterized in the literature. Likewise, the high-probability existence of an OGP for large n [JMSS23], and the related locality bound [CLSS22b] for algorithms such as the QAOA, have been established.

To further motivate our QAOA simulations and better characterize how the OGP manifests in Max-k-SAT, we present small-n numerical estimates of: (1) the minimum approximation ratio η for near-optimal configurations in OGP clusters, (2) how clause density and clause arity affect the OGP occurrence rate for random formulae, and (3) how changes in problem size (n) scale the forbidden region of the OGP. We also examine the ramifications of the lower bound on locality $p = \Omega\Big(\frac{\log n}{\log(\text{poly}(n))}\Big)$ from [CLSS22b] as applied to Max-k-SAT, and propose an alternative minimum ansatz depth for the QAOA to operate globally. Lastly, we simulate the QAOA at different values of our alternative bound to evaluate approximation quality.

The remainder of the paper is organized as follows. Section 2 surveys background on Max-k-SAT, the Overlap Gap Property, and the QAOA framework. Section 3 applies the current locality bound to Max-k-SAT and presents our alternative bound. Section 4 describes the methods used to generate random Max-k-SAT formulae, detect the OGP, and optimize QAOA parameters. Section 5 outlines the simulations performed, and Section 6 reports the QAOA's approximation results. Section 7 summarizes the results and closes with a discussion on quantum utility and possible future research directions.

3.1 Minimum Locality Bounds for Max-k-SAT

The OGP imposes limitations on the performance of local algorithms, including the QAOA, in finding near-optimal solutions for some optimization problems. For a large family of constraint satisfaction problems exhibiting the OGP, there exists a lower bound on the ansatz depth p below which the QAOA will w.h.p. fail to find near-optimal solutions[CLSS22b]:

$$2p + 1 \geq \frac{\log(n)}{\log\left(\frac{d(k-1)}{\log(2)}\right)}, \tag{3.1}$$

To apply the bounds to Max-k-SAT, we need to consider the limitations introduced in [JMSS23]. Specifically, these bounds only apply to (1) max-CSPs with (2) $k \geq 4$. In reworking the locality bound with these considerations, we use the asymptotic limit of the satisfiability threshold, excluding the contribution from ϵ_k in Equation 2.32. Specifically, for $k = 4$, the critical clause-to-variable ratio is:

$$\alpha^*_{4,\text{crit}} = 2^4 \ln 2 - \frac{1 + \ln 2}{2} \approx 10.24378. \tag{3.2}$$

By excluding the ϵ_k term, which tends to zero as $k \to \infty$, we adopt a more conservative threshold that does not overestimate the required problem size for higher p values. The density term is $d = k\alpha$, leading to the final bound on the required p:

$$2p + 1 \geq \frac{\log n}{\log\left(\frac{\alpha^*_{4,\text{crit}}\,(k^2-k)}{\log 2}\right)}. \tag{3.3}$$

Using Equation 3.3, the minimum problem size required for an ansatz depth just beyond $p = 1$ is about 5.6 million variables. For $p = 2$, this balloons to nearly 1 trillion variables and 10 trillion clauses. For $p = 7$—the largest ansatz depth optimized here on a single core of commodity hardware—on the order of 10^{34} variables are necessary. Regardless,

we need not go that far, at $p = 2$, the minimum problem size is well beyond the capability of modern classical SAT solvers intended for large formulae, even if those formulae are trivially satisfiable [Nad23].

These minimum locality requirements from the presence of OGP remain comparatively modest. Real NISQ hardware has already run a single iteration of the QAOA up to $p = 15$ [SP23], and classical SAT solvers designed to operate at scale often exhibit large reduction in performance on random or cryptographic formulae [GV20]. Therefore, the QAOA remains a candidate for further exploration. Moreover, for problems at that scale, other sources of difficulty, such as raw dimensionality, become pressing for optimization [LTW$^+$24].

3.1.1 Hypergraph Correspondence Bound

The results of [CLSS22b] and [JMSS23] apply broadly to general constraint satisfaction problems. As such, the significant problem sizes resulting from Equation 3.3 for small minimum p values may stem not from a lack of hardness, but from a looseness in the bound. Here we provide a simply estimate of the ansatz depth required for the QAOA to have full support on all variables.

Beginning with Equation 2.28, note that each k-clause w_i has 2^k terms in the resulting Hamiltonian. The highest locality term is k-local and acts on all the variables in w_i:

$$u_i = \bigotimes_{x \in w_i} Z_x. \tag{3.4}$$

As the lower locality terms only operate on a subset of the qubits of u_i, the growth in support with increasing ansatz depth is dominated by the u_i's. By growth in support we mean, as the depth p of the QAOA increases, the local operators in the Heisenberg picture evolve into increasingly non-local operators whose support extends over larger regions of the

system. At sufficiently large p, these operators can act non-trivially on all qubits, effectively becoming global operators. This evolution signifies a transition from a local algorithm to a global one, enabling the QAOA to 'see' the entire problem structure.

To model the the growth in support covered by the ansatz, we construct a k-uniform hypergraph $H = (V, E)$, where the vertex set V corresponds to the variables $\{x_i\}_{i=1}^n$, and the hyperedge set E consists of the clauses $\{w_i\}$. Formally, the hypergraph $H_{\phi_k(n,m)}$ associated with a Max-k-SAT formula $\phi_k(n, m)$ is defined as:

$$H_{\phi_k(n,m)} = (V, E), \quad V = \{x_i\}_{i=1}^n, \quad E = \{w_i\}_{i=1}^m. \tag{3.5}$$

Note that the negation of a variable in a clause from the source $\phi_k(n, m)$ does not affect the corresponding hyperedge. The diameter (defined in Section 2.1.1.1) of the resulting hypergraph $H_{\phi_k(n,m)}$ provides a lower bound on the ansatz depth required for the QAOA to have full support across all variables:

$$2p + 1 \geq \text{diam}(H_{\phi_k(n,m)}). \tag{3.6}$$

Using Equation 3.6 we can, for given $\phi_k(n, m)$ and p value, determine if the QAOA is operating as global or local algorithm. Similarly, for a given clause density and arity, we are able to estimate an upper bound on problems size n that QAOA can be applied to while operating globally. Figure 3.1 shows the latter relationship for random graphs at the satisfiability threshold when $k = 3, 4$. With only mild fluctuation, the diameter of these $G_{\phi_k(n,m)}$ grows slowly, with 100,000-node $k = 4$ graphs needing only $p > 2$ for full support.

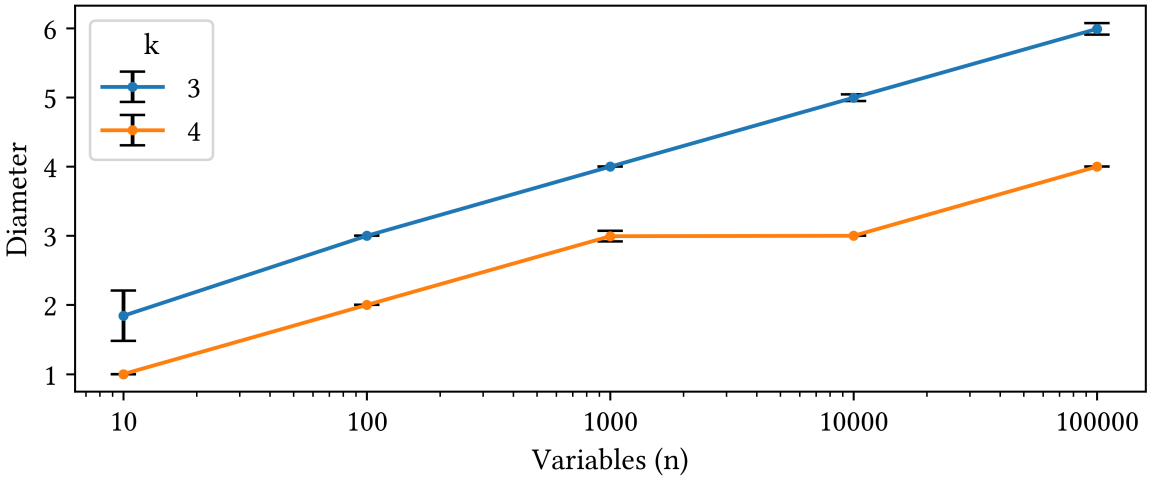

Figure 3.1: The mean diameters of random k-uniform hypergraphs derived from k-SAT formulas for $n = 10, 100, 1000, 10000, 100000$ and clause size $k = 3, 4$ at the satisfiability threshold. Each data point represents the mean hypergraph diameter calculated from 1500 randomly generated formulas. The x-axis shows the problem size n, while the y-axis corresponds to the hypergraph diameter.

Unlike the minimum problem size derived from the theoretical bound for ansatz depth $p = 2$, which is well outside our capacity to simulate, the hypergraph bound brings down problem sizes closer to NISQ or simulable scale. As we detail in Section 3.3.6, even the hypergraph bound is just beyond what can be simulated for ansatz depth $p = 2$.

3.2 Methods

In deciding the type of problems to simulate, we are constrained by the exponential simulation cost in the number of qubits. We saw above in section 3.1 that the asymptotic bounds we concern ourselves with genuinely operate in the large n regime with limited natural analogs on the scale of dozens of qubits afforded by simulation. As such, some pertinent problem considerations, such as ensuring all formulas considered exhibit an OGP, can be accommodated. In contrast, others, such as the role dimensionality plays in the

hardness of optimization, remain unaddressed. Our approach emphasizes analyzing the properties that can be feasibly considered within the simulation budget.

As we will detail further in Section 3.3, we conducted a series of numerical simulations to describe the performance scaling of the QAOA on random k-SAT formulas exhibiting an OGP. Our core method involves simulating the QAOA on random formulas while varying algorithm properties such as the hyperparameters of the QAOA, or k-SAT properties n, k, and m while systematically increasing the ansatz depth. Beyond the core QAOA simulations, we empirically detected OGP-related problem properties, such as OGP occurrence rates, as a function of clause density. Also, for both comparison and scaling reasons, we analyzed the effectiveness of classical Max-k-SAT solvers in enumerating solutions.

In all the experiments presented, we consistently used the same methods for generating random Max-k-SAT instances (Subsection 3.2.1) and detecting an OGP (Subsection 3.2.2). Since detecting an OGP requires analyzing each formula's solution space, we needed to generate solutions. For smaller instances (small n), we employed brute-force enumeration to exhaustively characterize the solution space. For larger instances, we used classical SAT solvers to find near-optimal solutions. When necessary, we performed classical simulations of the QAOA using the optimization procedures detailed in Subsection 3.2.3. Finally, details regarding the implementation and computational resources are also provided in Subsection 3.2.5.

3.2.1 Random k-SAT Generation

Random k-SAT instances are generated by uniformly sampling m clauses from the set of all possible non-contradictory clauses formed from n variables and their negations. Contradictory clauses—those containing both a variable and its negation—are excluded because they effectively reduce a clause w_i from k to $k-1$ variables. This occurs because

the clause no longer depends on a variable that is present in both literal form.

Algorithm 1 Generate Random k-SAT Instances

Input: k, n

$V \leftarrow \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$

$C \leftarrow \{c \in \binom{V}{k} : \forall x_i \in c, (\neg x_i \notin c)\}$

function RandomFormula(m)

    $I \leftarrow$ random_sample(C, m) ▷ without replacement

    return I

Algorithm 3.2.1 outlines the procedure for generating random k-SAT instances used in our simulations. The algorithm begins by constructing the set V of all literals, which includes each variable x_i and its negation $\neg x_i$ for $i = 1, 2, \ldots, n$. This results in $2n$ literals. The set C is then formed by generating all possible k-element subsets of V. Clauses that contain both a variable and its negation are contradictory and are removed. The valid clauses are then used to form random k-SAT instances by uniformly sampling m clauses without replacement for each formula.

3.2.2 Detecting an OGP

When α is beyond the algorithmic threshold and n becomes asymptotically large, the probability that a random k-SAT formula is OGP bearing approaches one. For problem sizes below such high-probability conditions, the presence of an OGP cannot be assumed and instead needs to be detected. In turn, detecting an OGP necessitates the enumeration of near-optimal solutions, as the OGP is a clustering of the pairwise distance of these near-optimal solutions. In general, as Max-k-SAT is NP-hard, such an enumeration of even the near-optimal solutions can be costly in the worst case. However, since problem sizes are

already constrained by the exponential cost of simulating a quantum circuit classically, the cost of brute force enumeration of similarly sized random k-SAT formula is often already within our computation budget. Accordingly, for small n, we enumerate all 2^n possible configuration. For problem sizes where exhaustive enumeration is still prohibitive, we use a classical SAT Solver (see Section 3.3.3 for solver selection). Again, we are buoyed by the simulation limitations, as detailed in Section 3.4.4, classical solvers enumerate well on the small random instances we are concerned with.

Per Section 3.4.1, the smaller problem sizes further restrict the clause density. To be OGP bearing with even modest probability, random formulas need to be over-constrained in clause density. Further complicating the detection procedure, the approximation ratio above which the near-optimal solutions that cluster into an OGP for random k-SAT is currently unknown. Therefore, given an optimal number of satisfied clauses C_{max}, we iteratively descendent through different thresholds $\eta_{-i} = (C_{max} - i)/C_{max}$ for the number of satisfying clauses. Here, i indexes the difference from optimal for each successive threshold. Considering only the η_{-i}-optimal configurations that meet or exceed the current threshold, we compute all the pairwise distances in this reduced solution space. We then check for clustering behavior by detecting significant gaps in the Hamming distances, which indicate the presence of the OGP's characteristic forbidden region. The following algorithm performs the analysis:

Algorithm 2 OGP Detection

```
1:  function DETECTOGP(confs)
2:      distances ← ∅
3:      for (c_i, c_j) ∈ (confs choose 2) do
4:          distances ← distances ∪ {hammingDist(c_i, c_j)}
5:      bins ← sort(distances)
6:      β ← 0
7:      for all (b_{i−1}, b_i) ∈ pairs(bins) do
8:          if b_i − b_{i−1} = 1 ∨ b_{i−1} = ∅ then
9:              β += 1
10:         else if b_i − b_{i−1} > 1 then
11:             if b_i − b_{i−1} − 1 > β then
12:                 return true
13:             else
14:                 β += (b_i − b_{i−1} − 1)
15:     return false
```

For each possible threshold below the maximum number of satisfying clauses, `detectOGP` is called with the corresponding set of configurations above that threshold level. We begin by calculating the Hamming distances between all pairs of configurations in `confs`. We then sort and store these distances. The main part of the algorithm iterates through the sorted distances (`bins`), checking for gaps after the contiguous region of size β, where β represents the length of the current contiguous block of distances. If the gap between consecutive distances b_i and b_{i-1} is larger than β, that indicates the presence of an OGP. The calls to `detectOGP` continue until either a significant gap is found or the first

$n/2$ positions in `bins` are filled. Once the first $n/2$ locations are contiguous, we stop, as the presence of an OGP is no longer possible. The remaining distances are insufficient to form a forbidden region larger than β.

3.2.3 Parameter Optimization

Farhi et al. originally proposed a polynomial grid search [FGG14] approach to parameter optimization. Since then, in practice, most work either uses a gradient-free classical optimizer [HHS$^+$24, SAT$^+$24, SSC$^+$22, PJMS$^+$24, CCZZ24, BMWV$^+$23, Cro18] like COBYLA [Pow94] or use precomputed parameter [ARCB21, WL21, GGF$^+$23, BM22b]. For the k-SAT decision problem fixed-parameter performance has compared favorably to classical SAT solvers [BM22b]. However, recent work has shown that further per-instance refinement of suitable fixed parameters can yield better approximations [HHS$^+$24]. Furthermore, this landscape independence property, where a single set of "good" parameters yield similar performance across many different formulas, has only been confirmed up to logarithmic ansatz depth [CLSS22b]. Considering the above, we opted to perform optimizations on a per-formula basis.

While there is no shortage of optimization procedures applicable to variational methods [LMS24, CCZZ24, BMWV$^+$23, Loc22] like the QAOA, we employ a simpler two-stage meta-optimization procedure akin in structure to the standard two-phase [Sch01] or coarse-to-fine [Sze11] optimization schemes. Our base optimizer remains COBYLA. As detailed in the following two Subsections 3.2.3.1,3.2.3.2, our focus was to minimize the detrimental approximation quality effects of random initial parameter selections on the optimization [LSCA21] while also remaining in a fixed computational budget. As is often the case with such bespoke optimization procedures, the computation budget informs many of the so called "magic number" hyper-parameters selected. Here a hyper-parameter is not

a QAOA optimization parameter, but instead a parameter of the optimizer or optimization procedure. The same practical limitations influence the choices of hyper-parameters in both the shallow and deep optimization phase.

After the hyper-parameters are set, we first perform multiple shallow (in terms of iteration counts) optimizations from random initial QAOA parameters $\theta_i = (\beta_i, \gamma_i)$. We then filter and select the best-performing parameter sets. Next, we refine these selected parameters in a deeper optimization phase with more iterations, focusing the search around promising regions. The second optimization stage only succeeds if it's final expectation value improves on the values from the shallow phase and any previous ansatz depth. If this condition is not met, we retry the optimization with alternative parameter sets.

3.2.3.1 Shallow Optimization Phase

At ansatz depth $p = 1$, five parameter vectors $\theta_i = (\beta_i, \gamma_i)$ for $i = 1, \ldots, 5$ are randomly generated. Each vector serves as a starting point for optimization. The QAOA is run for 40 optimization iterations, with each iteration t updating the parameter vector $\theta_i^{(t)}$. At the end of the optimization, the final expectation value $E(\theta_i^{(40)})$ and the corresponding parameters $\theta_i^{(40)}$ are recorded. During the optimization, the lowest intermediate expectation value $E_{\min}(\theta_i^{(t)})$ is tracked. If the parameters associated with this intermediate value differ significantly from the final parameters, they are also recorded.

The final collection of expectation values undergoes a Jenks binary partitioning, which detects natural boundaries in one-dimensional data. For a detailed explanation of Jenks partitioning, refer to [Jen67]. Since we are performing a minimization, the cluster of higher expectation values is filtered out. The remaining parameter sets are then sorted by the total probability of states above the OGP threshold for the given instance.

For higher values of p, the procedure differs only at the first step. Instead of five

random parameter vectors, two are produced. The remaining three vectors are generated by extending the best parameters from the previous deep optimization round with new random components. Specifically, if $\theta^* = (\beta^*, \gamma^*)$ is the best parameter set from the previous round, the new vectors are $\theta_i^{p+1} = (\beta^* || \delta_\beta, \gamma^* || \delta_\gamma)$ for $i = 3, 4, 5$, where $||$ denotes concatenation. Each increase in depth p increases the length of β and γ by one.

3.2.3.2 Deep Optimization Phase

The deep optimization phase refines the parameter vectors identified during the shallow phase. The optimization is allowed to run for a number of rounds that scales linearly with the depth p, ranging from 1500 to 4500 rounds.

The success condition for the deep optimization phase is defined as achieving a final expectation value greater than the maximum of two values: the best expectation from the previous deep optimization run and the best expectation from the current shallow optimization phase. Formally, the success condition is:

$$E_H(\theta_i) > \max(E_H^{\text{prev}}, E_H^{\text{shallow}}) + \epsilon$$

where $E_H(\theta_i)$ is the final expectation of deep optimization phase of the current ansatz depth, E_H^{prev} is the best expectation from the previous deep run, and E_H^{shallow} is the best expectation from the current ansatz depth's shallow optimization.

If the optimization fails to exceed this expectation by a tunable ϵ, it is rerun using the next set of parameters forwarded from the shallow optimization phase. If no candidate parameter vectors remain from the shallow phase, another shallow optimization phase is conducted. For each ansatz depth p a maximum of seven deep optimization rounds are performed, after which the best performing run is selected and the depth is increased.

As mentioned hereto, the number of optimization rounds scales linearly with the

ansatz depth p, and multiple attempts are allowed to reduce the likelihood of optimization failure. However, the maximum values selected for optimization depth and repetitions were not guided by theoretical considerations but rather by the practical limitations of the simulation platform.

3.2.4 Success Condition for Overall Optimization

For each ansatz depth p, at the conclusion of the deep optimization phase, the resulting state vector is analyzed to determine if the current p is sufficient to overcome the OGP obstruction—two criteria condition success for the overall optimization. In defining both criteria, let $\{s_1, s_2, \ldots, s_l\}$ denote the set of states above the OGP threshold. Let $p(s_i)$ denote the probability of state s_i, and let P_{OGP} represent the total probability of all states above the OGP threshold, it follows directly that:

$$P_{\text{OGP}} = \sum_{i=1}^{l} p(s_i)$$

1. Combined Probability Criterion: The optimization is considered successful if the combined probability of all states above the OGP threshold exceeds 0.015:

$$P_{\text{OGP}} > 0.015$$

This criterion ensures that if a modest number of measurement shots are performed, the probability of at least one of those shots measuring a state above the OGP threshold approaches unity.

2. Most Probable State Criterion: Let $\{t_1, t_2, \ldots, t_l\}$ be the l most probable states in the state vector. The overall optimization is successful if at least one of these l

most probable states is an OGP state:

$$\exists i \in \{1, 2, \ldots, l\} \text{ such that } t_i \in \{s_1, s_2, \ldots, s_l\}$$

This criterion ensures that OGP states are among the most probable and prevents scenarios where the QAOA detects the OGP states with modest probability while other higher-probability states are present.

3.2.5 Implementation Details

All simulations were conducted on the Tufts High Performance Computing (HPC) Cluster. For each experiment, we utilized up to 1,500 x86-64 processor cores simultaneously. The cluster consists of various processor models (SKUs), but the specific models used for each simulation were not recorded. Collectively, the simulations used a total of approximately 12 TB of RAM per experiment.

Python 3.10 [vR95] was used for all scripting and simulation tasks. Random SAT formula generation was implemented using in-house developed software. We conducted the QAOA simulations using a version of QOKit [LSS+23], a high-performance quantum QAOA simulator, built from source to enable the C codepath for performance optimization. We carried out optimization using a modified version of Qiskit's [Qis23] quantum algorithms wrapper for the COBYLA optimizer [VGO+20]. The RC2 solver, implemented in the PySAT [IMM18] library, was used to enumerate near-optimal solutions of the generated SAT instances. Finally, HyperNetX [PAA+23] was used for hypergraph diameter scoring.

3.3 Experimental Design

Building on the theoretical considerations discussed earlier in Section 3.1, we designed numerical simulations to assess the performance of the QAOA on Max-k-SAT instances exhibiting an OGP. Our investigation focuses on three objectives. First, we analyze the properties of the OGP in random Max-k-SAT instances to understand its influence on the performance, implementation feasibility, and potential quantum utility of the QAOA. Second, we evaluate the QAOA's performance in the global regime—when the QAOA is at sufficient depth to have full operator support in the Heisenberg picture—on instances exhibiting an OGP to determine the algorithm's capability when it is theoretically unobstructed. Third, we investigate the performance of the QAOA when its locality approaches the hypergraph diameter, which serves as a proxy for the OGP obstruction bound discussed in the literature (see Subsection 3.1.1) [CLSS22b, JMSS23].

3.3.1 Investigating OGP Occurrence Rates

To assess how clause density affects the occurrence rate of the OGP, we conducted experiments with $n = 13$ variables and clause widths $k = 3, 4, 5$, varying the number of clauses m. The experimental parameters are summarized in Table 3.1 below:

Table 3.1: Experimental Parameters for OGP Occurrence Detection

	$k = 3$	$k = 4$	$k = 5$
Variables (n)	13	13	13
Clauses (m)	100 - 3500	100 - 3500	100 - 2600
formulas per (n, k, m)	1000	1000	1000

Our procedure was as follows:

1. The values of k considered were 3, 4, and 5. For $k = 3$ and $k = 4$, the number of

clauses m was varied from 100 to 3,500 in increments of 100; for $k = 5$, m was varied from 100 to 2,600 in increments of 100.

2. For each combination of (n, k, m), we generated 1,000 random k-SAT formulas using the method described in Subsection 3.2.1.

3. Since we are only considering $n = 13$, for each generated k-SAT formula, we enumerated all $2^{13} = 8{,}192$ possible assignments and applied the OGP detection algorithm as described in Subsection 3.2.2.

Despite the small problem size of $n = 13$, this choice offers two advantages. First, with 13 variables, brute-force enumeration of all $2^{13} = 8{,}192$ possible assignments for each of the 96,000 random formulas is computationally feasible. Second, the small n allows us to empirically estimate the occurrence rate of OGP instances in small uniformly random Max-k-SAT formulas. This estimate produces a lower bound on the prevalence of OGPs for larger n because, as discussed in Subsection 3.2.2, for clause densities α above the algorithmic threshold and sufficiently large n, all Max-k-SAT formulas will, with high probability, exhibit an OGP. Therefore, for fixed k and α, as n increases, the expected fraction of random formulas not exhibiting an OGP decreases until it is exponentially small [CLSS22b]. As such, a natural estimate for the minimal OGP occurrence rate is the rate when α is near or above the algorithmic threshold for a small value of n.

3.3.2 Estimating η-Optimality for OGP Instances

A Max-k-SAT assignment ϕ is η-optimal if it satisfies at least a fraction η of the maximum possible satisfied clauses. The OGP arises only among pairs of assignments whose fraction of satisfied clauses, measured relative to the optimum, exceeds some threshold η_{OGP} [GS13]. Although Max-k-SAT has a tight inapproximability result $\eta_{\text{apx}} \leq 1 - 2^{-k}$ [Has01], and a

link between inapproximability and the OGP may exist, no characterization η_{OGP} exists.

To numerically estimate η, we generated 7,600 random SAT formulas with $n = 20$, $k = 3$, and $m = 700$ that exhibit an OGP, and recorded the resulting η for each formula. We created these instances as in Subsection 3.2.1 and detected OGPs as in Subsection 3.2.2. Each OGP configuration was enumerated using the SAT solver from Subsection 3.3.3. Additionally, we analyzed all formulas from Subsection 3.3.5.

3.3.3 Scaling Solution Enumeration with RC2 Solver

As we'll discuss in Section 3.4.1, for small n, we see relatively low random OGP occurrence rates that peak at about 15% for $n = 13$. Consequently, brute-force enumeration for each generated formula would allocate over 80% of our computational budget to formulae ultimately discarded. To better utilize available computational resources, we employ the classical RC2 MaxSAT solver for solution enumeration. In addition to reducing resource usage, RC2 enabled us to scale n up to 26 variable and to explore classes of formulae whose occurrence rates lie well below 1% (Section 3.3.6), all while remaining within our simulation budget.

RC2 is a heuristic SAT solver, and although solver performance generally trends inversely with problem size, some solvers exhibit poor performance on randomly generated formulae even for small n [GV20]. To verify that RC2 accurately enumerates near-optimal solutions, we compared its results with the brute-force enumeration from the $k = 5$ experiments described in Subsection 3.3.1.

RC2 is a MaxSAT solver that leverages existing decision-problem SAT solvers as backends and offers several tuning parameters. To conduct a comprehensive evaluation, we

tested RC2 with each of the following PySAT [IMM18] solvers:

$$\text{solvers} = [\texttt{cd}, \texttt{cd15}, \texttt{g42}, \texttt{lgl}, \texttt{mcb}, \texttt{mcm}, \texttt{mpl}, \texttt{mc}, \texttt{m22}, \texttt{mgh}, \texttt{gc3}, \texttt{gc4}, \texttt{g3}, \texttt{g4}, \texttt{mg3}],$$

using these tuning parameters:

$$\text{tuning\_parameters} = [\texttt{exhaust}, \texttt{adapt}, \texttt{incr}, \texttt{minz}],$$

where `exhaust` exhaustively enumerates solutions at each cost level, `adapt` adjusts heuristics based on partial progress, `incr` adds constraints incrementally, and `minz` reduces each unsatisfiable core size. We also tested the enumeration styles

$$\text{enumeration\_styles} = [1, 0, -1],$$

where style 1 adds blocking clauses for each optimum, style 0 reinitializes the solver after each optimum, and style -1 employs a reversed blocking-literal approach. For a more detailed description of these parameters, see the PYSAT [IMM18] documentation.

Table 3.2: RC2 Solver Configurations and Tuning Parameters

SAT Solvers	**Tuning Parameters**	**Enumeration Styles**
Available solvers in PySAT: `cd`, `cd15`, `g42`, `lgl`, `mcb`, `mcm`, `mpl`, `mc`, `m22`, `mgh`, `gc3`, `gc4`, `g3`, `g4`, `mg3`	`exhaust`, `adapt`, `incr`, `minz`	1, 0, -1

Some solvers in the above list, such as `cd`, `cd15`, `g42`, `lgl`, `mcb`, `mcm`, `mpl`, `mc`, `m22`, do not support `incr`. Any configuration pairing these solvers with `incr` was therefore excluded. From the remaining valid combinations, we obtained around 350 solver configurations in

total.

Our procedure was as follows:

1. We reused the set of random $k = 5$-SAT formulae with $n = 13$ variables and 100–2600 clauses m, generated as described in Section 3.3.1.
2. We applied RC2 with each solver configuration and tuning parameter combination from Table 3.2 to enumerate the near-optimal solutions for each formula.
3. We compared the OGP detection results from RC2 with the results of brute-force enumeration to gauge accuracy.
4. We recorded the execution time for each configuration.

Our goal was to find the fastest RC2 configuration whose detection of OGP configurations aligned with the $k = 5$ brute-force enumeration. We assessed each configuration according to its execution time and the number of OGP configurations correctly enumerated, seeking both accuracy and efficiency.

3.3.4 Comparing QAOA Performance with and without OGP

The presence of an OGP limits the average-case approximation performance of local algorithms. This limitation manifests in the pairwise solution-space as forbidden regions of low-quality solutions that algorithms with limited locality are, w.h.p., unable to optimize through [GS13, GS17, BH22, CLSS22b]. However, as detailed in Section 3.1, the obstruction rely on the limited locality. Once the QAOA's locality exceeds that bound (Equation 3.3), the corresponding proof may no longer hold, but the underlying solution-space topology remains unchanged, and the effects of an OGP on approximation performance in this regime remain unknown.

To evaluate the OGP's effect on approximation performance when the QAOA has sufficient locality, we generated approximately 2,000 random SAT formulae with $n = 16$, $k = 4$, and $m = 800$ that exhibit an OGP, as described in Subsection 3.2.1. We also generated another 2,000 formulae that do not exhibit an OGP, identified using the method in Subsection 3.2.2. With 800 clauses and 16 variables, the hypergraph diameter of all generated formulae is 1 (see Section 3.1.1). Therefore, the QAOA at ansatz depth $p = 1$ meets both the obstruction bound (Equation 3.3) and the hypergraph bound. Finally, we optimized the QAOA to $p = 7$ for both OGP and non-OGP samples, as detailed in Section 3.2.3.

3.3.5 Assessing the Impact of Scaling n, k, and m on QAOA

To evaluate how scaling each random k-SAT parameter (n, k, m) affects approximation performance when an OGP is present, we generated random formulae using the method from Subsection 3.2.1. We enumerated near-optimal solutions with the RC2 SAT solver (Subsection 3.3.3), performed OGP detection (Subsection 3.2.2), and then applied QAOA (Subsection 3.2.3). The simulation parameters are summarized in Table 3.3:

Table 3.3: Experimental Parameters for Depth Scaling

k=4, m=800	**n=16, m=800**	**n=16, k=4**
$n = 14$	$k = 3$	$m = 200$
$n = 16$	$k = 4$	$m = 500$
$n = 18$	$k = 5$	$m = 800$
$n = 20$	$k = 6$	$m = 1100$

The experiments proceeded in three stages to isolate the effect of each parameter:

1. We varied the number of variables n from 14 to 20 while keeping $k = 4$ and $m = 800$ fixed.

2. We varied the clause width k from 3 to 6 while keeping $n = 16$ and $m = 800$ fixed.

3. We varied the number of clauses m from 200 to 1100 while keeping $n = 16$ and $k = 4$ fixed.

For each (n, k, m) combination in Table 3.3, we generated random formulas in sample sizes ranging from 1,100 to 3,500. he exact sample sizes were determined computational resource constraints. We then optimized the QAOA to an ansatz depth of $p = 7$ for each formula to analyze performance scaling.

3.3.6 Exploring QAOA Depth Requirements via Hypergraph Diameter

As detailed in Section 3.1, evaluating QAOA performance at the OGP obstruction bound for local algorithms, even an ansatz depth $p = 2$ necessitates formulae with nearly 10^{12} variables. In Section 3.1.1, we introduced a mapping from SAT formulae $\phi_k(n, m)$ to a k-uniform hypergraph $H_{\phi_k(n,m)}$. The diameter $\mathrm{diam}(H_{\phi_k(n,m)})$ of this hypergraph functions a lower bound (Equation 3.6) on the ansatz depth p required for the QAOA to act non-trivially on all qubits. By running the QAOA at increasing depth on formulae with different diameters, we examined how graph diameter affects approximation performance. Specifically, we compared $\mathrm{diam}(H_{\phi_k(n,m)}) = 2$ and 3.

Ideally we would compare the QAOA's approximation performance on formulae whose diameters align with different minimum ansatz depths. However, although the hypergraph mapping reduces the 10^{12}-variable problem sizes to scales closer to what simulation afford, an ansatz depth of $p = 2$ still involves hundreds of variables. Even so, this reduction opens possibilities for further exploration. With suitable filtering and an

appropriate choice of (n, k), we are able to compare the QAOA's approximation performance on formulae at the satisfiability threshold $\alpha_{k,\text{crit}}$ with:

1. $\text{diam}(H_{\phi_k(n,m)}) = 3$ without an OGP,
2. $\text{diam}(H_{\phi_k(n,m)}) = 2$ with an OGP,
3. $\text{diam}(H_{\phi_k(n,m)}) = 2$ without an OGP,
4. $\text{diam}(H_{\phi_k(n,m)}) = 3$ with an OGP.

The $\text{diam}(H_{\phi_k(n,m)}) = 3$ formulae are particularly important because diameter 3 is the maximum reach for ansatz depth $p = 1$. Any performance difference between diameters 2 and 3 would suggest further OGP-related approximation constraints. Additionally, although the OGP obstruction bound for local algorithms is verified only for clause widths $k \geq 4$, we use $k = 3$ out of necessity because raising k to 4 lowers the expected diameter.

To select a suitable variable count, we generated about 320,000 3-SAT formulae with variable counts from 20 to 29. The number of diameter-three hypergraphs in each set is shown in Table 3.2. Each candidate size was sampled 32,000 times, with the highest observed occurrence rate of 0.090625% for $n = 29$. All tests were performed at the satisfiability threshold for each variable count (for example, $(n = 26, k = 3)$ corresponds to $m = 123$ per Equation 2.32). Increasing m introduces more edges into the corresponding hypergraph and thereby lowers the expected diameter. Because these diameters were so infrequent, we chose $n = 26$ as it was the largest problem size we could simulate under our computational budget.

Similarly, as described in the results on OGP occurrence rates (Figure 3.3), the closer a given clause density α is to $\alpha_{k,\text{crit}}$, the lower the probability that a random formula exhibits an OGP, with values near that threshold often below 1%. The combination of these

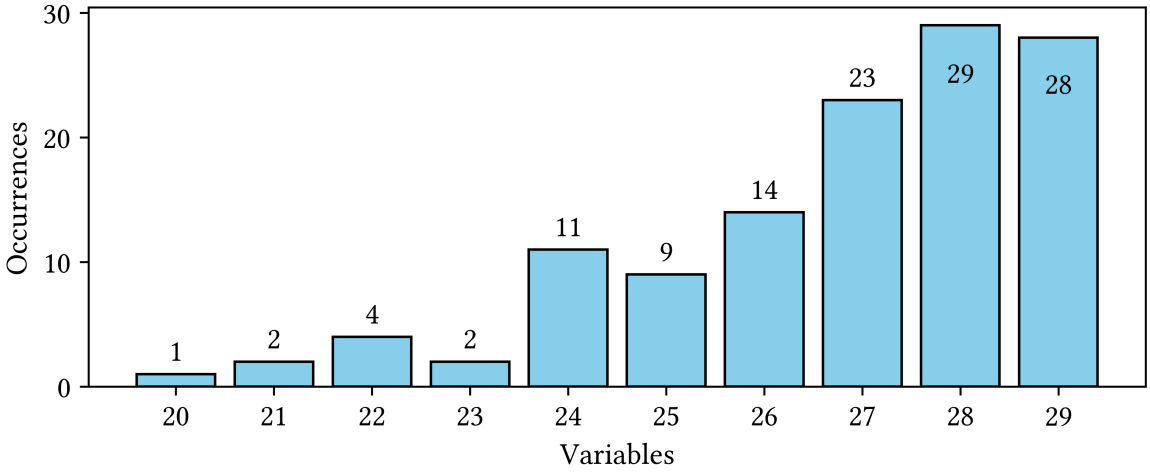

Figure 3.2: Occurrences of diameter-three hypergraphs for 3-SAT formulae with variable counts from 20 to 29, out of 32,000 random samples per variable count. Each bar on the x-axis corresponds to the number of variables, and the y-axis indicates how many of those samples yield hypergraphs with diameter 3. The text labels on each bar show the exact number of occurrences.

two low occurrence conditions, $\text{diam}(H_{\phi_k(n,m)}) = 3$ with an OGP, necessitated substantial rejection sampling.

As in the other simulations, each random $(26, 3, 123)$ 3-SAT formula was generated using the procedure in Section 3.2.1. About 500 samples were taken for each of the four configurations. We employed HyperNetX [PAA+23] to compute hypergraph diameters, used the RC2 SAT solver (see Subsection 3.3.3) to enumerate near-optimal solutions, and performed OGP detection following Section 3.2.2. QAOA was then optimized as described in Subsection 3.2.3, but only up to $p = 4$ due to resource constraints.

3.4 Results

Presented are the numerical findings from the simulations detailed in Section 3.3. Specifically, we estimate the QAOA approximation performance across a range of Max-k-SAT instances

that exhibit OGPs. We consider how changes in n, k, m influence algorithmic behavior (Sections 3.4.3), as well as how different minimum requirements for full support in the Heisenberg picture affect performance (Section 3.4.6). As a point of comparison, we consider the QAOA performance on instances of Max-k-SAT without an OGP (Section 3.4.5), and how well classical solver-based algorithms perform on similar Max-k-SAT instances (Section 3.4.4). We also provide numerical estimates for OGP properties as they appear in Max-k-SAT (Sections 3.4.1, 3.4.2).

3.4.1 OGP Occurrence as a Function of Clause Density

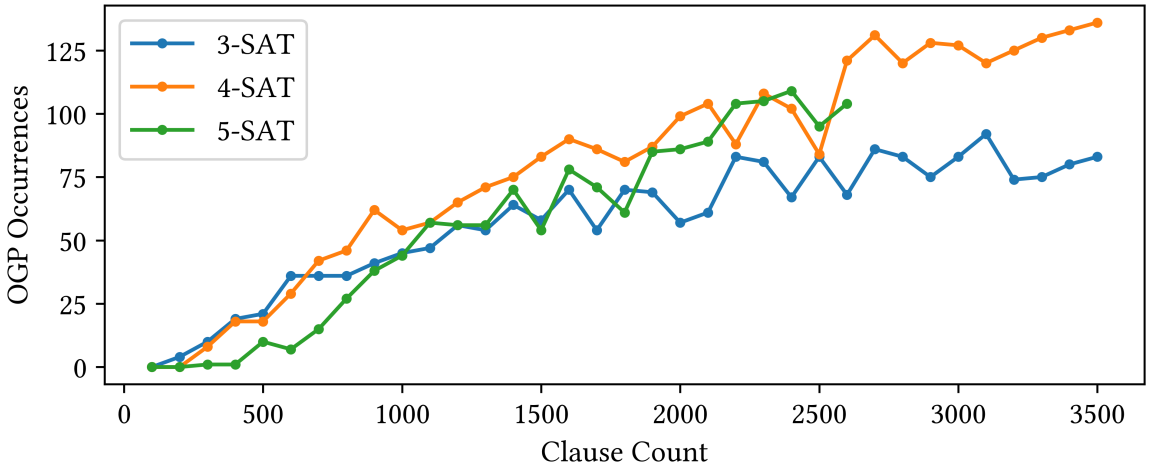

Figure 3.3: Overlap Gap Property (OGP) occurrence rates as a function of clause count for 3-SAT, 4-SAT, and 5-SAT problems with $n = 13$ variables. The curves represent 3-SAT (blue), 4-SAT (orange), and 5-SAT (green), respectively. For each clause count, we generated 1,000 random formulas and calculated the OGP occurrence rates. The OGP occurrence rate increases with clause count in all three cases and is minimal below the satisfiability threshold.

Figure 3.3 shows OGP occurrence rates plotted against the number of clauses m for 3-SAT, 4-SAT, and 5-SAT instances with $n = 13$ variables. No OGPs appear at $m = 100$ regardless of arity. For 4-SAT and 5-SAT, OGP occurrence rates rise gradually once the

clause density α exceeds certain thresholds, specifically $m = 300$ for 4-SAT and $m = 500$ for 5-SAT. Unlike 4-SAT, which has no OGP below $m = 300$, 5-SAT exhibits single OGP samples at $m = 300$ and $m = 400$, but does not start growing until $m = 500$. For 3-SAT, OGP occurrences begin at the next data point considered, $m = 200$. As such, data resolution precludes us from identify a similar threshold for 3-SAT.

Examining how OGP occurrence rates change with increasing clause density, the 3-SAT curve levels off near 8%. Because the total number of possible clauses (including contradictory ones) for 3-SAT with 13 variables is $\binom{2n}{3} = \binom{26}{3} = 2{,}600$, this leveling off at higher densities suggests a saturation point due to the finite clause space. By contrast, the 4-SAT and 5-SAT occurrence rates increase more steadily with α. For 4-SAT, the occurrence reaches 13% at $\alpha \approx 270$ (3,500 clauses). Our 5-SAT samples end at $m = 2600$, here the 4-SAT and 5-SAT rates are similar, with the 5-SAT rate just over 10% at $\alpha \approx 200$.

k	n	**Critical** $\alpha^*_{k,\mathbf{crit}}$	**Threshold Clauses** m^*	**OGP Occurrence** $m_{\mathbf{OGP}}$	**Threshold Multiple**
3	13	4.2670	$m^* = 4.2670 \times 13 \approx 55$	200	$\frac{200}{55} \approx 3.636$
4	13	10.2438	$m^* = 10.2438 \times 13 \approx 133$	300	$\frac{300}{133} \approx 2.256$
5	13	21.1170	$m^* = 21.1170 \times 13 \approx 275$	300	$\frac{300}{275} \approx 1.091$

Table 3.4: Satisfiability thresholds and first OGP occurrence for k-SAT with $n = 13$ variables. The "Threshold Multiple" is the ratio of the first OGP occurrence clause count to the satisfiability threshold clause count.

For $(n = 13, k = 3)$, the first time a single instance occurs is at $m = 200$ clauses, corresponding to a clause density over three times the satisfiability threshold. For $(n = 13, k = 4)$, no OGPs appear until $m = 300$, which is just over twice the corresponding satisfiability threshold. For $(n = 13, k = 5)$, an OGP is first observed just beyond the threshold at $m = 300$; however, there are only single occurrences at both $m = 300$ and $m = 400$, increasing to 10 by $m = 500$. These findings indicate that while higher k requires higher clause densities to achieve comparable OGP occurrence rates, the difference between

the satisfiability threshold and first OGP appearance diminishes as k increases. Specifically, for $k = 5$, OGPs appear at clause counts much closer to the threshold than for $k = 3$ or $k = 4$. Finally, no OGPs appear for $k = 3$, $k = 4$ or $k = 5$ at clause counts below their satisfiability threshold.

3.4.2 Estimating η-Optimality for OGP Instances

Figure 3.4 shows the distribution of thresholds for OGP onset, above which the OGP topology occurs, as measured in approximation ratios. The upper panel is for SAT formulae with $n = 20$, $k = 3$, and $m = 700$. The average approximation ratio is 0.998. The discrete gaps in the distribution reflect the inherent discrete values of Max-SAT solutions. For $m = 700$, approximation ratio changes occur in multiples of $\frac{1}{700}$. Similarly, Figure 3.4 (lower) shows the distribution of approximation ratios for formulae where the OGP was detected via brute-force enumeration, as described in Subsection 3.3.1. Since we consider samples with varying n, k, m here, there are no gaps in the distribution. The average approximation ratio remains 0.998. This ratio exceeds the inapproximability bound of 0.875 for 3-SAT [Has01]. For the small n samples considered here, the difference between these two approximation ratios implies that the OGP obstructs only the most near-optimal solutions

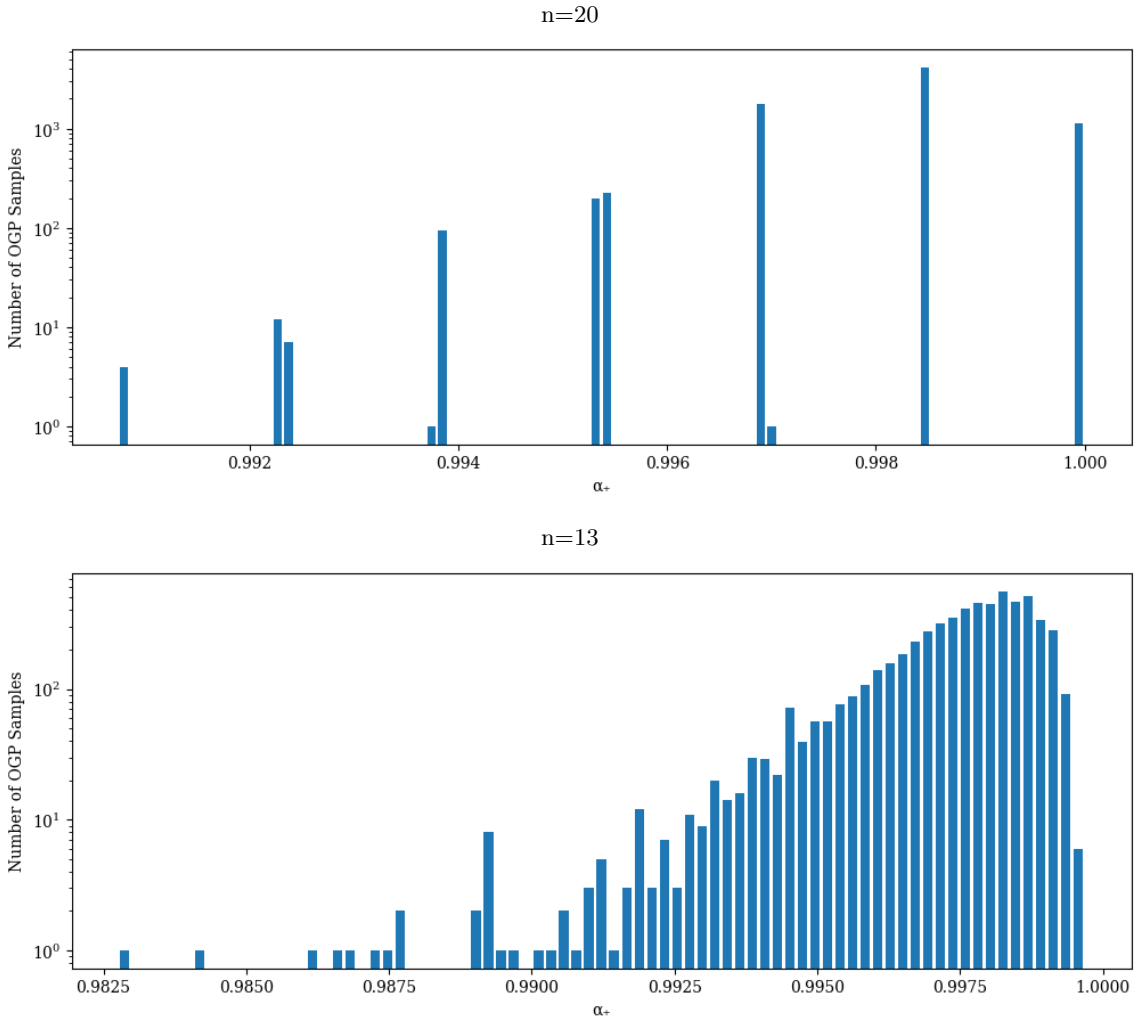

Figure 3.4: Distribution of thresholds for OGP onset in terms of approximation ratios for SAT formulas. The upper panel shows the distribution for $n = 20$, $k = 3$, and $m = 700$, whereas the lower panel shows the distribution for $n = 13$, $k = 3, 4, 5$, and $m = 100 - 3500$ in increments of 100. Both distributions have an average ratio near 0.998.

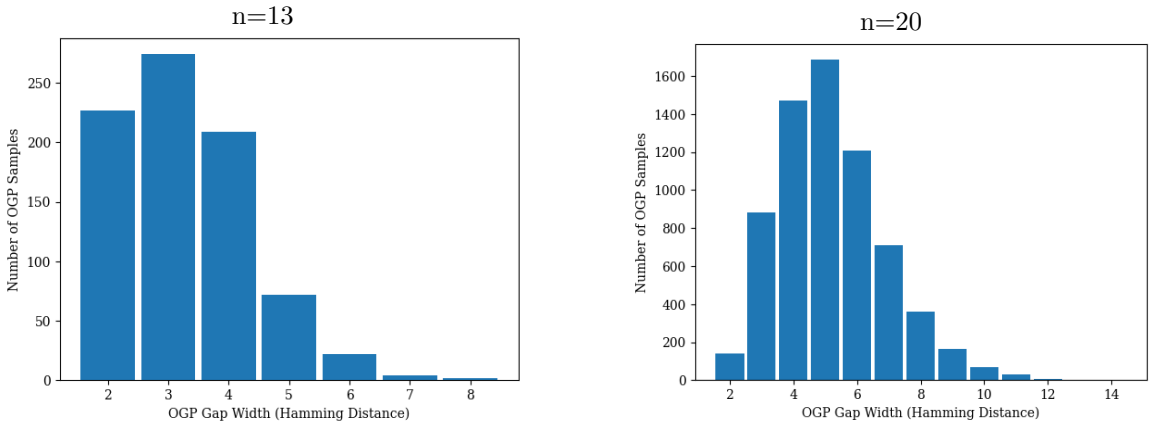

Figure 3.5: Distribution of OGP Gap width with increasing n. The gap width is the minimum hamming distance between pairs of configurations. The left panel shows the distribution of gap widths for $n = 13$, $k = 3, 4, 5$, and $m = 100 - 3500$ in increments of 100. The right panel shows the distribution of gap widths for $n = 20$, $k = 3$, and $m = 700$. The central gap width increases from 3 to 5 as n grows, indicating a shift toward larger gap widths with larger problem sizes.

Although the average point for OGP occurrence remained the same, the most common gap width rose from 3 to 5, as shown in Figure 3.5. By gap width, we refer to the minimum Hamming distance between pairs of configurations above the OGP threshold. This change aligns with Panel 1 of Figure 3.6, which shows an increase in the required ansatz depth from $p = 1$ to $p = 3$ to maintain the same measurement probability when the variable count increases from $n = 14$ to $n = 20$. Regardless of n, all samples have full support at $p = 1$ in the QAOA which may indicate a further dependency on gap width beyond full support.

3.4.3 QAOA Approximation Performance Scaling with n, k, and m

Figure 3.6 summarizes QAOA simulations for changes in the number of variables (n), the number of variables per clause (k), and the number of clauses (m). We observe

that the minimum ansatz depth needed to maintain the same measurement probability for configurations above the OGP threshold increases with n and k, but remains nearly constant with respect to m. Using the criterion in Section 3.2.4, the minimum ansatz depth rises from $p = 1$ to $p = 3$ as n goes from 14 to 20. The theoretical bounds in Section 3.1 predicted an ansatz depth of $p = 1$ for all simulated values of n.

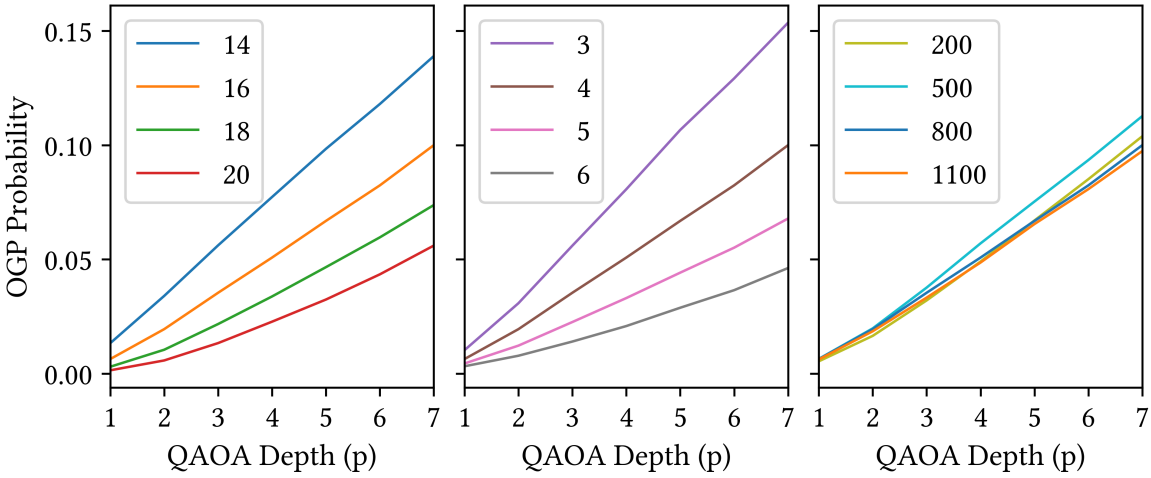

Figure 3.6: Approximation performance of the QAOA as a function of ansatz depth p. Left: $n \in \{14, 16, 18, 20\}$ (curves labeled by n). Middle: $k \in \{3, 4, 5, 6\}$ (curves labeled by k). Right: $m \in \{200, 500, 800, 1100\}$ (curves labeled by m). "OGP probability" denotes the total probability mass on configurations whose satisfied-clause fraction exceeds the instance-specific OGP threshold. For each (n, k, m, p) we optimize QAOA parameters (Section 3.2.3) and average this probability over the indicated ensemble sizes (about 1,100–3,100 formulas per panel). The probabilities increase approximately linearly with p. For a fixed p, approximation quality increases as n or k increase, but shows little sensitivity to changes in m.

Similarly, raising k from 3 to 6 boosts the required ansatz depth from $p = 1$ to $p = 2$ to retain the same average approximation performance. As k increases, the average probability of measuring states above the OGP threshold declines. As with changes in n, these findings differ from theoretical bounds that predict $p = 1$. Regarding those bounds, there are two important points: they currently apply only to even $k \geq 4$, and we observe no

further difference in approximation quality for values of k that fall within or outside that regime. In addition, Equation 3.3 has a $\frac{1}{k}$ factor suggesting better performance with larger k, but we do not see that trend. Finally, in the Heisenberg picture, higher k raises the minimal locality of the QAOA cost Hamiltonian, which would otherwise suggest a smaller overall p for full support. In practice, that higher locality does not dominate approximation performance.

When varying the number of clauses (m), the measured OGP-above-threshold probability under the optimized QAOA state changes little over the range of m tested. The QAOA depths at which OGP states meet our acceptance criterion (between $p = 1$ and $p = 2$) almost match the theoretical bounds. A slight performance increase is observed for formulae with $m = 500$ clauses, where the probability of measuring OGP states is somewhat higher beyond $p = 3$.

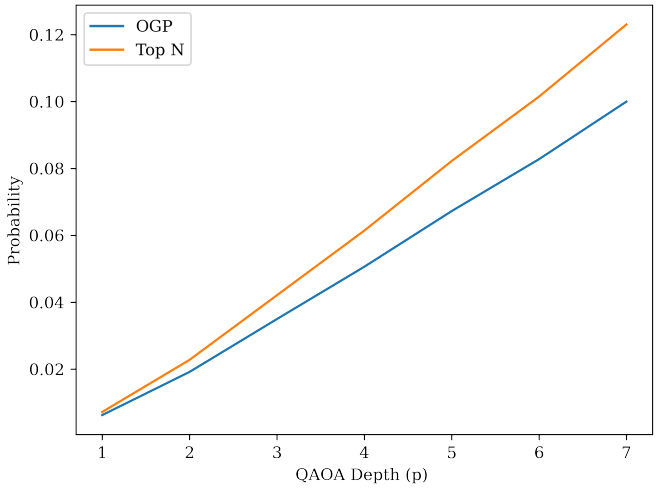

Figure 3.7: Comparison of the probabilities for measuring (1) all configurations above the OGP threshold and (2) the top N most likely configurations, both as functions of QAOA ansatz depth (p). Each data point is an average over about 2,000 formulas with an OGP. All formulas have 16 variables, 4 variables per clause, and 800 clauses. The orange line shows the probability of the top N states after optimization, where N matches the number of states above the threshold. The blue line is the total probability of all configurations above the OGP threshold. Both lines grow linearly with increasing p. While the probabilities are close, the most likely configurations always have a higher combined probability, the gap between the two curves also expands with p.

Figure 3.7 shows the QAOA measurement probabilities for $n = 16$, $k = 4$, and $m = 800$. It compares the total probability of configurations above the OGP threshold with the probability of the most probable states resulting from the QAOA optimization. Both as a function pf ansatz depths. Both probabilities increase linearly as p grows. While the most probable states have a consistently higher combined probability, they grow similarly because the same configurations appear in both sets. This similarity suggests that the QAOA is not strongly under-weighing OGP configurations due to topological obstructions.

3.4.4 Solver-Based Scaling and Validation

Table 3.5 lists the ten fastest configurations that successfully enumerated all configurations above the OGP threshold for the 1,423 5-SAT formulas generated in Subsection 3.3.1 of the Experimental Design. We selected the fastest configuration, which utilized the gc3 solver, for all subsequent solver-based OGP enumeration.

SAT Solver	**RC2 Parameters**	**Enumeration**	**Average Time (s)**
gc3	[True, False, False, False]	1	3.5648
gc3	[True, True, False, False]	1	3.6000
gc4	[True, False, False, False]	1	3.6628
gc4	[True, True, False, False]	1	3.6913
gc4	[True, False, True, False]	1	4.0082
gc4	[True, True, True, False]	1	4.0486
gc3	[True, False, False, True]	1	4.1430
gc3	[True, True, False, True]	1	4.1812
gc4	[True, False, False, True]	1	4.2144
g3	[True, False, False, False]	1	4.2164

Table 3.5: The top ten fastest configurations for the RC2 solver, sorted by average enumeration time in seconds. Each configuration consists of a specific SAT solver (gc3, gc4, g3) and a set of tuning parameters. The enumeration column indicates the enumeration style used, with 1 being the consistent choice across all configurations. The fastest configuration employed the Glucose 3 [ALS13] (gc3) solver with the parameters [True, False, False, False], achieving an average time of 3.5648 seconds. All solvers listed effectively enumerating all configurations above the OGP threshold for the 1423 5-SAT formulas detailed.

3.4.5 QAOA Performance with and without OGP

In Figure 3.8, we present the average performance across 2000-3000 formulae of the QAOA on random Max-4-SAT formulas with and without an OGP. By "performance" we mean

the probability that a measurement outcome from the optimized QAOA state lies above the OGP threshold (i.e., is near-optimal). For formulae that have an OGP, we consider the average probability of measuring a configuration beyond the OGP threshold. For formulae without an OGP, we average the probability of the six most satisfied configurations. All formula have a hypergraph diameter of one ($\text{diam}(H_{\phi_k(n,m)}) = 1$) and are theoretically unobstructed at $p = 1$. The curves representing averages across OGP and non-OGP formulae show near indistinguishable performance for depths p up to 7. Specifically, the probability for detecting near-optimal solutions remain within a margin less than 2×10^{-3} on average. At each ansatz depth, the probabilities of detecting configurations above the OGP threshold are nearly identical to the combined probability of the six most probable states.

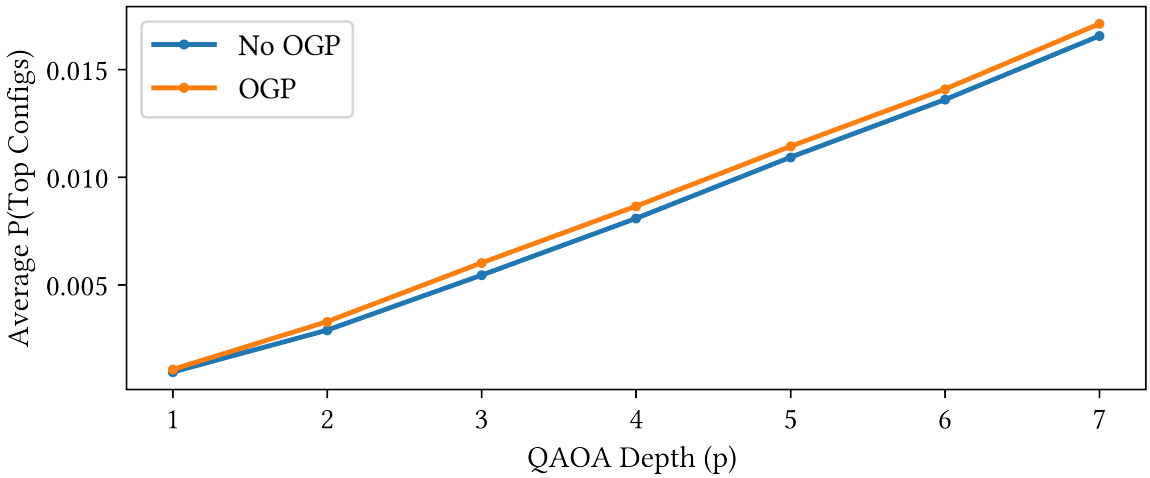

Figure 3.8: Comparison of QAOA performance on random Max-4-SAT formulas (n=16 variables, m=800 clauses) with and without the Overlap Gap Property (OGP). The figure shows the average probability of detecting a near-optimal configurations as a function of QAOA ansatz depth p. The orange line shows the average probability of measuring a state beyond the OGP threshold, whereas the blue line indicates the average probability of the six most satisfied configurations. Each point represents a nest average over approximately 2,000 to 3,000 formulas.

3.4.6 QAOA Performance at the Satisfiability Threshold

Figure 3.9 shows QAOA approximation performance for random 3-SAT formulas with 26 variables at the satisfiability threshold ($m = 123$). Each data series corresponds to one combination of hypergraph diameter and OGP presence. Each data point is a nested average over about 500 formulas. For formulas that have an OGP, we compute the average probability of the configurations above the OGP threshold:

$$\bar{P}_{\text{OGP}} = \frac{1}{l} \sum_{i=1}^{l} p(s_i)$$

where l is the number of configurations above the OGP threshold, and $p(s_i)$ is the probability of configuration s_i. For formulas that do not exhibit an OGP, we compute the average probability of the top six most probable states:

$$\bar{P}_{\text{top-6}} = \frac{1}{6} \sum_{i=1}^{6} p(s_i)$$

Here, $p(s_i)$ is the probability of the i-th most probable configuration. We chose six configurations because six is both the mode and median number of configurations above the OGP threshold for diameter-three and diameter-two formulas.

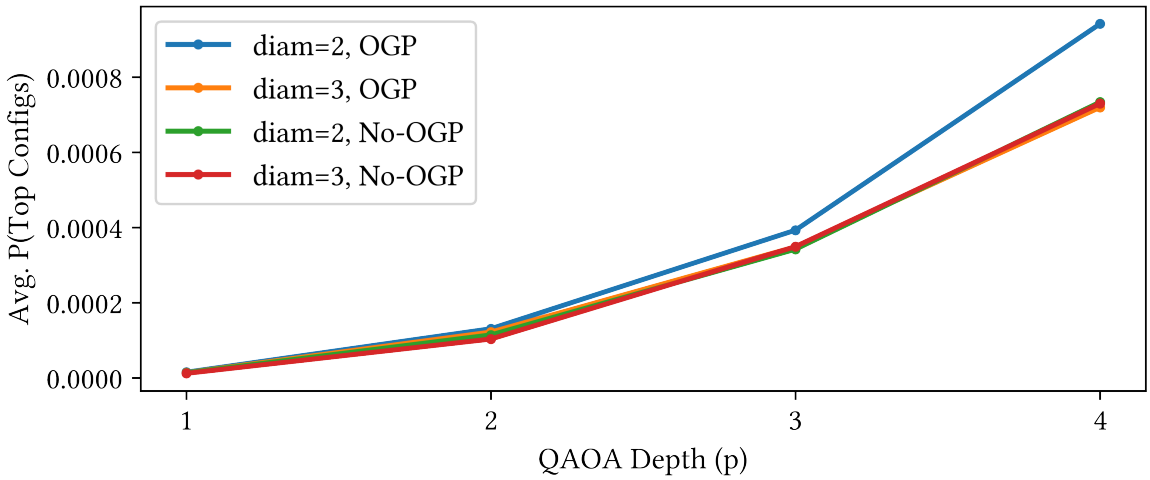

Figure 3.9: Probability of detecting configurations above the OGP threshold as a function of QAOA ansatz depth p for random 3-SAT formulas with 26 variables at the satisfiability threshold ($m = 123$). The four curves show different combinations of hypergraph diameter and OGP presence: (i) diam $= 3$ without an OGP (red), (ii) diam $= 3$ with an OGP (orange), (iii) diam $= 2$ without an OGP (green), and (iv) diam $= 2$ with an OGP (blue). Each data point is based on about 500 formulas for each depth from 1 to 4, and represents a nested average. For each sample, the average probability of seeing the states above the threshold is computed. These values are, in turn, averaged across samples. For formulas with an OGP, we compute the average probability of the configurations above the OGP threshold; for formulas without an OGP, we compute the average probability of the top six most probable states. Note that all but the blue line are completely overlapped.

At ansatz depths $p = 1$ and $p = 2$, the probabilities of detecting configurations above the OGP threshold are nearly identical across all four cases. As the ansatz depth increases to $p = 3$ and $p = 4$, a slight separation emerges between the curves. Notably, the case with hypergraph diam $= 2$ and OGP exhibit a marginally higher probability compared to the others. However, the overall differences remain small.

Chapter 4

Non-Local Optimization

Earlier work characterized the locality of the standard QAOA; here we ask whether longer-range interactions can be encoded at fixed depth while offering similar approximation quality and increased operator support. We tested three higher locality mixers and a continuous-time approximation protocol Ballistic Optimization against the standard QAOA at depth one on 3-regular graph with. The higher-locality mixers match or under-perform standard QAOA, whereas Ballistic Optimization lifts the mean approximation ratio by about 0.13.

Noisy Intermediate-Scale Quantum (NISQ) devices are near-term quantum computers distinguished by their limited qubit counts, susceptibility to undetectable and uncorrectable errors from noise which limits them to order 10^3 gates [Pre18]. While their practical utility remains a research question, with potential applications in fields such as quantum chemistry, optimization, and quantum machine learning [BCLK$^+$22].

Like the related Variational Quantum Eigensolver (VQE) [PMS$^+$14], the QAOA is a hybrid quantum-classical algorithm with ansatz state produced by a parameterized circuit. The objective function value is estimated from samples taken after multiple identical

preparations of the ansatz state. The parameters are then tuned via a classical optimization loop, or in some cases, may be classically precomputed. However, unlike VQE, where the approximate ground state is for a physical system, the QAOA ground state optimizes a cost function of a combinatorial problem encoded into the Hamiltonian. A distinguishing feature of the QAOA is its tunable depth. As the depth parameter, $p \to \infty$, the QAOA recovers the optimal solution, and at shallower finite depths, it operates as a monotonic approximator [FGG14].

In contrast to exact solution algorithms, which only seek to find the optimal solution to a problem, the performance of approximation algorithms is characterized by proximity to the optimal solution. An algorithm is α-approximation if, for any input, it returns a feasible solution whose value is within a factor of α of the value of an optimal solution. This factor α is known as the approximation ratio, and it quantifies the worst-case deviation from optimality that the associated approximation algorithm exhibits.

Early in its development, QAOA was shown—at the level of expressivity (i.e., assuming access to optimal parameters)—to achieve an approximation ratio exceeding the best-known classical bound for MAX-3-LIN-2 [FGG15, FGG14]. This optimization problem comprises n variables and a collection of m 3-variable linear terms $(x_1 + x_2 + x_3)$. The objective is to maximize the number of satisfied terms under modulo 2 arithmetic. With an approximation ratio $\Omega(1/D^{3/4})$ [FGG14], the QAOA improved upon the best classical approximation algorithms. However, subsequent development improved the state-of-the-art classical bound to $\Omega(1/D^{1/2})$, outperforming the QAOA [BMO$^+$15, Has19].

Max-Cut was another case where the (expressive) approximation ratio of $QAOA_p$ was computed analytically [FGG14]. The objective of Max-Cut is a bipartitioning of the adjacent nodes in a graph. The QAOA's worst-case performance for $p = 1$ on 3-regular graphs is an approximation ratio of 0.6924. While not competitive with the best-known

classical algorithms, this bound established expectations for the effectiveness of the QAOA as p increased [FGG14]. In 2021 the 3-regular $p = 2$ approximation ratio of 0.7559 was computed [WL21].

Even against the harder case of general graphs, the 3-regular $p = 2$ performance of the QAOA does not outperform the best known classical algorithm, Goemans-Williamson, which boasts an approximation ratio of ≈ 0.87856 [GW95]. Furthermore, for cubic (at most degree 3) graphs, Halperin *et al.* introduced an algorithm, that uses the Goemans-Williamson as a subroutine, with an approximation ratio of 0.9326 [HLZ04]. While there is no known upper bound for cubic graphs (or 3-regular graphs), the Halperin algorithm is near the tightest known upper bound on Max-Cut's approximation ratio (for general graphs, 0.941) [Has96]. Finally, considering only local classical algorithms, recent findings have demonstrated that a local classical algorithm can outperform the QAOA at $p = 2$ for high-girth regular graphs, casting doubt on the quantum advantage for small p [BM22a].

While it has always been understood that the QAOA's approximation quality is depth-dependent, recent analysis has clarified this relationship. Specifically, at shallow (logarithmic and below) circuit depths, the QAOA optimization procedure is, with high probability, unlikely to find high-quality approximations. This is due to the Overlap Gap Property (OGP), where the solution space contains narrow clusters of high-quality, high-similarity solutions surrounded by broader regions of varied, suboptimal alternatives. Algorithms like the QAOA, which locally explore the solution space, are hampered by the OGP. 'Small changes'—alterations reliant solely on fixed amounts of local information like adjacent nodes in a graph—prove insufficient. The presence of an OGP, with gaps exceeding the algorithm's local purview, necessitates that such algorithms traverse lower-quality regions to reach optimal or near-optimal solutions. Such paths are unlikely to be taken as they appear less optimal for incremental optimization regimes like the QAOA's

[FGG20, BGMZ22, CLSS22a].

The local structure of the QAOA becomes evident when analyzing the Hamiltonians that comprise the algorithm. In a given basis (here we use the Pauli basis $\{\sigma_x, \sigma_y, \sigma_z\}$), a Hamiltonian can be expressed as tensor products of basis operators and identity I:

$$H = \sum_{i=1}^{m} \beta_i Q_i \tag{4.1}$$

where β_i are real coefficients. This representation of a Hamiltonian is considered efficient if the number of terms m increases only polynomially with the number of qubits. The locality k of the Hamiltonian is the maximum locality of any term Q_i, which refers to the number of non-identity tensor factors in Q_i [Llo96]. As we will detail in the following section, both the mixing and cost Hamiltonians that comprise the QAOA exhibit this local structure.

In response to the challenges outlined, this Chapter presents a new class of QAOA mixers and an optimization technique that incorporates terms that couple vertices beyond adjacency. This approach is designed to enhance the QAOA's performance at shallow depths, addressing the limitations linked to the algorithm's inherent local structure.

4.1 Obstructions

4.1.1 OGP Obstructions for Local Algorithms

In Chapter 3, we saw that the locality of the QAOA is tied to its depth. Obstructions to the QAOA manifest as lower bounds on the p required to reach near-optimal approximations. These bounds are for average-case performance; how well the QAOA does on typical instances. The problem instances are not plants (specifically selected hard or easy instances) but are instead randomly sampled from some natural distribution of instances [FGG20, CLSS22a, BGMZ22].

This type of average-case QAOA obstruction was first introduced by Farhi *et al.* in [FGG20], where they demonstrated an OGP-based obstruction of the QAOA for the Maximum Independent Set (MIS) problem, emphasizing that the QAOA must 'see' the entirety of the graph to yield an optimal solution. Specifically, for graphs draw from an Erdos-Renyi distribution with fixed average degree, the QAOA cannot do better than 0.854 times the size of optimal set if p is less than a graph-degree-dependent factor of $\log(n)$. This was shown by discretely interpolating between two graph instances that have an OGP and demonstrating that at some point along the interpolation, there must exist an overlap between the two instances in the forbidden range described in 2.39, a contradiction.

This style of interpolation analysis was then extended to the hypergraph (a graph where a single edge can connect many nodes) setting in [CLSS22a]. There the authors demonstrate the existence of a $\log(n)$ depth obstruction for random instances of MAX-k-XOR when $k \geq 4$. More importantly, the analysis introduces a broader class of local algorithm (defined below in 4.1.1.2) that subsumes both local quantum algorithms like the QAOA and local classical algorithms (defined below in 4.1.1.1). To do this they also use a strengthened version of the OGP known as the 'coupled' OGP, a notion first introduced in [CGPR19]. A coupled OGP applies near optimal solutions σ_1 and σ_2 of any pairs of sampled instances G_1 and G_2. Above some threshold α_c, the overlap between the cross sample solutions $R(\sigma_1, \sigma_2)$ is also not in the forbidden region $[\overline{\delta}, \overline{\beta}]$.

4.1.1.1 Classical Local Algorithms

In the broadest sense, a local algorithm is a process that computes portions of its final output by only considering fixed portions of its given input. While this definition has a huge design space of possible algorithms, we can narrow it to a graph setting by borrowing from the distributed network literature [Lin87, Lin92].

For some $G = (V, E)$ where each node $v_i \in V$ is labeled $S = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, a local algorithm begins by randomly assigning a label to each vertex $X_{v_i} = \sigma_i$ and then updates v_i's label based on the labels of a small subset of V. This process occurs for some time t and concludes by outputting the final σ_is for each node.

A restricted version of this stochastic process (a random process that evolves over time) $\{X(t)\}_t$ was used in the obstruction analysis found [GS13, CGPR19] to capture most classical local algorithms while simplifying the allowed algorithm design. Here the initial label $X(0)_{v_i}$ assigned to each $v_i \in V$ is sampled from for the same distribution as an i.i.d (independent and identically distributed) random variable. The update step is also restricted to a fixed radius p neighborhood surrounding each v_i:

$$B(v_i, p) = \{v_k \in V(G) \mid \text{dist}(v_i, v_j) \leq p\} \tag{4.2}$$

Where $\text{dist}(v_i, v_j)$ is graph distance, i.e. the length of the shortest path in the graph between v_i and v_j. Finally, the update procedure for each v_i is limited to the same function.

4.1.1.2 Generic Local Algorithms

A hypergraph $G = (V, E)$ where V is a set of vertices and E is a collection of non-empty subsets of V, each subset representing a hyperedge. A hypergraph is a generalization of a graph where a single hyperedge can connect any number of vertices.

An algorithm A on some hypergraph G is local to depth p if, when assigning a label $A(G)_v \in S$, the following always holds [CLSS22a]:

1. For every set of vertices $L \in V$, the joint marginal distribution of the labels $(A(G)_v)_{v \in L}$ depends only on the union of the $p-$neighborhoods of $v \in L$ in G.

2. $A(G)_v$ is statistically independent of the joint distribution of $\{A(G)_{v'}\}$ for every v'

that is farther than a distance of $2p$ from v.

This generalization allowed the concentration results [CGPR19] to be extended to the QAOA. Finally [BGMZ22] extend the interpolation analysis to Max-q-XORSAT and mixed spin models in the infinite size limit to show an obstruction for any constant depth.

4.2 Non-Local Mixers

The methods used to present the OGP-based obstructions in [FGG20, CLSS22a] leverage the locality of an algorithm to demonstrate average-case concentration of expectations/approximations. We investigated the transition from local to non-local mixers in quantum circuits, examining if non-locality could provide a means of circumventing the Overlap Gap Property (OGP) obstructions. Our focus was to design and assess whether employing non-local mixers can effectively represent long-range interactions and mitigate these obstructions. The specific non-local mixers explored in the following sections are designed to probe varying forms of non-locality and each reflecting differing degrees of instance-specific information within their composition.

4.2.1 $X \otimes X$ Mixers

The QAOA's transverse field mixer is 1-local with each term acting on a single qubit: $M_i = I_0 \otimes I_1 \otimes \cdots \otimes X_i \otimes I_{i+1} \otimes \ldots I_{n-1}$ This structure can be modified by replacing the 1-local operation with a 2-local $X \otimes X$ mixing primitive:

$$M_{ij} = I_0 \otimes I_1 \otimes \ldots X_i \otimes I_{i+1} \otimes \ldots X_j \otimes \ldots I_{n-1} \tag{4.3}$$

The $X \otimes X$ mixing primitive permits the formulation of multiple mixers. For instance, the $X \otimes X$ operation can be applied to all pairs of nodes in the source graph:

$$M_I = \sum_{(i,j) \in N \times N} X_i \otimes X_j \tag{4.4}$$

It can be applied to consecutive nodes in a total node order:

$$M_{II} = \sum_{i \in n} X_i \otimes X_{i+1} \tag{4.5}$$

Or to nodes that are adjacent in the source graph:

$$M_{III} = \sum_{(i,j) \in E} X_i \otimes X_j \tag{4.6}$$

In each case, a problem instance renders a specific $X \otimes X$-type mixer instance. This is in contrast with the transverse field mixer, which remains constant for all problem instances of the same graph size.

To evaluate the $X \otimes X$ mixers, the graph-adjacency mixer (4.6) was then tested against the transverse field mixer on 100 randomly generated 14-node 3-regular graphs at $p = 1$. A summary of the results is provided below in figure 4.4. The data suggests that the standard $QAOA$ demonstrates a marginal edge over the $X \otimes X$ mixer variant at $p = 1$. The largest predicted cut size difference is one edge, but most results (60) show no difference in performance. Both methods display a comparable spread in results, indicating a similar distribution of outcomes across the sampled graphs. The extremal outcomes—those at the distribution's tails—show that neither method consistently outperforms the other for any of the samples in the ensemble. These observations suggest that while the standard QAOA likely offers a slight advantage, the overall efficacy of the two approaches is broadly

comparative at the tested depth.

4.2.2 Graph-Distance Mixers

As shown in section 2.4.2, the Max-Cut *graph* locality (vs. operator locality) of the QAOA with the transverse-field mixer is directly tied to the depth p. That, however, need not be the case. We can capture graph structure by classically pre-processing the input graph G. Specifically, we begin with graph distance:

$$d : V \times V \to \mathbb{N}_0 \tag{4.7}$$

where $d(i, j) = \infty$ if i and j are disconnected, $d(i, i) = 0$, and $d(i, j) = |P_{min}^{(i,j)}|$. Here $P_{min}^{(i,j)}$ is the shortest path from i to j. If we take the 2-local $X \otimes X$ mixing primitive defined in equation 4.3 and expand the adjacency mixer 4.6 with graph distance:

$$M_{IV} = \sum_{d(i,j) \bmod 2 \equiv 1} X_i \otimes X_j \tag{4.8}$$

Since $d(i, j)$ of adjacent nodes is one, we've expanded the total number of $X \otimes X$ terms to include more vertices that are an odd distance away from some node i and are more likely to be in the opposing partition.

For intuition, in a 3-regular tree (which is bipartite) all vertices at odd distance from a root lie in the opposite partition; high-girth regions of random 3-regular graphs approximately inherit this behavior (cf. Fig. 4.1). Otherwise, the graph would not expand outward. Here, if we pick any node and look at nodes an odd number of steps away, they tend to be on the opposite partition in an optimal cut. This idea borrows from how such nodes are always in the opposite partition in a bipartite graph and trees are always bipartite.

We are now also considering interactions between vertices separated by distances

up to the graph diameter, $\text{diam}(G) = \max_{u,v \in V} d(u,v)$. This can be refined by observing that nodes farther (in terms of graph distance) from a given node i have less influence. To incorporate this observation we first introduce the eccentricity of a node u:

$$e(u) = \max_{v \in V} d(u,v) \tag{4.9}$$

which is the furthest distance between u and any other node in G. For each pair of u and v, we can use eccentricity as an "influence scaling factor", and divide by the minimum of the two $w_{(u,v)} = \min(e(u), e(v))$:

$$M_V = \sum_{d(i,j) \bmod 2 \equiv 1} \frac{(w_{(i,j)} + 1) - d(i,j)}{w_{(i,j)}} X_i \otimes X_j \tag{4.10}$$

While M_{IV} and M_V incorporate longer-range (as measured in graph distance) interactions, they are still sums of 2-local $X \otimes X$ primitives, and as such, are variant of the $X \otimes X$ mixer. This means they are still local in the operator sense.

M_V is not the only way to incorporate the perspective that nodes an odd graph distance apart should be in the opposing partition:

$$M_{VIII} = \sum_{u \in V} \sum_{\substack{i=1 \\ i \bmod 2 \equiv 1}}^{e(u)} X_u \bigotimes_{v \in V_{d(u,v)=i}} X_v \tag{4.11}$$

Here we take all nodes an odd distance away from a given node u and create a single term for that neighborhood. The locality of the largest such tensor product term ($X_u \otimes X_{v_1} \otimes X_{v_2} ... X_{v_m}$) is now no longer fixed, as is the case with M_{IV} and M_V, but instead is a function of both graph size and diameter ($O(n)$).

As was the case with $M_{IV} \to M_V$ we can incorporate some eccentricity-based saliency scaling. For clarity, let e_u^i be the maximum value of $\frac{(e_{(i,j)}+1)-d(i,j)}{e_{(i,j)}}$ for all nodes

distance i from u:

$$M_{VIII} = \sum_{u \in V} \sum_{\substack{i=1 \\ i \bmod 2 \equiv 1}}^{e(u)} e_u^i X_u \bigotimes_{v \in V_{d(u,v)=i}} X_v \tag{4.12}$$

4.2.3 Graph Traversal Mixer

For an unweighted simple graph, variants of the breadth-first search (BFS) algorithm can be used to compute (1) graph distance from some source node of interest and (2) detect if a graph is bipartite. Since both procedures are based on BFS, they proceed in much the same way. Firstly by selecting some node $v \in V$ in the graph and annotating it with a 0. For graph distance, select the source node, for bipartite detection, any random node will work. Then consider v's adjacency $u_i \in adj(v)$, for each u_i, check if it is annotated. For distance, the annotation is simply a number, for bipartite detection it is a partition ($\{0, 1\}$). If it is unassigned, assign the incremented distance or assign the opposing partition. For each u_i that was not annotated at the time of processing, consider the nodes in $adj(u_i)$ and repeat the procedure. For graph distance, simply return the annotated graph after processing. For the bipartite detection, if at any point an already annotated node is revisited and the partition assignments clash, terminate and signal frustrated, otherwise after the full graph is processed signal bipartite.

If we consider how the support for a specific edge grows with p when running the QAOA we see a similar expansion occur. The key difference being that the expansion spreads outward from both nodes of some initial edge $e_i = (v, u)$ instead of a single starting node. We can create a variant of BFS that captures QAOA depth instead of graph distance by starting from an edge instead of a single node:

$$adj(v, u) = adj(v) \cup adj(u) - (v, u) \tag{4.13}$$

All subsequent calls to $adj(\cdot)$ revert to the single node version outlined at the start of this section. The modification no longer strictly captures graph distance, but instead computes the distance from the initial (v, u) edge, which is the same as the QAOA depth.

We can make a similar adjustment to the bipartite detection algorithm. Simply place the two nodes of the starting edge into opposing partitions and run the bipartite algorithm with 4.13.

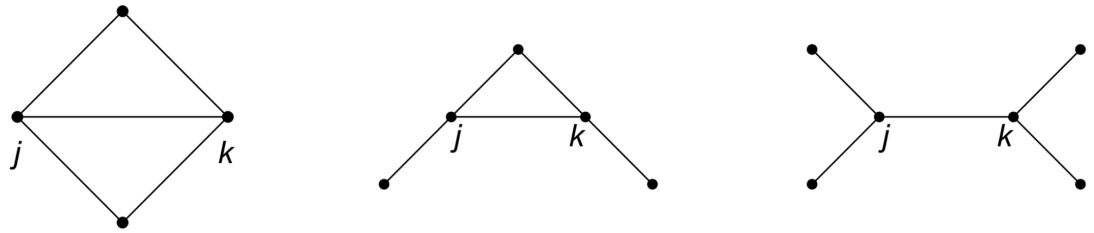

Figure 4.1: From [FGG14], for $p = 1$ the QAOA can only see the following 3 sub-graphs when processing a 3 regular graph.

To see why this might be useful in a non-bipartite graph, consider a random non-bipartite 3-regular graph $G = (V, E)$. In [FGG14] the authors show that, at $p = 1$ for the standard QAOA, all 3-regular graphs are decomposed into the 3 subgraphs shown above in figure 4.1. If we select a node from the interior of G, it is more likely to be part of the third (bipartite) subgraph than a node selected at the periphery. Said differently, for a large graph, the bipartite-detection traversal discussed above will complete more iterations before terminating for nodes in interior of a graph.

The traversal described here, up to a terminal depth t, captures a bipartite region rooted at the first node and grows like the support of the QAOA ansatz as p increases. The bipartition symmetry of the graph translates to a Z symmetry. Therefore, for some bit-string $p_i \in \{0, 1\}^t$ that represents a partitioning of a region of the graph, the operator that swaps between the two degenerate configurations (flips all the bits in the bit-string

describing the partition) is:

$$B_i = |p_i\rangle \langle \bar{p}_i| + |\bar{p}_i\rangle \langle p_i| \tag{4.14}$$

During the traversal a node that cannot be assigned a partition terminates the traversal. The final assignment of this terminal node v_t influences the final assignment of the two nodes in the initial $e_i = (v, u)$ edge. This is because the assignment of the terminal node affects which of the two projections in 4.14 is the more optimal state.

This terminal detection traversal information can be encoded into the mixer associated with this graph. One way of doing this is to use an $X \otimes X$ term for each pair (v, v_t and u, v_t). Alternatively, $X \otimes X \otimes X$ applied to all three captures more traversal data. Specifically that v_t is only terminal for a traversal that begins with the edge e_i, as opposed to a traversal that begins exclusively on either u or v.

The resulting $X \otimes X$ and $X \otimes X \otimes X$ primitive mixers are much like the other mixers detailed in previously in 4.2.2 and 4.2.1. The traversal process that generated these frustration-capturing terms also generated the per-edge Z flipping terms described in 4.14. These terms can be used instead of the tensor product of X terms:

$$M_{VII} = \sum_{e_i \in E} B_{e_i, v_t^i} \otimes X_{v_t^i} \tag{4.15}$$

Here v_t^i is the terminating vertex of the traversal that began at edge e_i, We can extend this style of traversal mixer by refining the termination condition of the traversal. Instead of immediately exiting upon reaching a frustrated node, we can define a radius r of nodes surrounding (in graph distance) the frustrated node and not process (traverse the adjacency) of nodes in this undefined region. This will extend the size of each B_{e_i} and add more terminal node terms $X_{v_{t_1}^i}, X_{v_{t_2}^i} \ldots$ terms

$$M_{VIII} = \sum_{e_i \in E} B_{e_i} \otimes X_{v^i_{t_1}} \otimes X_{v^i_{t_2}} \dots \tag{4.16}$$

Here the $v^i_{t_1}$ to $v^i_{t_k}$ nodes influence on the assignment of not only the source edge e_i but the unfrustrated region surrounding it.

4.3 Ballistic Optimization

As described in 2.3.5 The exponential map, applied to a fixed $H' \in \mathfrak{u}(2^n)$ generates a one-parameter ($\gamma(t) = U_{Ht}$) subgroup of the larger $U(2^n)$ group. Thus selecting an H amounts to selecting a tangent, and selecting a t is selecting a point along the path defined by that initial tangent. If we choose $H = \gamma H_C + \beta H_B$ as the joint generator, the resulting one-step evolution is $U_{\text{ballistic}} \approx e^{-iH}$—an approximation to the exact time evolution under H (rather than a Trotter product) under $H_C + H_B$ rather than a Trotter product.

$$U_{\text{ballistic}} = e^{-i(H_C\gamma + H_B\beta)} \tag{4.17}$$

Where H_C is the equation 2.24 and H_B is equation 2.18 and (γ, β) are the parameters.

Note that H_C and H_B do not commute. Therefore, while the alternating structure of the QAOA ansatz is unbiased in so much as it recovers the optimal solution in the large $p \to \infty$ limit, it is an approximation of unitary $U_{\text{ballistic}}$ in equation 4.17 for any fixed value of p.

Furthermore, $H = \gamma H_C + \beta H_B$ is 2-local (thus $O(1)$-sparse in the computational basis), so we approximate $U_{\text{ballistic}} = e^{-iH}$ to error ε using state-of-the-art sparse-Hamiltonian simulation (e.g., QSVT/LCU) [LC19, BCS$^+$20]. This changes the objective in Eq. 2.20 to

the ballistic evolution (with $U \approx e^{-iH}$ to precision ε):

$$f_{opt} = \min_{\boldsymbol{\gamma},\boldsymbol{\beta}} \langle +| \, e^{i(H_C\gamma + H_B\beta)} \, H_C \, e^{-i(H_C\gamma + H_B\beta)} \, |+\rangle \tag{4.18}$$

Or more generally when given a set of tangents $H_j \in \boldsymbol{H}$ and an objective O:

$$f_{opt} = \min_{\boldsymbol{t},\boldsymbol{H}} \langle +| \, e^{iH_j t} \, O \, e^{-iH_j t} \, |+\rangle \tag{4.19}$$

Qualitatively, Ballistic Optimization, compared to QAOA, exchanges the tuneable depth trotterization for a larger single optimization pass. This tradeoff makes it less suitable for current NISQ devices as compared to QAOA at $p = 1$ but also upper bounds gate complexity to a polynomial factor of the sparsity. We will analyze this method numerically here, and in detail in Chapter 5.

validate this is correct:

4.3.1 Implementing Ballistic Optimization

Let $H_X = \sum_i X_i$ and $H_{ZZ} = \sum_{(i,j)\in E} J_{ij} Z_i Z_j$. Let $\beta, \gamma \in \mathbb{R}$. Let $U_{\text{ballistic}}(t) = \exp\big(-it\,[\,\beta H_X + \gamma H_{ZZ}\,]\big)$. Fix an integer $p \geq 1$ and set $\Delta = t/p$. Let $\|\cdot\|$ denote the operator norm. The second order symmetric product is

$$S_2(\Delta) \;=\; \exp\big(-i\gamma\tfrac{\Delta}{2} H_{ZZ}\big)\, \exp\big(-i\beta\Delta H_X\big)\, \exp\big(-i\gamma\tfrac{\Delta}{2} H_{ZZ}\big),$$

with local error $O(\Delta^3)$ and global error

$$\big\| S_2(\Delta)^p - e^{-it(\beta H_X + \gamma H_{ZZ})} \big\| \;=\; O\Big(\big\|[H_{ZZ},[H_{ZZ},H_X]]\big\| + \big\|[H_X,[H_X,H_{ZZ}]]\big\| \Big)\, \frac{t^3}{p^2},$$

under standard BCH estimates for symmetric splittings [Tro59, Str68]. A symmetric composition of order four is obtained by

$$S_4(\Delta) \;=\; S_2(x_1\Delta)\, S_2(x_0\Delta)\, S_2(x_1\Delta),$$

where

$$x_1 = \frac{1}{2 - 2^{1/3}}, \qquad x_0 = -\frac{2^{1/3}}{2 - 2^{1/3}}, \qquad 2x_1 + x_0 = 1,$$

with local error $O(\Delta^5)$ and global error $O(t^5/p^4)$ controlled by norms of degree five nested commutators [Suz90].

Write $C(\theta) = \exp(-i\gamma\theta H_{ZZ})$ and $M(\theta) = \exp(-i\beta\theta H_X)$. One slice of $S_4(\Delta)$ merges adjacent cost rotations to give seven exponentials

$$C\big(\tfrac{x_1\Delta}{2}\big)\, M(x_1\Delta)\, C\big(\tfrac{x_0+x_1}{2}\Delta\big)\, M(x_0\Delta)\, C\big(\tfrac{x_0+x_1}{2}\Delta\big)\, M(x_1\Delta)\, C\big(\tfrac{x_1\Delta}{2}\big).$$

Implement $U_{\text{ballistic}}(t)$ by $\big[S_4(\Delta)\big]^p$. Per slice angles are

$$\begin{aligned}\big(\gamma_{r,1}, \beta_{r,1}, \gamma_{r,2}, \beta_{r,2}, \gamma_{r,3}, \beta_{r,3}, \gamma_{r,4}\big) = \Big(&\tfrac{x_1\gamma\Delta}{2},\, x_1\beta\Delta,\, \tfrac{(x_0+x_1)\gamma\Delta}{2},\, x_0\beta\Delta,\\ &\tfrac{(x_0+x_1)\gamma\Delta}{2},\, x_1\beta\Delta,\, \tfrac{x_1\gamma\Delta}{2}\Big), \quad r = 1, \ldots, p \quad (4.20)\end{aligned}$$

which satisfy $\sum_r(\gamma_{r,1} + \gamma_{r,2} + \gamma_{r,3} + \gamma_{r,4}) = \gamma t$ and $\sum_r(\beta_{r,1} + \beta_{r,2} + \beta_{r,3}) = \beta t$.

Counting by sublayers inside a single slice yields four cost layers and three mixer layers. Across p slices, adjacent cost layers at slice boundaries merge, producing a total of $6p + 1$ alternating layers, with $3p$ mixer layers.

Under the pair convention where one QAOA depth unit is $e^{-i\gamma_j H_{ZZ}} e^{-i\beta_j H_X}$, the stack corresponds to $3p$ pairs and would expose $6p$ free angles if treated as unconstrained QAOA. A slice wise ballistic parameterization uses at most $2p$ scalars $\{(\beta_r, \gamma_r)\}_{r=1}^p$, and a

fully tied ballistic parameterization uses two.

4.4 Methodology

To quantify the non-local approximate optimization described in sections 4.2 4.3, we undertake a numerical study on an ensembles of randomly generated (as generated by networkx and detailed in [SN99, JV03]) graphs. We compare the performance of the new mixers and optimization method to the standard QAOA at $p = 1$. We also compare results agains the classical Goemans-Williamson[GW95] semi-definite programming approximation algorithm for Max-Cut. The following sections detail the sample generation, testing procedure, and implementation details.

Execution of all tests occurred on the Tufts University Slurm Cluster, utilizing x86-64 hardware and Linux. For these tests, Qiskit[Qis23] and NetworkX[Net23] operated within a Python environment.

4.4.1 Classical Algorithms

We incorporated benchmarks for two classical algorithms: an off-the-shelf Goemans-Williamson algorithm implementation [GW95] sourced from the Rigetti QuantumFlow project [Cro18], alongside a naive brute-force max-cut algorithm that evaluates all $O(2^n)$ bit strings of length n.

Algorithm 3 Brute Force Max-Cut

```
1:  function MAXCUT(G)
2:      n ← |G|
3:      L_G ← LAPLACIAN(G)
4:      C_max ← 0
5:      for all i ∈ 2^{n−1} + 1 do
6:          b ← TOBITARRAY(i, n)
7:          b ← 2b − 1
8:          C_i ← SCORECUT(b, L_G)
9:          if C_i > C_max then
10:             C_max ← C_i
11:     return C_max
```

Figure 4.2: Pseudocode for the brute-force algorithm implemented (in the python programming language) to compute the actual Max-Cut. Line 3 above compute the laplacian matrix ($L_G = D - A$) for the input graph G. Line 6 assign b the binary representation of the number i. Line 7 maps $b^{\{0,1\}} \rightarrow b^{\{-1,1\}}$. Line 8 uses the laplacian to score the cut ($b^T L_G b$).

4.4.2 Non-Local Mixers

To evaluate the new mixers detailed in 4.2, the test suite generated 100 3-regular graphs of 14 nodes each using NetworkX's `random_regular_graph` function from `networkx.generators.random_graphs` module. Quantum circuits for the single-qubit X, two-qubit $X \otimes X$, and Sign mixers utilized RXX gates to generate explicit circuits. Direct exponentiation of the Hamiltonians for structured mixers employed Qiskit's Hamiltonian exponentiation function. Simulation of quantum circuits used Qiskit's statevector simulator. Parameter optimization relied on the COBYLA optimizer from SciPy's `minimize` function. The Python interpreter

version 3.8 executed all procedures. Like the transverse field mixer, all the mixer developed in 4.2 can grow with depth. We tested at $p = 1$" Computations were carried out on the Tufts University SLURM cluster, utilizing either 8 or 16 cores with 128GB RAM, and each ensemble took approximately one week to complete.

4.4.3 Ballistic Optimization

When testing Ballistic Optimization we used the standard transverse field mixer. Unlike the random non-local mixer ensemble, the test suite here encompassed all non-isomorphic 3-regular graphs ranging in size from $n = 6$ to $n = 12$. For $n = 14$, we also included all bipartite graphs. We randomly generated 10 million 3-regular graphs for sizes $n = 6$ to $n = 12$. Then we checked if each sample was bipartite. To maintain sample uniqueness within the testing ensemble, we verified that no newly generated graph was isomorphic to any existing sample. These check were done using built in method's found in NetworkX,

	6	**8**	**10**	**12**	**14**
bipartite	1	1	2	5	12
frustrated	1	4	17	80	0

Figure 4.3: Tabular distribution of 3-regular non-isomorphic graphs in the Ballistic Optimization ensemble. Columns are indexed by graph size and rows by graph type, specifically whether they are bipartite. This ensemble was created by randomly generating 10 million 3-regular of size $n = 6, 8, 10, 12$ and testing for isomorphism and bipartite-ness.

In contrast to the non-local mixer tests, the Ballistic Optimization tests computed the expectation value directly. Construction of the Hamiltonians utilized Qiskit's operator flow (opflow) module. Direct exponentiation of these Hamiltonians preceded expectation value calculation via Qiskit's `eval` function. For optimization, the trial employed SciPy's `brute` function, operating on a 1000-point grid. Refinement followed within the bounds of

the chosen grid subregion. Execution of all trials occurred under Python version 3.8.

4.5 Results

This section gives the outcomes of the numerical simulations described in section 4.4. Following this, focus shifts to the Ballistic Optimization methodology, as detailed in 4.3. Broadly, modifying $QAOA_1$ with the various mixers outlined in this paper, showed no improvement in the average approximation ratio for the ensemble laid out in 4.4.2. Conversely, Ballistic Optimization consistently outperformed $QAOA_1$ for the ensemble referenced in 4.4.3. Ballistic Optimization also outperformed the worst Goemans-Williamson approximations for most frustrated sample.

4.5.1 $X \otimes X$ Mixers

The adjacency mixer (4.6) predicts, on average, a quarter of an edge fewer than the standard QAOA. The distribution measures remain comparable as per figure 4.4.

4.5.2 Graph Traversal Mixer

The graph traversal mixer performs on average one edge worse than the standard transverse field mixer (figure 4.4). However in exchange, the spread has shrunk from the three edges to a single edge and the quality for the worst prediction has increased slightly by a quarter of an edge.

4.5.3 Ballistic Optimization

Figure 4.7 shows that Ballistic Optimization significantly outperforms $QAOA_1$ for all samples outlined in 4.4.3. Figure 4.7 show that for overwhelming majority of non-bipartite

samples, Ballistic Optimization also outperforms the worst cut obtained from the Goemans-Williamson but fails to outperform the best cut. For the bipartite samples, the Goemans-Williamson recovers the optimal cut (figure 4.6). Broadly, the bipartite performance of Ballistic Optimization degrades as graph size increases. However, the reasons for the degradation are unclear. One possible explanation is a failure in optimization. During initial testing, a parameter search procedure similar to the one outlined in section 4.4.2 was used, but performing a single COBYLA minimization would often fail to select optimal parameters without multiple attempts at the minimization. We instead switched to a 1000-point brute-force grid search to stabilize the optimization. The decision to select 1000 points was a rough heuristic, and as graph size increases, the granularity may not been fine enough to approach an optimal value. These results are more positive than those for the other nonlocal mixers considered in this Chapter. This preliminary analysis motivates the more detailed treatment of ballistic optimization given in Chapter 5.

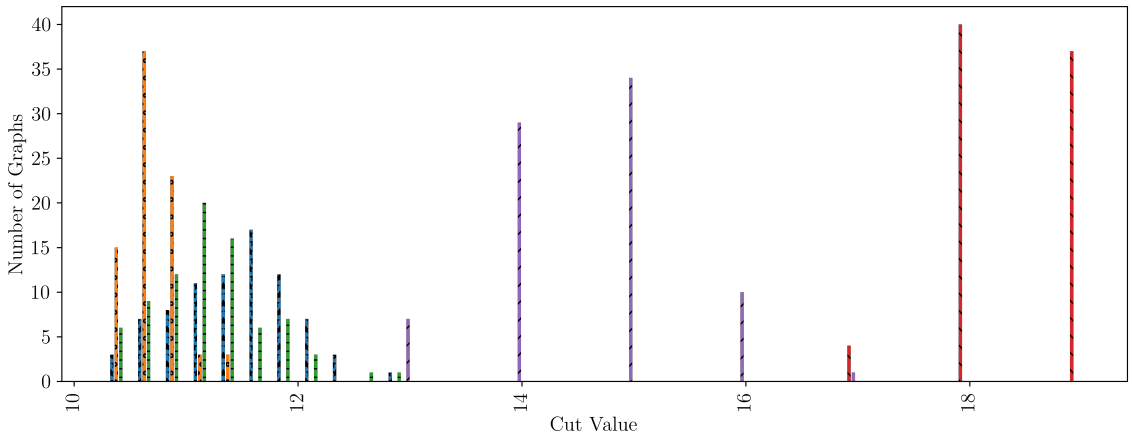

Figure 4.4: Comparison of cut sizes for Max-Cut (§2.2.2). A legend in eac panel maps markers/colors to Graph-Traversal mixer, $QAOA_{p=1}$, adjacency $X \otimes X$ mixer, Goemans–Williamson, and brute force. Graph Traversal 4.2.3 mixer (orange 'o'), *QAOA* 2.4.1 (blue '*'), adjacency $X \otimes X$ 4.2.1 mixer (green '-'), Goemans-Williamson 4.4.1 (purple '/') and brute force Max-Cut 2.2.2 (red '\'). The algorithms were run on 100 randomly generated 14-node, 3-regular graphs (as in 4.4.1). All QAOA based algorithms were run at $p = 1$. The x-axis is cut sizes in the number of edges. Overall none of the quantum algorithms neared the performance of the Goemans-Williamson. The standard QAOA slightly outperforms the adjacency $X \otimes X$ mixer with a maximum difference of one edge. It also outperformed the graph traversal mixer by an average nearly one edge. The graph traversal mixer exhibited the strongest prediction concentration.

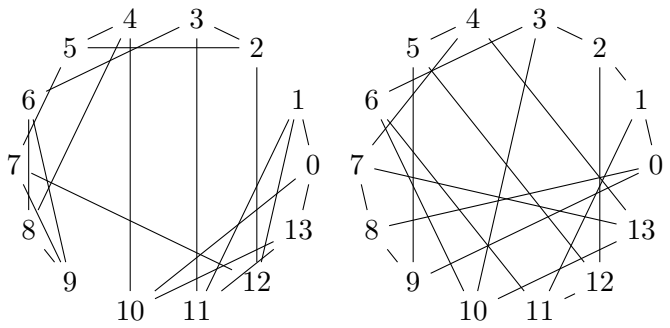

Figure 4.5: Two 14-node, 3-regular graphs from the ensemble of 100 randomly generated graph detailed in section 4.4.1. These graphs represent the extremes in expected cut sizes obtained with the adjacency $X \otimes X$ mixer (defined in 4.2.1). Despite their differing expected cut size, 10 and 13, respectively, both graphs share an actual max cut of 18 edges.

$\|G\|$	Max	$\mathbf{E}_{Ballistic}$	$\mathbf{E}_{QAOA}$	GW	$\mathbf{Apx-Ratio}_{Ballistic}$	$\mathbf{Apx-Ratio}_{QAOA}$
6	9	8.515	6.232	9	0.9461	0.6925
8	12	10.816	8.309	12	0.9014	0.6925
10	15	13.441	10.386	15	0.8961	0.6925
12	18	14.783	12.464	18	0.8213	0.6925
12	18	15.434	12.464	18	0.8574	0.6925
12	18	14.684	12.464	18	0.8158	0.6925
12	18	14.852	12.464	18	0.8251	0.6925
12	18	14.753	12.464	18	0.8196	0.6925

Figure 4.6: Table of comparative results for Max-Cut 2.2.2 optimization techniques across all non-isomorphic, bipartite 6, 8, 10, and 12-node, 3-regular graphs 4.4.3. The **Max** column is the exact max cut obtained from an exhaustive search (implementation detailed in 4.4.1). **GW** denotes the Goemans-Williamson algorithm 4.4.1. The $\mathbf{E_{Ballistic}}$ and $\mathbf{E_{QAOA}}$ are the expected cut sizes from Ballistic Optimization 4.3 and $QAOA_{p=1}$ 2.4.1, respectively. $\mathbf{Apx-Ratio}_{Ballistic}$ and $\mathbf{Apx-Ratio}_{QAOA}$ are the corresponding approximation ratios. Δ_{apx} is the performance gap between the two methods with positive value indicating Ballistic Optimization outperformning the QAOA. Results confirm the QAOA's worst-case lower-bound performance for bipartite graphs (from [FGG14]). Also, Ballistic Optimization exhibits a decline in approximation quality with increasing graph size while still outperforming the QAOA across the board.

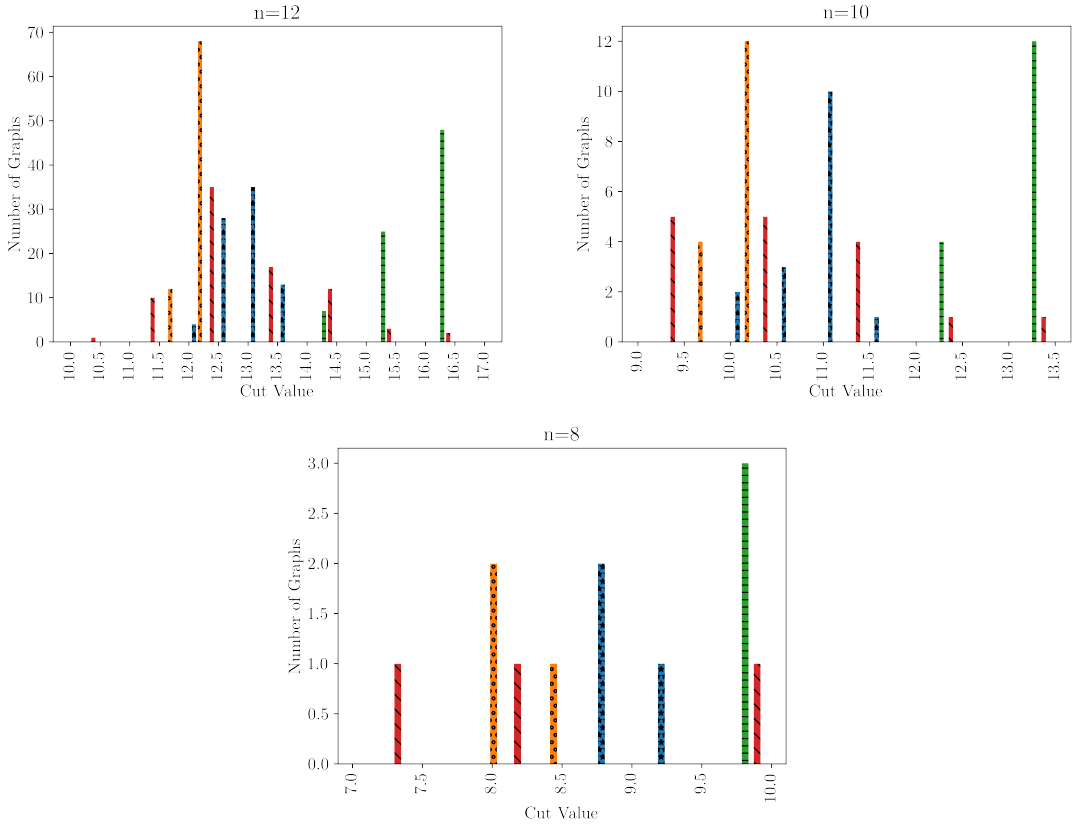

Figure 4.7: Comparison of cuts sizes for Max-Cut 2.2.2 from the following algorithms: Goemans-Williamson 4.4.1 (red '\'), $QAOA_{p=1}$ 2.4.1 (orange 'o'), Ballistic Optimization 4.3 (blue '*') and brute force Max-Cut 2.2.2 (red '\'). The algorithms were run on all non-isomorphoric, non-bipartite graphs 4.4.3 of $n = 12$ on the top-left, $n = 10$ on the top-right, and $n = 8$ on the bottom. The x-axis is cut sizes in the number of edges. Overall Ballistic Optimization always outperformed the $QAOA_{p=1}$ and largely outperforms the worst case Goemans-Williamson. However, for this ensemble, when the Goemans-Williamson algorithm is run 100 times, the best result mirrored the brute force Max-Cut and outperforms Ballistic Optimization. Furthermore as n increases the difference in performance between Ballistic Optimization and $QAOA_{p=1}$ lessens.

Chapter 5

Locality of Ballistic Optimization

Chapter 4 establishes Ballistic Optimization's empirical advantage over $QAOA_1$ but leaves the underlying mechanism unaddressed. This chapter provides an analytic explanation. Focusing on the "ring of disagrees" Max-Cut instance, we expand the continuous-time evolution in a Baker–Campbell–Hausdorff series, obtain closed-form even-order coefficients $c_{2q} = (8\beta\gamma)^q 2^{q-1}\mathrm{Cat}_q$, and show that operator support becomes $n-1$ at commutator order $k = 2N-2$. The analysis predicts a runtime condition for full reach, and simulations up to confirm these finding.

The preliminary numerical experiments reported in Chapter 4 of this dissertation show that Ballistic Optimization (BO) attains the highest mean approximation ratio among all quantum algorithms tested, exceeding every depth-$p = 1$ instance of the Quantum Approximate Optimization Algorithm (QAOA). In this Chapter we will investigate BO in more detail, focusing on the locality of this algorithm.

While structurally the QAOA and ballistic Hamiltonians are similar, unlike the alternating Trotterization of the QAOA, ballistic optimization is a direct, non-Trotterized continuous evolution under a time-independent Hamiltonian. To probe the origin of the

performance observed oin Chapter 4 we adopt the analytically tractable "ring of disagrees," i.e. the antiferromagnetic Max-Cut instance on the N-vertex cycle graph C_N [FGG14]. Farhi, Goldstone, and Gutmann analyzed the ring of disagree when introducing QAOA to derive closed-form depth-$p = 1$ approximation ratio, establishing it as a natural place for comparison to other variational quantum optimizers [FGG14]. For even N the cycle graph is bipartite and possesses full dihedral (rotations and reflections) symmetry, allowing every vertex (and edge) to be mapped to any other by an automorphism; this uniformity greatly simplifies the bookkeeping of operator growth compared with the 3-regular graphs examined in the preceding chapter [Die17].

Following Farhi *et al*'s analysis of QAOA, we analyze the dynamics of the objective function in the Heisenberg picture, in which an observable O evolves as $O(t) = e^{iHt}\, O\, e^{-iHt}$ while the quantum state remains fixed [JN17]. Hadamard's lemma gives an exact Taylor expansion of the ballistic evolution in nested commutators,

$$O(t) = \sum_{n=0}^{\infty} \frac{(it)^n}{n!}\, \mathrm{ad}_H^n(O), \qquad \mathrm{ad}_H^n(O) = \underbrace{[H, [H, \ldots [H, O]]]}_{n \text{ times}}, \tag{5.1}$$

which is the operator form of the Baker–Campbell–Hausdorff (BCH) series [Bri15]. Because the BCH series converges for all bounded Hamiltonians, it provides a non-perturbative and formally exact representation of the Heisenberg evolution; see Hall for a detailed proof of convergence criteria and coefficient structure [Bri15].

Each term $\mathrm{ad}_H^n(O)$ can be viewed as an ordered sequence of n local Hamiltonian terms acting on O, so the nested-commutator hierarchy explicitly records the "history" of the action of H on O and quantifies how operator support spreads across the lattice—a perspective widely used in modern studies of operator growth and Krylov complexity [ESS19].

This leads to our central result: a formula characterizing the operator spreading

of O which we validate against numerical simulation. Finally, we analyze the growth of operator locality within the expansion, revealing the precise scaling that dictate how quickly the evolution can generate system-spanning operators.

5.1 BCH Analysis

In this section we analyze the BCH expansion for ballistic optimization applied to the ring of disagrees. Throughout this section let

$$H(\beta,\gamma) = \beta\, H_X \;+\; \gamma\, H_{ZZ}, \qquad H_X \;=\; \sum_{j=1}^{N} \sigma_j^x, \qquad H_{ZZ} \;=\; \sum_{\{j,k\}\in E} \sigma_j^z \sigma_k^z \tag{5.2}$$

and fix the observable to be the cost reduced Hamiltonian $O := H_{ZZ}$. Nested commutators of H with O will be written $\mathrm{ad}_H^n(O) = [H, [H, \dots [H, O] \dots]]$ with n occurrences of H.

5.2 The Ring of Disagrees

Let $C_N = (V, E)$ be the N-vertex cycle graph, i.e. $V = \{1, \dots, N\}$ and $E = \{\{j, j+1\} \mid 1 \leq j < N\} \cup \{\{N, 1\}\}$. For each vertex $j \in V$ assign a binary decision variable $z_j \in \{-1, +1\}$. Define the disagree objective:

$$\mathrm{DIS}(z) = \sum_{\{j,k\}\in E} \frac{1 - z_j z_k}{2} \tag{5.3}$$

where $z = (z_1, \dots, z_N) \in \{-1, +1\}^N$. Equation (5.3) counts the number of edges whose endpoints *disagree* in sign; equivalently, it is the MaxCut objective on C_N. The ring of disagree problem is a maximization of the objective:

$$\text{maximize}\, z \in \{-1, +1\}^N\, \mathrm{DIS}(z) \tag{5.4}$$

Because C_N is bipartite when N is even, the optimum of (5.4) is N. If N is odd, any assignment must leave at least one edge agreeing, so the optimum is $N - 1$.

We can map each classical variable z_j to a qubit via $z_j \mapsto \sigma_j^z$, where σ_j^z is a Pauli-Z operator acting on qubit j. The objective (5.3) becomes the diagonal Hamiltonian

$$H_{\text{cost}} = \sum_{\{j,k\}\in E} \frac{1 - \sigma_j^z \sigma_k^z}{2}. \tag{5.5}$$

For optimization purposes the additive constant $|E|/2 = N/2$ (the number of nodes and edges is the same) may be dropped because adding or subtracting a scalar multiple of the identity leaves every eigenvector unchanged, it has no effect on which bit-strings minimize the energy. This yields the problem Hamiltonian:

$$H_{ZZ} = \sum_{\{j,k\}\in E} -\sigma_j^z \sigma_k^z. \tag{5.6}$$

Minimizing H_{ZZ} is equivalent (results in the same bit string) to maximizing (5.3).

5.3 Baker–Campbell–Hausdorff Expansion

Let $|+\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)$ be the single-qubit plus state and let $|+\rangle^{\otimes N}$ denote its N-qubit tensor power. For a time-independent Hamiltonian H and an observable O define the Heisenberg evolution

$$O(t) = e^{iHt} O e^{-iHt} \tag{5.7}$$

The Hadamard Lemma and the Baker–Campbell–Hausdorff theorem expresses the expectation value of $O(t)$ in state $|\psi\rangle$ as a power series of expectations of nested commutators

[Cam97, Hau06]

$$\langle O(t)\rangle = \sum_{n=0}^{\infty} \frac{(it)^n}{n!} \langle [H,O]^{(n)}\rangle, \quad [H,O]^{(0)} = O,\ [H,O]^{(n)} = [H,[H,O]^{(n-1)}] \tag{5.8}$$

Taking the expectation value in the state $|+\rangle^{\otimes N}$ yields

$$\langle O\rangle_+(t) = \langle +|^{\otimes N} O(t) |+\rangle^{\otimes N} = \sum_{n=0}^{\infty} \frac{(it)^n}{n!} \langle +|^{\otimes N} [H,O]^{(n)} |+\rangle^{\otimes N} \tag{5.9}$$

5.4 Truncating the BCH Expansion

The BCH series shows that there are contributions to the expectation value at all orders. The order k nested commutator will be an $O(k)$-local Pauli operator. The coefficients of the Pauli operators in this series at order k will determine which terms dominate the behavior of the algorithm. At very large k, the denominator will dominate, and the terms will fall rapidly with k. At smaller k the numerator will dominate, and the terms will grow. The trade-off between these two behaviors will determine the values of k that contribute most to the expectation value, and this will determine the effective locality of the ballistic algorithm.

In the following subsections we will bound the locality of terms in eq. 5.8. In Subsection 5.4 we consider some bounds that arise from the norm of the terms. In Subsection 5.5 we consider which terms contribute to the expectation value in the state $|+\rangle^n$, and how these terms arise at each order k, and develop our first estimate of the expectation value. In Subsection 5.5 we refine this estimate, and compare both these estimates to numerical calculations for $k < 30$.

Let H be a bounded Hamiltonian with operator norm $|H|$, let O be a bounded observable, and let $t \in \mathbb{R}$ be the rescaled evolution time. For every non-negative integer n

define

$$\mathcal{A}_n(t) = \frac{|t|^n}{n!} \,|\operatorname{ad}_H^n(O)|, \quad \operatorname{ad}_H^n(O) = [H, [H, \ldots, [H, O]\ldots]]. \tag{5.10}$$

The function $\mathcal{A}_n(t)$ gives the operator norm of the nth term in the series. Next we develop several bounds on $\mathcal{A}_n(t)$.

First we use the inequality $|[A, B]| \leq 2|A|\,|B|$ to obtain an upper bound on $\mathcal{A}_n(t)$. Applying the inequality n times gives:

$$|\operatorname{ad}_H^n(O)| \leq (2|H|)^n\,|O| \quad (n \geq 0). \tag{5.11}$$

Then define:

$$\mathcal{U}_n(t) := |O|\,\frac{(2|H|\,|t|)^n}{n!} \tag{5.12}$$

so $\mathcal{A}_n(t) \leq \mathcal{U}_n(t)$ for all n.

Given our upper bound $\mathcal{U}_n(t)$ we can ask at what value of n does $\mathcal{U}_n(t)$ begin to decrease? We apply a ratio test, defining for every $n \geq 0$,

$$R_n := \frac{\mathcal{U}_{n+1}(t)}{\mathcal{U}_n(t)} = \frac{2|H|\,|t|}{n+1}. \tag{5.13}$$

The inequality $R_n \leq 1$ holds whenever

$$n + 1 \geq 2|H|\,|t| \tag{5.14}$$

so the sequence $\big(\mathcal{U}_n(t)\big)_{n\geq 0}$ decreases for all $n \geq 2|H|\,|t| - 1$.

We may obtain an upper bound on $\mathcal{U}_n(t)$ from Stirling's inequality $n! \geq (n/e)^n$, which implies

$$\mathcal{U}_n(t)\mathcal{U}_n(t) \leq |O|\,\left(\frac{2e\,|H|\,|t|}{n}\right)^n \tag{5.15}$$

The right-hand side is at most $|O|$ provided

$$n \geq 2e\,|H|\,|t|. \tag{5.16}$$

This indicates that $\mathcal{U}_n(t)$ is upper bounded by $|O|$ when $n \geq 2e\,|H|\,|t|$, and the ratio test tells us that $n \geq 2e\,|H|\,|t|$ will already be decreasing at this value of n.

We now define a truncation index beyond which we will discard terms in the BCH expansion. In units where $|H| \leq 1$

$$k_{\min} := \lceil 2e\,|t| \rceil. \tag{5.17}$$

For every $n \geq k_{\min}$ the envelope terms satisfy $\mathcal{U}_n(t) \leq 1$ and form a non-increasing sequence.

5.5 Naive bound on coefficients for a single XI sector

Next we consider the effect on the sum of considering only the extecpation value in the state $|+\rangle^n$. Because $|+\rangle$ is an eigenstate of every σ^x with eigenvalue $+1$ and satisfies $\langle+|\sigma^y|+\rangle = \langle+|\sigma^z|+\rangle = 0$, only Pauli words composed solely of σ^x and identities contribute to the expectation (5.9). We can therefore write an expression for the expectation value reflecting this. Fix real parameters β and γ in the ballistic hamiltonian. For each even integer $k = 2q \geq 0$ write

$$\mathrm{ad}_H^k(O) \;=\; c_k \sum_{j=1}^{N} \sigma_j^x \;+\; \text{Pauli words containing } \sigma^y \text{ or } \sigma^z, \tag{5.18}$$

where $c_k \in \mathbb{R}$ is the coefficient of σ_j^x. Because all edges in the ring of disagrees are equivalent under translation, the coefficient is the same for every site j.

Closed form for the coefficients

To obtain our first estimate of the coefficients c_k appearing in eq. 5.18 we consider the sequences of nested commutators of H_{ZZ} and H_X that can generate a single σ_j^x term. We focus on a single site j, and compute the contributions to c_k that arise from sequences of commutators resulting in a single σ_j^x at that site. Proceeding order by order: At depth $k = 0$ no σ^x appears, so $c_0(j) = 0$. At first order the operator $\mathrm{ad}_H^1(O) = [H, O]$ contains one power of H_X and therefore one σ^x on some site j. This turns that σ_j^x into σ_j^y, so no σ^x-only word survives. Hence $c_1(j) = 0$. At second order $k = 2$ we have two nested commutators: $[H_X, [H_X, O]]$ and $[H_{ZZ}, [H_X, O]]$. The first nested commutator $[H_X, [H_X, O]]$ does not result in any single σ_j^x terms. The second commutator takes the $\sigma_j^y \sigma_{j+1}^z$ produced by the first H_X and the $\sigma_j^z \sigma_{j+1}^z$ term converts that back to a single σ_j^x. At third order no single $sigma_j^x$ terms can occur.

Terms at even order $k = 2q$ containing an initial sequence of an even number of H_X will result in a term proportional to the original objective function. One more H_X followed by a $H_Z Z$ will result in a single σ_j^x term. Subsequent H_X terms will leave this invariant, while an even number of $H_Z Z$ terms will repeatedly convert it to σ_j^y and back again. Amongst these sequences we select those with an equal number of H_X and H_{ZZ} terms. At order $k = 2q$ there are

$$\binom{2q}{q} \tag{5.19}$$

such sequences.

Every admissible ordering contributes the same absolute magnitude, because each of the $q + 1$ (H_X, H_{ZZ}) pairs yields the product $(2\beta)(2\gamma) = 4\beta\gamma$. Multiplying the induction value $\left|c_{2q}(j)\right| = (4\beta\gamma)^q \binom{2q}{q}$ by an additional factor $4\beta\gamma$ and by the ratio

$\frac{\binom{2q+2}{q+1}}{\binom{2q}{q}} = \frac{(2q+2)(2q+1)}{(q+1)^2}$ gives

$$\left|c_{2(q+1)}(j)\right| \;=\; (4\beta\gamma)^{q+1}\binom{2q+2}{q+1} \tag{5.20}$$

The final closed form estimate for the coefficients c_k derived in this section is

$$\boxed{\left|c_{2q}(j)\right| \;\simeq\; (4\beta\gamma)^{q}\binom{2q}{q}, \qquad c_{2q+1}(j)=0} \quad \text{for every } j \in V. \tag{5.21}$$

We can estimate the asymptotic growth of this estimate using Stirling's approximation [Ric12]

$$\binom{2q}{q} \sim 4^q/\sqrt{\pi q} \tag{5.22}$$

the magnitude in (5.21) behaves asymptotically as

$$\left|c_{2q}(j)\right| \;=\; \frac{(\,16\beta\gamma\,)^q}{\sqrt{\pi q}}\big(1+O(q^{-1})\big). \tag{5.23}$$

In the BCH series each term is multiplied by $|t|^{2q}/(2q)!$. The ratio of successive even terms therefore satisfies $r_q = \frac{16\,\beta\gamma\,t^2}{(q+1)^2}\,\big(1+O(q^{-1})\big)$. Setting $r_q = 1$ gives the *optimal truncation depth*

$$k_\star \;\approx\; 8\,\sqrt{\beta\gamma}\,|t|. \tag{5.24}$$

Thus, given our first estimate, the IX sector grows with the *central binomial coefficient* $\binom{2q}{q} \sim 4^q/\sqrt{\pi q}$ [Ric12]. Substituting (5.21) into (5.9) yields

$$\langle H_{ZZ}\rangle_+(t) = \sum_{q=0}^{\infty} \frac{(2it)^{2q}}{(2q)!}\,(4\beta\gamma)^q\binom{2q}{q},. \tag{5.25}$$

This estimate is compared with numerical calculations for $k < 30$ in Figure 5.1.

Two-edge refinement for the single–X coefficient

The "naive" single-edge derivation above implicitly assumed that every σ_j^z created by H_{ZZ} acted on the *same* edge incident to the marked qubit j. On a ring, however, qubit j has two neighbors, so each time a driver insertion is required it can be attached either to the left edge $(j-1, j)$ or to the right edge $(j, j+1)$. These choices are independent for the q driver–mixer pairs that make up an admissible word at depth $2q$. Replacing $(4\beta\gamma)^q$ by $(8\beta\gamma)^q$ throughout the large-q analysis of §5.5 gives

$$\left|c_{2q}^{(2)}(j)\right| \sim \frac{(\,32\beta\gamma\,)^q}{\sqrt{\pi q}}, \qquad r_q^{(2)} = \frac{64\,\beta\gamma\,t^2}{(q+1)^2}\big(1 + O(q^{-1})\big), \qquad k_\star^{(2)} \approx 16\,\sqrt{\beta\gamma}\,|t|. \tag{5.26}$$

Thus the two-edge correction preserves the square-root optimal depth while simply doubling the exponential base from 4 to 8 per driver–mixer pair. The improvement achieved by this estimate is shown in Figure 5.1 through comparison with numerical calculations for $k < 30$.

5.6 Catalan Refinement

Figure 5.1 contrasts the single–edge formula (5.21), its two–edge refinement (5.5) and exact numerics for a twelve-qubit ring. From $k = 4$ onward the two-edge curve lies strictly below the data. Table 5.1 shows exact numerical magnitudes for $|c_k(X_j)|$ with parameters $\beta = 1$, $\gamma = 1$. The formula derived in this section, equation (5.30), reproduces these values.

Table 5.1: Exact magnitudes $|c_k(X_j)|$ for $\beta = \gamma = t = 1$. These values are reproduced by the Catalan formula, equation (5.30).

k	0	2	4	6	8
$\lvert c_k(X_j)\rvert$	1	8	256	10 240	458 752

In the following subsections we will develop a refined estimate of the coefficients of

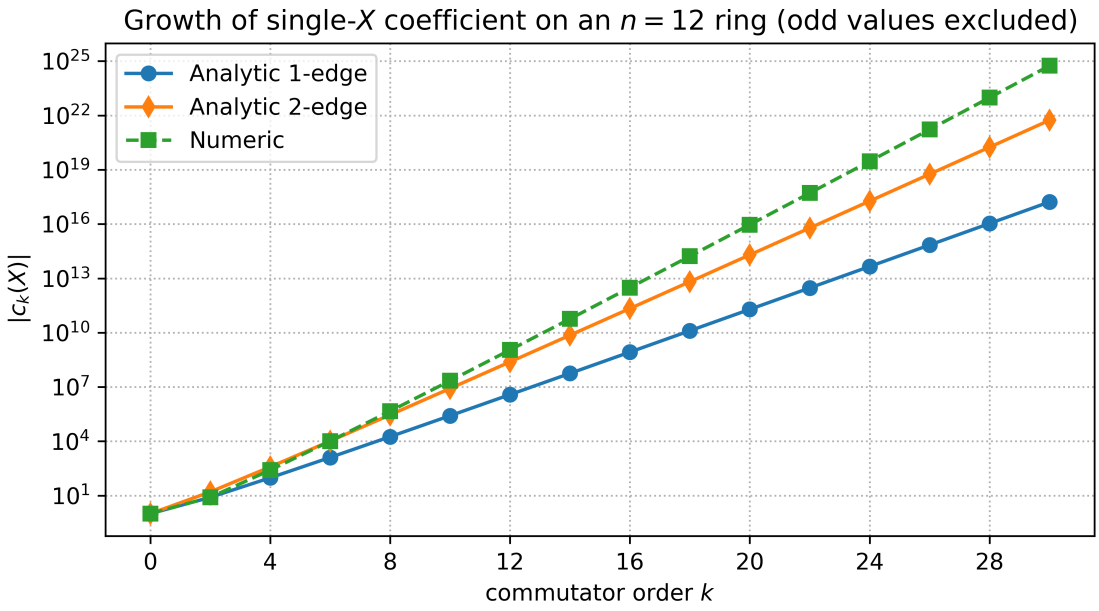

Figure 5.1: Growth of the single-σ_j^x coefficient, $|c_k(X_j)|$, on an $N = 12$ ring for $\beta = \gamma = 1$. Blue circles represent the single-edge prediction from equation (5.21). Orange diamonds represent the two-edge refinement from equation (5.5). Green squares represent exact numerical values. Odd k values are excluded as $c_k(X_j) = 0$. The Catalan expression, equation (5.30), matches the green squares.

terms that contribute to the expectation value.

5.6.0.1 Operator Sequences and Derivation for $k = 2$

The coefficient $c_k(X_j)$ represents the amplitude of the single σ_j^x term within the k-th order nested commutator term $(\mathrm{ad}_H)^k(O)$. For $k = 0$ the operator is $O = H_{ZZ}$ contains no σ_j^x terms, so the true coefficient $c_0(X_j)$ is 0.

For $k = 1$, the relevant term is $(\mathrm{ad}_H)^1(H_{ZZ}) = \beta[H_X, H_{ZZ}]$. A typical component of this commutator, for instance $\beta[\sigma_j^x, \sigma_j^z\sigma_{j'}^z]$, evaluates to $-2i\beta\sigma_j^y\sigma_{j'}^z$. This resulting operator contains σ_j^y but does not contain σ_j^x. Consequently, the coefficient $c_1(X_j)$ is 0.

For $k = 2$, corresponding to $q = 1$, we focus on terms within $(\mathrm{ad}_H)^2(H_{ZZ})$ that

originate from sequences involving one application of an operator from βH_X and one from γH_{ZZ} in the nested commutators. The relevant part of $(\mathrm{ad}_H)^2(H_{ZZ})$ is

$$[\beta H_X + \gamma H_{ZZ}, [\beta H_X, H_{ZZ}]] \supset \beta\gamma[H_{ZZ}, [H_X, H_{ZZ}]]. \tag{5.27}$$

The term $\beta\gamma[H_X, [H_{ZZ}, H_{ZZ}]]$ is zero because $[H_{ZZ}, H_{ZZ}] = 0$. To generate a σ_j^x operator, the process begins with an initial term from $O = H_{ZZ}$ that includes an operator on qubit j. Let this initial term be $O_{j,j'} = \sigma_j^z \sigma_{j'}^z$, where j' is a neighbor of j. The sequence of operations is as follows:

1. **Action from βH_X:** The first commutation involves selecting $\beta\sigma_j^x$ from H_X to act on $O_{j,j'}$. The resulting intermediate operator S_1 is

$$\begin{aligned} S_1 = [\beta\sigma_j^x, O_{j,j'}] &= \beta[\sigma_j^x, \sigma_j^z\sigma_{j'}^z] \\ &= \beta([\sigma_j^x, \sigma_j^z])\sigma_{j'}^z \\ &= \beta(-2i\sigma_j^y)\sigma_{j'}^z = -2i\beta\sigma_j^y\sigma_{j'}^z. \end{aligned}$$

   This operation transforms the σ_j^z component of $O_{j,j'}$ into a σ_j^y component.

2. **Action from γH_{ZZ}:** The second commutation involves selecting a term from H_{ZZ} to act on S_1. To transform σ_j^y to σ_j^x, this H_{ZZ} term must also involve an operator on qubit j. We choose the term $\gamma O_{j,j'} = \gamma\sigma_j^z\sigma_{j'}^z$. The resulting state S_2 is

$$\begin{aligned} S_2 = [\gamma O_{j,j'}, S_1] &= [\gamma\sigma_j^z\sigma_{j'}^z, -2i\beta\sigma_j^y\sigma_{j'}^z] \\ &= -2i\beta\gamma[\sigma_j^z\sigma_{j'}^z, \sigma_j^y\sigma_{j'}^z] \end{aligned}$$

Because $(\sigma^z_{j'})^2 = I$, where I is the identity operator, this simplifies to

$$\begin{aligned} S_2 &= -2i\beta\gamma[\sigma^z_j, \sigma^y_j] \\ &= -2i\beta\gamma(-2i\sigma^x_j) = -4\beta\gamma\sigma^x_j. \end{aligned}$$

For a fixed initial choice of $O_{j,j'}$ (for example, j' being the right neighbor, $j+1$), this sequence of operations contributes $-4\beta\gamma\sigma^x_j$ to the expansion. Qubit j is connected to two neighbors on the ring structure, $j-1$ and $j+1$. Therefore, the initial term $O_{j,j'}$ can be either $\sigma^z_j\sigma^z_{j+1}$ or $\sigma^z_j\sigma^z_{j-1}$. Each of these two distinct initial choices yields a term with magnitude $4\beta\gamma$. Summing the contributions from these two possibilities gives the total magnitude for $c_2(X_j)$:

$$|c_2(X_j)| = 2 \cdot (4\beta\gamma) = 8\beta\gamma. \tag{5.28}$$

Where $\beta = 1$ and $\gamma = 1$, this yields $|c_2(X_j)| = 8$. This value matches the entry in Table 5.1. This result is also consistent with equation (5.30) for $q = 1$, which gives $(8\beta\gamma)^1 \cdot 2^{1-1} \cdot \mathrm{Cat}_1 = 8\beta\gamma$.

Table 5.2 presents specific operator sequences, referred to as paths, and their corresponding intermediate Pauli string states. These data are obtained from numerical simulations for $k = 2$ and $k = 4$, with parameters $\beta = 1, \gamma = 1, t = 1$. Each path represents a sequence of terms from H_X and H_{ZZ} applied in the nested commutators, $(\mathrm{ad}_H)^k(O)$. The Catalan-based formula is intended to provide the total coefficient of σ^x_j that results from summing all such contributing paths.

5.6.0.2 Combinatorial Structure of Operator Sequences

The formula for the coefficient $|c_{2q}(X_j)|$, equation (5.30), involves Catalan numbers. The presence of Catalan numbers often indicates an underlying combinatorial structure related

Table 5.2: Operator sequences and intermediate Pauli states from simulation for $\beta = \gamma = t = 1$, yielding σ_0^x (denoted X I...). The initial state (State 0) for each path is a $Z_u Z_v$ term from H_{ZZ}. S_i denotes State i. The "Operators" column lists the sequence of terms from $H = H_X + H_{ZZ}$ applied in the nested commutators. The "Orient." column is a symbolic representation of this sequence where X_i is denoted by $\times$, and Z_{uv} is denoted by L (left) or R (right) depending on the edge relative to the target qubit 0.

Operators	Orient.	Intermediate States ($S_0 \to S_1 \to \cdots \to S_k \sim \sigma_0^x$)
$k = 2$		
$X_0 Z_{0,1}$	$\times R$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,11}$	$\times L$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$k = 4$		
$X_0 X_0 X_0 Z_{0,1}$	$\times \times \times R$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_1 X_1 X_0 Z_{0,1}$	$\times \times \times R$	$Z_0 Z_1 \mathrm{I} \cdots \to Z_0 Y_1 \mathrm{I} \cdots \to Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_1 X_0 X_1 Z_{0,1}$	$\times \times \times R$	$Z_0 Z_1 \mathrm{I} \cdots \to Z_0 Y_1 \mathrm{I} \cdots \to Y_0 Y_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_0 X_1 X_1 Z_{0,1}$	$\times \times \times R$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to Y_0 Y_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,1} Z_{0,1} Z_{0,1}$	$\times RRR$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,11} Z_{0,1} Z_{0,1}$	$\times LRR$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,1} Z_{0,11} Z_{0,1}$	$\times RLR$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots Z_{11} \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,11} Z_{0,11} Z_{0,1}$	$\times LLR$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \cdots \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_0 X_0 X_0 Z_{0,11}$	$\times \times \times L$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_{11} X_{11} X_0 Z_{0,11}$	$\times \times \times L$	$Z_0 \mathrm{I} \ldots Z_{11} \to \mathrm{I} \ldots Z_0 Y_{11} \to \mathrm{I} \ldots Z_0 Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_{11} X_0 X_{11} Z_{0,11}$	$\times \times \times L$	$Z_0 \mathrm{I} \ldots Z_{11} \to \mathrm{I} \ldots Z_0 Y_{11} \to Y_0 \mathrm{I} \ldots Y_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_0 X_{11} X_{11} Z_{0,11}$	$\times \times \times L$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Y_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,1} Z_{0,1} Z_{0,11}$	$\times RRL$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \cdots \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,11} Z_{0,1} Z_{0,11}$	$\times LRL$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \cdots \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,1} Z_{0,11} Z_{0,11}$	$\times RLL$	$Z_0 Z_1 \mathrm{I} \cdots \to Y_0 Z_1 \mathrm{I} \cdots \to X_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$
$X_0 Z_{0,11} Z_{0,11} Z_{0,11}$	$\times LLL$	$Z_0 \mathrm{I} \ldots Z_{11} \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \cdots \to Y_0 \mathrm{I} \ldots Z_{11} \to X_0 \mathrm{I} \ldots$

to non-crossing partitions or balanced sequences. In this context, we can explore such a structure by considering sequences of effective operations derived from the Hamiltonian terms βH_X and γH_{ZZ}. These effective operations contribute to the overall transformation from an initial σ_j^z-based operator (within $O = H_{ZZ}$) to the final σ_j^x operator. The $(\beta\gamma)^q$ scaling of the formula suggests that sequences involving a net count of q contributions from βH_X and q contributions from γH_{ZZ}.

An effective H_X operation transitions the $\sigma_j^z \to \sigma_j^y$ on the target qubit j. An effective H_{ZZ} operation facilitates the subsequent $\sigma_j^y \to \sigma_j^x$ step on qubit j. An operator word $w = w_1 w_2 \cdots w_{2q}$ is a sequence of effective H_X operations and effective H_{ZZ} operations. For this sequence to successfully result in σ_j^x via the intermediate σ_j^y state, ordering is

necessary.

The numerical simulations in Table 5.2, and appended path data, generate by considering all distinct sequences of $k = 2q$ Hamiltonian terms (where each term is from H_X or H_{ZZ}). These sequences are not pure dyck paths. For example, a sequence of three H_X applications, $(\mathrm{ad}_{H_X})^3$, effectively produce one net $\sigma^z \to \sigma^y$ transformation. The simulation data shows that for $k = 4$ ($q = 2$), raw operator sequences such as $(X_0, X_0, X_0, Z_{0,1})$ contribute to the total coefficient $c_4(X_0)$. The formula in equation (5.30) accurately reproduces the sum of all these varied raw path contributions when $\beta = \gamma = 1$.

Dyck Path Correspondence and Balance Condition. To provide a combinatorial interpretation for the Cat_q factor in equation (5.30), we consider idealized sequences composed of exactly q effective H_X operations and q effective H_{ZZ} operations, as defined above, that lead to the final σ_j^x state. For such a sequence to be "productive" in the $\sigma_j^z \to \sigma_j^y \to \sigma_j^x$ pathway, it must adhere to a balance condition: when the $2q$ operations are considered in sequence, the number of effective H_X operations performed must always be greater than or equal to the number of effective H_{ZZ} operations that have subsequently acted to convert the intermediate σ_j^y-like state. This condition ensures that an effective H_{ZZ} operation always has a σ_j^y-like state available to act upon.

If an effective H_X operation is mapped to an "up-step" and an effective H_{ZZ} operation to a "down-step", these balanced sequences correspond to Dyck paths. A Dyck path of $2q$ steps (semilength q) starts and ends at height zero and never goes below height zero, comprising q up-steps and q down-steps. The number of such distinct Dyck paths is the q-th Catalan number:

$$\mathrm{Cat}_q = \frac{1}{q+1}\binom{2q}{q}. \tag{5.29}$$

Where $\binom{2q}{q}$ is the binomial coefficient. The appearance of Cat_q in equation (5.30) suggests

that these balanced, non-crossing operational structures are characteristic of the way contributing paths combine or are weighted in the overall sum. The formula's success implies that the total sum over all microscopic operator sequences, which can be intricate and not strictly one-to-one with simple Dyck paths, is governed by a count related to these idealized balanced structures.

Interpretation of Coefficient Factors.

1. **Characteristic Amplitude and H_{ZZ} Edge Choices** $(8\beta\gamma)^q$**:** The derivation for $k = 2$ ($q = 1$) showed that $|c_2(X_j)| = 8\beta\gamma$. A transformation sequence involves one effective H_X operation (proportional to β, providing a factor of 2 from $[\sigma_j^x, \sigma_j^z]$) and one effective H_{ZZ} operation (proportional to γ, providing another factor 2 from $[\sigma_j^z, \sigma_j^y]$). Together, these give a $4\beta\gamma$ factor. Additionally, the H_{ZZ} term, $\sigma_j^z\sigma_{j'}^z$, requires j' to be a neighbor of j. On the ring, there are two choices for this neighbor (left or right). This introduces a factor of 2. For $q = 1$, the combined factor is $2 \cdot (4\beta\gamma) = 8\beta\gamma$. The term $(8\beta\gamma)^q$ in the general formula, equation (5.30), suggests that the coefficient is constructed from q such fundamental operational units. Each unit contributes this characteristic amplitude, which includes the q-fold product of $\beta\gamma$, the 4^q from q pairs of Pauli commutations, and a factor of 2^q accounting for the two choices of neighboring edges for each of the q effective H_{ZZ} operations acting on site j.

2. **Dyck Path Count** (Cat_q)**:** This factor, as previously discussed, represents the number of distinct structural arrangements of q effective H_X operations and q effective H_{ZZ} operations that satisfy the balance condition necessary for the $\sigma_j^z \to \sigma_j^y \to \sigma_j^x$ transformation pathway. It reflects the number of non-crossing ways these operations can be productively ordered.

3. **Additional Multiplicity** (2^{q-1})**:** Equation (5.30) also includes an additional factor of 2^{q-1}. This 2^{q-1} factor accounts for further orientation or selection possibilities that arise. For a word of length $k = 4$ we, we see that path that On qubit j a maximal alternating substring $\cdots X_jY_jX_j \cdots$ of *odd* length is called an odd mixer block. When bracketed inside the nested commutator it produces a single σ_j^y. A subsequent driver on either edge of j then converts that σ_j^y to σ_j^x. If another mixer arrives before the driver, the block length becomes even and the net action commutes back to σ_j^z, killing the contribution.

Exact Coefficient and Asymptotics. Combining these factors, the magnitude of the coefficient for σ_j^x is given by:

$$|c_{2q}^{(\mathrm{Cat})}(X_j)| = (8\beta\gamma)^q \, 2^{\,q-1} \, \mathrm{Cat}_q, \quad q \geq 0 \tag{5.30}$$

Coefficients for odd orders, $c_{2q+1}^{(\mathrm{Cat})}(X_j)$, are 0. The formula in equation (5.30) reproduces the values in Table 5.1.

For large q, the asymptotic behavior of Cat_q is $\mathrm{Cat}_q \sim 4^q/(\sqrt{\pi}\, q^{3/2})$ [FS09]. Substituting this into equation (5.30) for $q \geq 1$ yields:

$$\begin{aligned} |c_{2q}^{(\mathrm{Cat})}(X_j)| &\sim (8\beta\gamma)^q \cdot 2^{q-1} \cdot \frac{4^q}{\sqrt{\pi} q^{3/2}} \\ &= (2^3\beta\gamma)^q \cdot 2^{q-1} \cdot (2^2)^q \cdot \frac{1}{\sqrt{\pi} q^{3/2}} \\ &= (\beta\gamma)^q \cdot 2^{3q} \cdot 2^{q-1} \cdot 2^{2q} \cdot \frac{1}{\sqrt{\pi} q^{3/2}} \\ &= (\beta\gamma)^q \cdot 2^{6q-1} \cdot \frac{1}{\sqrt{\pi} q^{3/2}} = \frac{(64\beta\gamma)^q}{2\sqrt{\pi}\, q^{3/2}}. \end{aligned} \tag{5.31}$$

The terms in the Baker-Campbell-Hausdorff (BCH) expansion for a quantity like $\langle H_{ZZ} \rangle_+ (t)$

are of the form $a_{2q} = \frac{(-1)^q t^{2q}}{(2q)!} c_{2q}(X_j)$. The ratio of the magnitudes of successive non-zero terms in such an expansion, $|a_{2(q+1)}|/|a_{2q}|$, is therefore

$$r_q^{(\mathrm{Cat})} = \frac{t^2}{(2q+2)(2q+1)} \frac{|c_{2(q+1)}^{(\mathrm{Cat})}(X_j)|}{|c_{2q}^{(\mathrm{Cat})}(X_j)|}.$$

For large q, using the asymptotic form from equation (5.31), this ratio becomes

$$\begin{aligned} r_q^{(\mathrm{Cat})} &\sim \frac{t^2}{4q^2} \frac{(64\beta\gamma)^{q+1}/(2\sqrt{\pi}(q+1)^{3/2})}{(64\beta\gamma)^q/(2\sqrt{\pi}q^{3/2})} \\ &\sim \frac{t^2}{4q^2} \cdot 64\beta\gamma \cdot \left(\frac{q}{q+1}\right)^{3/2}. \end{aligned} \tag{5.32}$$

Using the approximation $(q/(q+1))^{3/2} = (1 - 1/(q+1))^{3/2} \approx 1 - 3/(2(q+1)) \approx 1 - 3/(2q)$ for large q,

$$\begin{aligned} r_q^{(\mathrm{Cat})} &\approx \frac{16\beta\gamma t^2}{q^2} \left(1 - \frac{3}{2q} + O(q^{-2})\right) \\ &\approx \frac{16\beta\gamma t^2}{q^2} \left(1 + O(q^{-1})\right). \end{aligned} \tag{5.33}$$

The optimal truncation index $q_\star$ is estimated by setting $r_q^{(\mathrm{Cat})} = 1$. This yields $q_\star^2 \approx 16\beta\gamma t^2$, so $q_\star \approx 4\sqrt{\beta\gamma}|t|$. The corresponding optimal commutator order $k_\star^{(\mathrm{Cat})} = 2q_\star$ is therefore

$$k_\star^{(\mathrm{Cat})} \approx 8\sqrt{\beta\gamma}|t|. \tag{5.34}$$

Comparison with the Two-Edge Estimate. The two-edge refinement, detailed in Section 5.5 and given by equation (5.5), is described as yielding coefficients with an asymptotic behavior $\left|c_{2q}^{(2)}(j)\right| \sim \frac{(32\beta\gamma)^q}{\sqrt{\pi q}}$. The asymptotic form derived for the Catalan refinement in equation (5.31) is $\frac{(64\beta\gamma)^q}{2\sqrt{\pi}\, q^{3/2}}$. The Catalan refinement predicts a significantly

faster growth of coefficients at large q.

5.7 Commutator Sum Growth

For any operator O_p we define

$$N_k(O_p) \;=\; \langle +|^{\otimes n}\, \mathcal{S}_k(O_p)\, |+\rangle^{\otimes n}\,, \qquad \mathcal{S}_k(O_p) = \sum_{l=0}^{k} [\![H, O_p]\!]_l, \tag{5.35}$$

where $[\![H, O_p]\!]_l$ denotes the l-fold nested commutator of H with O_p .

As every edge of the ring is equivalent under translation, it is sufficient to analyse one representative edge. Fix

$$O_{01} \;=\; -\sigma_0^z\, \sigma_1^z, \tag{5.36}$$

the ZZ term acting on qubits 0 and 1. Substituting (5.36) into the definition of N_k in (5.35) gives the truncated expectation

$$N_k(O_{01}) \;=\; \langle +|^{\otimes n} \Big(\sum_{l=0}^{k} [\![H, O_{01}]\!]_l \Big)\, |+\rangle^{\otimes n}\,. \tag{5.37}$$

All subsequent formulas in this subsection will use $N_k(O_{01})$; the results for any other edge follow by symmetry.

The nested commutator sum $\mathcal{S}_k(O)$, projection onto $|+\rangle^{\otimes n}$ eliminates every Pauli word that contains a σ^y or a σ^z. The surviving terms are the single–qubit operators X_j (with the identity on the other qubits). As such, quantity $N_k(O_{01})$ measures the sum of coefficients of single-qubit σ_j^x operators up to commutator order k. The coefficient $|c_k^{(\mathrm{Cat})}|$ is the absolute magnitude of the largest σ_j^x coefficient generated by commutations. The growth of $N_k(O_{01})$ normalizing by $|c_k^{(\mathrm{Cat})}|$ at each intermediate k $N_k(O_{01})$ is plotted in figure 5.2.

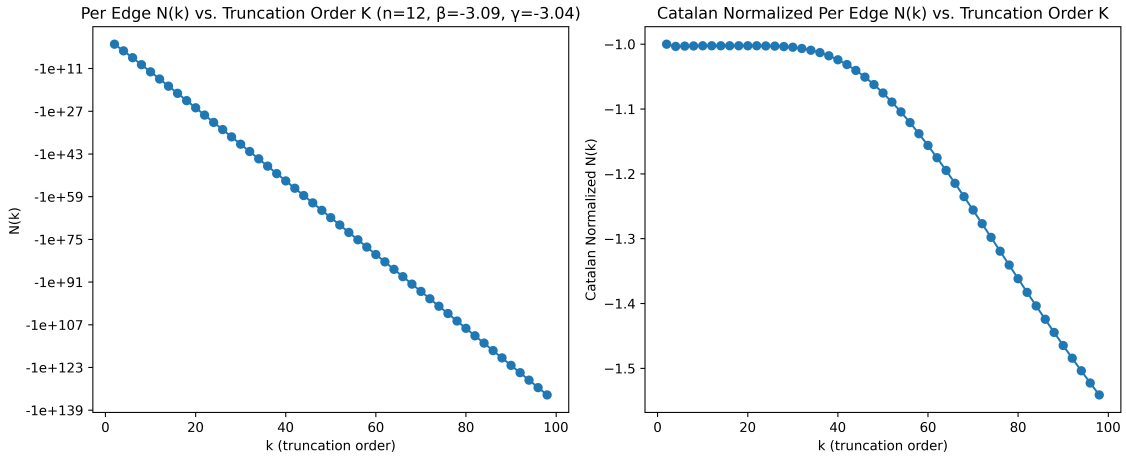

Figure 5.2: Growth of the truncated expectation $N_k(O_{01})$ for a ring of $n = 12$ qubits at the variational parameters $\beta = -3.09$ and $\gamma = -3.04$. **Left panel:** raw per-edge values on a logarithmic scale. **Right panel:** the same data divided by the Catalan bound C_k^{Cat} [equation (5.30)], which is the magnitude of the largest single-qubit coefficient appearing at order k.

Sensitivity to the relative magnitudes of β and γ. Equation (5.30) predicts that the largest single–qubit coefficient in $\mathrm{ad}_H^k(O_{01})$ scales as $(8\beta\gamma)^q$. This dependence assumes that every effective pair of operations contributes the product $\beta\gamma$. In the left panel the fitted values satisfy $|\beta| \approx |\gamma|$ ($\beta = -3.09$, $\gamma = -3.04$), hence successive factors of β and γ are almost equal in magnitude and the simple product captures the growth of $N_k(O_{01})$ accurately. When $|\beta|$ and $|\gamma|$ differ substantially, one factor dominates the product while the other remains sub-leading; the $(8\beta\gamma)^q$ then overestimates the contribution of the smaller coefficient and underestimates that of the larger one.

Accumulation of rounding error at large commutator depth. In exact arithmetic the ratio $N_k(O_{01})/C_k^{\mathrm{Cat}}$ would likely remain close to a constant at all commutator depths. The downward drift visible in the right panel for $k \gtrsim 80$ is most plausibly attributed to rounding error. Evaluating $\mathcal{S}_k(O_{01})$ involves thousands of nested additions and multiplica-

tions whose individual rounding errors compound. Even with 128 float point precision the accumulated error eventually becomes comparable to the true value of the ratio and the numerical curve departs from the ideal envelope. Using higher precision would delay, but not remove, this effect.

5.8 Intermediate–locality profile for even orders

Nested commutators of even order $k \leq k_{\text{small}}$ (A quantitative value of k_{small} is derived in the next section; only the qualitative pattern is required here) yield exactly n Pauli word that survives the projection onto $|+\rangle^{\otimes n}$, namely the one–local operator $X_j \otimes I^{\otimes(n-1)}$ on each site j. Odd orders produce no IX–only words and therefore make no contribution to the expectation value.

Although the surviving operator is one–local, the expansion explores higher locality internally. For a Pauli term P define

$$\lambda(P) = \#\{\text{non-identity factors in } P\}\,, \tag{5.38}$$

called the locality of P. During the symbolic evaluation we store, for every word P appearing in the k^{th} commutator, the quantity

$$\Lambda_k(P) = \max_{\text{generation path } \gamma \to P} \lambda(\gamma), \tag{5.39}$$

that is, the largest locality encountered on any path that results from the directed acyclic graph that generates P. The procedure Bubble-Commutator shown below computes this value and is called at each increase in k.

Algorithm 4 Bubble-Commutator

Require: Maps H and A; each key is a Pauli string and each value a locality λ

$\widetilde{A} \leftarrow$ EmptyMap()
$\mathcal{L} \leftarrow$ EmptyMap()
for all $(P_1, \lambda_1) \in H$ **do**
  for all $(P_2, \lambda_2) \in A$ **do**
    for all $P \in \text{Commutator}(P_1, P_2)$ **do**
      $\lambda \leftarrow \max\big(\text{Locality}(P), \lambda_1, \lambda_2\big)$
      $\widetilde{A}[P] \leftarrow \max\big(\text{GetDefault}(\widetilde{A}, P, 0), \lambda\big)$
      $\mathcal{L}[P]$ .append $\big((P_1, \lambda_1), (P_2, \lambda_2)\big)$
return $(\widetilde{A}, \mathcal{L})$

The maximal value of $\Lambda_k(P)$ over all words in the commutator satisfies

$$\max_P \Lambda_k(P) = \left\lfloor \frac{k}{4} \right\rfloor + 2 \tag{5.40}$$

The bound follows from the pattern established in Subsection 5.6: While raising locality by one unit requires only two layers of commutators, four consecutive commutators are necessary for any $k \leq k_{\text{small}}$. The factor of 2 increase in equation (5.40) occurs because the Dyck paths that yield single-X words must return to locality one after every excursion to higher locality. Paths that result in only single X words are also start as single X words. For every increase in locality, there must be a corresponding decrease in locality for the final term to be of the correct form.

5.9 General locality growth

Subsection 5.8 analyzed only even orders because they alone can return Pauli words built from σ^x and the identity. Removing that restriction reveals a steeper internal-locality envelope.

Definition. For every integer $k \geq 0$ set

$$\Lambda_k^\star \;=\; \max_{P \in [\![H,C]\!]_k} \Lambda_k(P), \tag{5.41}$$

where $\Lambda_k(P)$ is the path–wise locality defined earlier. Direct symbolic evaluation shows

$$\Lambda_k^\star = \left\lfloor \frac{k}{2} \right\rfloor + 2, \qquad 0 \leq k < 2n-2. \tag{5.42}$$

Two–step locality increment. Assume the current word already acts non–trivially on site j. Two successive commutators on the edge $(j, j+1)$ operate as follows:

(i) $[X_j, \cdot]$ converts a local σ_j^z into σ_j^y without altering locality,

(ii) $[Z_{j,j+1}, \cdot]$ rotates σ_j^y into σ_j^x and appends a fresh σ_{j+1}^z, raising locality by one.

Because the newly created σ_{j+1}^z remains available, the same two–step pattern can repeat on edge $(j+1, j+2)$. Consequently locality increases by one after every two commutators, which yields the slope $\frac{1}{2}$ in (5.42).

Locality scales. For even orders below k_{small} the locality of an operator on any path that *returns* to an IX word must come back down by the time the commutator chain closes, doubling the commutator budget and giving the slower growth $\lfloor k/4 \rfloor + 2$ derived in Subsection 5.8. When odd orders are allowed the envelope follows $\lfloor k/2 \rfloor + 2$ because no IX words are possible. At k_{small} both locality tracks reach the graph diameter$n$, and the first Pauli words with a single identity enter the IX sector.

Catalog at order $2n-3$. Apply the two–step rule successively on edges

$$(0,1),\ (1,2),\ \ldots,\ (n-3, n-2). \tag{5.43}$$

After $2(n-2)-1 = 2n-3$ commutators the ring hosts exactly $2n$ words,

$$\begin{aligned} &X_0 \dots X_{j-1} Y_j Z_{j+1} X_{j+2} \dots X_{n-1}, \\ &X_0 \dots X_{j-1} Z_j Y_{j+1} X_{j+2} \dots X_{n-1}, \end{aligned} \qquad j = 0, \dots, n-1. \tag{5.44}$$

Each word contains one σ^y, one σ^z and $n-2$ σ^x's; its locality is n.

5.10 Depth k_{small}

Definition. Let k_{small} be the smallest even order for which a Pauli word $I_j X^{\otimes(n-1)}$ appears in the expansion. Symbolic evaluation gives

$$k_{\text{small}} = 2n-2. \tag{5.45}$$

Two–step completion to a single–identity word. Act once more with the commutator on edge $(j, j+1)$. If the current word is $Y_j Z_{j+1}$, the extra commutator converts $Y_j Z_{j+1}$ into $X_j I_{j+1}$; if the word is $Z_j Y_{j+1}$ it converts into $I_j X_{j+1}$. Thus the $\sigma^y \sigma^z$ pair becomes a single σ^x and an identity. Exactly one identity appears, and exactly one such word arises from each edge, producing n single–identity words at the depth in (5.45).

5.11 Character of Ballistic Evolution

Coefficient–factorial competition. Equation (5.30) gives $|c_{2q}| \propto (8\beta\gamma)^q \, 2^{\,q-1} \mathrm{Cat}_q$ an exponential growth that competes with the factorial weight $1/(2q)!$ in the BCH series. The magnitude of the $2q$-th term in the expansion is given by $a_{2q}(t) = (it)^{2q} c_{2q}/(2q)!$ as in Eq. (5.17). Inserting the Catalan coefficient of Eq. (5.29) and applying Stirling's formula to

$(2q)!$ gives the successive–term ratio

$$r_q(t) \;=\; \frac{|a_{2(q+1)}|}{|a_{2q}|} \;\approx\; \frac{16\,\beta\gamma\,t^2}{(q+1)^2}. \tag{5.46}$$

This ratio governs the shape of the distribution of term magnitudes as a function of the commutator order q. The envelope $|a_{2q}|$ therefore *grows* while $r_q > 1$, reaches its apex when $r_q \simeq 1$, and decays once $r_q < 1$. Solving $r_q = 1$ gives the characteristic peak, the commutator order that carries the largest weight in the BCH series

$$q_{\text{turn}} \;\approx\; 4\sqrt{\beta\gamma}\,|t|. \tag{5.47}$$

Converting to the index $k = 2q$ corresponds to the truncation order defined in Section 5.4 is $k_\star = 2q_{\text{turn}} \approx 8\sqrt{\beta\gamma}\,|t|$

To capture how sharply the envelope decays, we set $q = q_{\text{turn}}+\delta$ under the assumption that $|\delta| \ll q_{\text{turn}}$. δ is the distance, in commutator index, from the peak order. Insert this shift into the ratio $r_q(t)$ of Eq. (5.46):

$$r_q(t) \;=\; \frac{16\,\beta\gamma\,t^2}{\big(q_{\text{turn}}+\delta+1\big)^2} \;=\; \frac{16\,\beta\gamma\,t^2}{\big(q_{\text{turn}}+1\big)^2}\big[1+\delta/(q_{\text{turn}}+1)\big]^{-2}. \tag{5.48}$$

Because the prefactor equals 1 by definition of q_{turn}

$$r_q(t) = (1+x)^{-2}, \qquad x = \frac{\delta}{q_{\text{turn}}+1}. \tag{5.49}$$

For $|x| \ll 1$ expand with the binomial series $(1+x)^{-2} = 1 - 2x + 3x^2 - \ldots$:

$$r_q(t) \simeq 1 - \frac{2\,\delta}{q_{\text{turn}}+1} + O\big(\delta^2/q_{\text{turn}}^2\big) \approx 1 - \frac{2\,\delta}{q_{\text{turn}}}. \tag{5.50}$$

Thus moving the index by a single step, $\delta = \pm 1$, changes the ratio by approximately $2/q_{\text{turn}}$.

For there, we fix a constant $0 < \eta < 1$ and restrict the shift to $|\delta| \le \eta q_{\text{turn}}^{1/2}$. Inside this window

$$\left|r_q(t) - 1\right| \le \frac{2\eta}{q_{\text{turn}}^{1/2}} + O\big(q_{\text{turn}}^{-1}\big) = O\big(q_{\text{turn}}^{-1/2}\big). \tag{5.51}$$

Hence every order in the band

$$q_{\text{turn}} - \eta\, q_{\text{turn}}^{1/2} \;\le\; q \;\le\; q_{\text{turn}} + \eta\, q_{\text{turn}}^{1/2} \tag{5.52}$$

a region of width $2\eta\, q_{\text{turn}}^{1/2} = O(q_{\text{turn}}^{1/2})$ has a term magnitude that differs from its neighbor by only a polynomial factor $O(q_{\text{turn}}^{-1/2})$. Consequently no single term dominates inside this neighborhood; one must sum an extensive set of adjacent orders to approximate the series, a characteristic signature of non-perturbative behavior [ESS19].

System–size threshold for global support. The closed–form coefficient

$$c_{2q} = (8\beta\gamma)^q\, 2^{\,q-1} \mathrm{Cat}_q \tag{5.53}$$

(Eq. (5.29)) contains no explicit dependence on the number of qubits N; the derivation assumed that every commutator sequence can attach a driver term to either edge of the marked qubit without revisiting an already-acted edge. That assumption remains valid while the internal locality $\Lambda_k^\star$ is below the ring diameter $N/2$. Once the commutator depth exceeds $k_{\text{wrap}} = 2N - 2$ (Eq. (5.45)) wrapping occurs and additional cancellations introduce N-dependent combinatorial factors.

For the IX–sector the locality bound $\Lambda_k = \lfloor k/4 \rfloor + 2$ (Eq. (5.42)) reaches one half of the lattice when $k \approx 2N$, and the first Pauli word with a single identity and $N - 1$ σ^x's appears exactly at $k = 2N - 2$; both statements are independent of the microscopic

parameters β, γ, t.

The commutator order at which BCH terms turn from growth to decay is

$$q_{\text{turn}} \;\approx\; 4\sqrt{\beta\gamma}\,|t| \qquad \text{(Section 5.4)}. \tag{5.54}$$

Because full support requires $q \gtrsim N$, a run time that satisfies

$$4\sqrt{\beta\gamma}\,|t| \;\gtrsim\; N \tag{5.55}$$

guarantees that the dominant band of terms $\left|q - q_{\text{turn}}\right| \le \eta q_{\text{turn}}^{1/2}$ samples operators whose internal histories already cover the entire ring. Only when $\sqrt{\beta\gamma}\,|t| \propto N$ does ballistic optimization employ coefficients whose commutator paths are influenced by the full geometry of the graph.

Chapter 6

Conclusion

The preceding three chapters interrogate the twin questions raised at the close of the introduction: How far can shallow, locality-bounded quantum circuits be pushed before the Overlap Gap Property (OGP) arrests progress, and what is gained by increasing locality of the optimization?

Chapter 3 establishes the empirical baseline. Systematic simulations of QAOA on random Max-k-SAT formulae show that the QAOA clears the OGP obstruction whenever p matches the hypergraph diameter of the instance.

Beginning with the properties of the OGP in random Max-k-SAT, our observed OGP onset threshold (Figure 3.4) η_{OGP} is significantly higher than the inapproximability bound $\eta_{\text{apx}} \leq 1 - 2^{-k}$, which for $k = 3$ is approximately 0.875 [Has01]. The large value of $\eta_{OGP} = 0.998$ can likely be attributed to the small problem sizes considered (specifically, $n = 13$ to $n = 26$). Although a decrease in this threshold is expected as n increases, doubling the problem size from $n = 13$ to $n = 26$ only led to a slight decline from an η_{OGP} of 0.9984 to 0.9981. Determining whether η_{OGP} will converge to the inapproximability bound or to another value, and the rate at which it approaches its asymptotic value, requires analysis of

larger problem sizes.

Turning to the OGP occurrence rate (Figure 3.3), an increase in clause density α correlates with a higher occurrence rate of OGPs in random Max-k-SAT instances. Although there appears to be no minimal OGP occurrence rate regardless of the number of variables n, for a fixed n, the minimal clause density required to observe OGPs increases with the clause arity k. While the dependency of OGP occurrence on k is known—the obstruction bounds for k-SAT are relevant only for $k \geq 4$—our results indicate a more nuanced relationship. Specifically, while fixing n and increasing k requires a larger α_k to observe similar OGP occurrence rates, when we normalize α_k by the satisfiability threshold $\alpha^*_{k,\text{crit}}$, we find higher OGP occurrence rates closer to the corresponding satisfiability threshold.

When the QAOA is theoretically unobstructed and all samples have a hypergraph diameter of one ($\text{diam}(H_{\phi_k(n,m)}) = 1$), we observe no difference in approximation performance between formulas that have an OGP and those that do not (Figure 3.8). Therefore, we find no indication of further obstruction on local algorithms, such as the QAOA, that do not arise from the OGP. Any such additional obstructions would have manifested as a further disparity in approximation quality. Nevertheless, even when the hypergraph diameter is held constant, increases in both k and n necessitate higher ansatz depths p to maintain the same probability of observing near-optimal configurations (Figure 3.6, Panels 1 and 2). This increase suggests, at least for the QAOA, that larger problem size contributes to hardness for random Max-k-SAT in ways not captured by our operator support model.

A further complication arises because the required ansatz depth does not scale with increases in clause density, as shown in Figure 3.6, Panel 3. As discussed (Figure 3.3), for fixed n and k higher clause density results in increased OGP occurrence rates. However, once there is sufficient clause density for an OGP to be present, further increases in clause density to do not necessitate higher ansatz depth to maintain approximation quality. This

apparent absence of scaling defies expectations and warrants further investigation.

Examining the QAOA's approximation performance near the satisfiability threshold (Figure 3.9), we find that the performance is consistent with our results for hypergraphs of diameter one ($\mathrm{diam}(H_{\phi_k(n,m)}) = 1$). Even at the largest diameter considered, $\mathrm{diam}(H_{\phi_k(n,m)}) = 3$, which is the maximum still within ansatz depth $p = 1$, we observe no performance differences between formulas that have an OGP and those that do not. These results indicate that once the QAOA has full support, no additional depth scaling is necessary to achieve approximation performance parity between formulas that exhibit an OGP and those that do not, even though formulas without an OGP are expected to be easier to optimize.

In terms of overall approximation accuracy, our results agree with the k-SAT decision problem results of Boulebnane and Montanaro [BM22b], which reported comparable performance between the QAOA and classical SAT solvers on small problem sizes. We find that the QAOA achieves similar performance to classical MaxSAT solvers, which correctly enumerated all near-optimal solutions for the small problem sizes considered. Moreover, given the modest required ansatz depth for full support of large random Max-k-SAT instances and the performance parity observed for instances with and without an OGP beyond that depth, our findings do not preclude the QAOA from some form of quantum utility on Max-k-SAT.

The results in Chapter 3 relate the expected approximation ratio of a depth p QAOA ansatz to the diameter $\mathrm{diam}(G)$ of the interaction hypergraph G that encodes a random Max-k-SAT instance.

Conjecture 6.0.1 (OGP–parity) *For any random max-k-SAT instance ensemble with fixed (n, k, m), let G be the associated interaction hypergraph and let p satisfy $2p + 1 \geq \mathrm{diam}(G)$. Then the expected approximation ratio of depth-p QAOA is identical when the*

ensemble is conditioned on exhibiting the Overlap Gap Property and when it is conditioned on not exhibiting the Overlap Gap Property.

The conjecture claims approximation parity when $2p + 1$ surpasses $\mathrm{diam}(G)$ every diameter-length path is encoded in the control unitaries, which yields global sensitivity to all clauses.

In chapter 4 we shifted away from considering the limitations imposed by the OGP to considering whether locality can be sidestepped rather than overcome. Three families of non-local mixers are introduced: global $X \otimes X$ couplings, distance-weighted graph mixers, and frustration-guided traversal mixers. None lift the mean cut value on 3-regular graphs at depth 1, though traversal mixers narrow variance by quenching extreme mis-cuts. A more radical step replaces the Trotterized alternation entirely. *Ballistic Optimization* evolves under the untrotterized sum $e^{-i(\beta H_X + \gamma H_{ZZ})t}$. On the same graph ensemble ballistic circuits exceed depth-1 QAOA and match or beat the worst Goemans–Williamson cuts while using gate counts polynomial in sparsity. The mixed numerical picture invites an analytic account of when ballistic evolution becomes truly global.

A QAOA circuit alternates a cost Hamiltonian H_C with a mixer Hamiltonian H_B. The transverse-field mixer has locality radius 1. Define a locality-r mixer family $\{M_r\}_{r>1}$ where each M_r is a sum of Pauli strings whose support lies inside radius r.

Conjecture 6.0.2 (locality-extended mixer parity) *There exists a p_0 and a $\{M_r\}_{r>1}$ such that, for every fixed depth $p < p_0$, the ensemble-average approximation ratio produced by replacing H_B with M_r equals the ratio obtained with H_B, while the locality per layer increases from radius 1 to radius r.*

The conjecture generalizes mixer design criteria that preserve the cost-subspace while enlarging support.

Finally chapter 5 supplies an account and explains the performance gap between ballistic optimization and QAOA at $p = 1$. Using the symmetric "ring of disagrees" Max-Cut instance (an N-vertex cycle) as a testbed, we derived the Heisenberg-picture evolution of ballistic optimization through the Baker–Campbell–Hausdorff (BCH) expansion.

A combination of numerical simulation and the BCH expansion shows that the amplitude of any single-qubit σ^x term after $k=2q$ commutators steps grows for even orders $k = 2q$ as:

$$|c_{2q}| \sim (8\beta\gamma)^q \, 2^{\,q-1} \mathrm{Cat}_q \tag{6.1}$$

where Cat_q is the q-th Catalan number and all odd orders vanish. The Catalan factor Cat_q appears because every occurrence of H_X in a nested-commutator word must be balanced by a later H_{ZZ} acting on the same qubit. Exact numerical simulations on rings up to $N = 12$ reproduce these coefficients, validating the analytic formula.

Beyond the coefficient magnitudes, we characterized how operator locality expands under ballistic dynamics. For the single-X sector, the internal support grows as

$$\Lambda_k = \lfloor k/4 \rfloor + 2 \tag{6.2}$$

so an operator with full N-qubit support first emerges at $k \approx 2N$ (precisely $k = 2N - 2$). Because the turning index of the BCH envelope is $q_{\mathrm{turn}} \approx 4\sqrt{\beta\gamma}\,|t|$, full support requires

$$4\sqrt{\beta\gamma}\,|t| \;\gtrsim\; N \qquad \text{or} \qquad t = \Theta\big(N/\sqrt{\beta\gamma}\big) \tag{6.3}$$

when β, γ are fixed. At shorter run times the series is dominated by lower-order terms that, although exponentially more numerous than the single layer of QAOA because of the Catalan combinatorics, still act on only $O(k)$ adjacent sites. Ballistic optimization therefore explores a much richer space of local configurations than depth-1 QAOA, yet it

still needs evolution times that scale linearly with problem size before truly system-spanning operators appear. The is captured by competition between the Catalan growth and the $(2q)!$ denominator yields a turnover at $q_* \approx 4\sqrt{\beta\gamma}\, t$. Global support is therefore guaranteed only when $t \gtrsim N/\sqrt{\beta\gamma}$, implying a linear-time threshold for ballistic quantum advantage on rings.

$$U_{\mathrm{BO}}(t) = \exp[-\mathrm{i}t(\beta H_B + \gamma H_C)] \tag{6.4}$$

be a trotterized ballistic unitary

Conjecture 6.0.3 (ballistic advantage) . *Their exists a fixed QAOA depth p_0, that for all $p < p_0$ there exists an evolution time t^* such that a realizable ballistic unitary $U_{\mathrm{BO}}(t^*) = \exp[-\mathrm{i}t^*(\beta H_B + \gamma H_C)]$ attains an expected approximation ratio on average greater than the average ratio achieved by any depth-p QAOA circuit on the same ensemble of instances*

6.1 Future Work

The investigation in this dissertation has sought to clarify the practical implications of locality constraints on near-term quantum optimization. While the findings herein resolve certain questions about the capabilities of shallow quantum algorithms, they also open several new avenues for future inquiry. The following paragraphs outline a selection of these research directions, probing quantum hardness, performance scaling, and design principles.

Clause Density Independence Is the observed insensitivity of QAOA performance to clause density a small-p artifact, and does a new hardness regime dominated by combinatorial complexity emerge at high depth?

In Chapter 3, we saw that for $n \leq 26$ and small p, the performance of QAOA on Max-k-SAT instances exhibiting an Overlap Gap Property (OGP) was insensitive to increases in clause density m. This result contradicts the classical intuition that higher constraint density should lead to computationally harder problem instances. While this phenomenon may simply be an artifact of the small problem sizes studied, the fact that it appears robustly across varying clause arities merits deeper theoretical investigation. Clarifying this behavior would significantly refine our understanding of quantum optimization algorithms' limitations relative to their classical counterparts.

Ballistic Optimization Beyond the Ring of Disagrees How does the operator growth in Ballistic Optimization change when moving from the "ring of disagrees" to harder to solve problem graphs, and can this be characterized?

Chapter 5's primary analytical result is the derivation of Catalan-number-driven coefficient growth for BO on the "ring of disagrees," a model whose underlying symmetries lead to highly structured, predictable dynamics. As discussed, the ring of disagrees is bipartite, and hence its Max-Cut can be solved efficiently in polynomial time. However, the NP-hard combinatorial problems where a quantum advantage is truly sought, such as Max-Cut on random 3-regular graphs, exhibit distinctly different dynamics. Investigating the differences in operator growth patterns for such cases could clarify the performance advantages observed and help guide the design of improved quantum optimization strategies.

Depth Scaling of BO vs. QAOA Performance How does the depth-to-approximation ratio scaling of the QAOA and the time-to-approximation ratio scaling of ballistic optimization compare as we increase p and t?

On 3-regular graphs with $n \approx 12$, our experiments showed that a single-layer BO circuit outperformed depth-1 QAOA by an average approximation ratio gap of ~ 0.13.

It remains open whether there exists a threshold depth or regime where the QAOA's structured alternating layers surpass BO's continuous approach. Moreover, understanding their relative performance in regimes where both algorithms have full support could inform whether continuously parameterized ansatz inherently provide expressive power advantages over discretely layered approaches.

Problem Size Scaling of BO vs. QAOA Does the performance gap between ballistic optimization and QAOA persist as we increase problem size?

Our experiments indicated that BO outperforms depth-1 QAOA by an average approximation ratio gap of ~ 0.13. However, this advantage diminished as n grew larger in our test set, raising the question of scalability. Is this reduction in relative performance advantage caused by holding BO's evolution time t constant while increasing n, or does it signal an inherent limitation in the continuous-time optimization approach itself? Clarifying this scaling relationship may be essential for determining the practical applicability of BO to large-scale quantum optimization problems.

Bibliography

[ABO96] Dorit Aharonov and Michael Ben-Or. Polynomial simulations of decohered quantum computers. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–55, 1996.

[Ach09] Dimitris Achlioptas. Random satisfiability. In *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 245–270. IOS Press, 2009.

[ACO08] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2008.

[ALS13] Gilles Audemard, Jean-Marie Lagniez, and Laurent Simon. Improving glucose for incremental SAT solving with assumptions: Application to MUS extraction. In *Theory and Applications of Satisfiability Testing – SAT 2013*, pages 309–317, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[AMM14] Matthew Amy, Dmitri Maslov, and Michele Mosca. Polynomial-time t-depth optimization of clifford+t circuits via matroid partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(10):1476–1489, 2014.

[AP03] Dimitris Achlioptas and Yuval Peres. The threshold for random k-SAT is $2^k ln2 - o(k)$, 2003.

[ARCB21] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte. Parameter concentrations in quantum approximate optimization. *Physical Review A*, 104(1), 2021.

[ARNSS16] Houssam Abdul-Rahman, Bruno Nachtergaele, Robert Sims, and Günter Stolz. Entanglement dynamics of disordered quantum XY chains. *Letters in Mathematical Physics*, 106(5):649–674, 2016.

[ATS03] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge, 2003.

[AvDK+04] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Journal on Computing*, 37(1):166–194, 2004.

[BBC57] R. Bellman, R.E. Bellman, and Rand Corporation. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957.

[BBC+95] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995.

[BBD+22] Sebastian Brandhofer, Daniel Braun, Vanessa Dehn, Gerhard Hellstern, Matthias Hüls, Yanjun Ji, Ilia Polian, Amandeep Singh Bhatia, and Thomas Wellens. Benchmarking the performance of portfolio optimization with QAOA. *Quantum Information Processing*, 22(1), 2022.

[BBL+25] Naphan Benchasattabuse, Andreas Bärtschi, Pedro García-Pintos Luis John Golden, Nathan Lemons, and Stephan Eidenbenz. Lower bounds on the number of rounds of the quantum approximate optimization algorithm required for guaranteed approximation ratios. *Physical Review A*, 111, 2025.

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. ACM, 2014.

[BCLK+22] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1), 2022.

[BCS+20] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent hamiltonian simulation with math. *Quantum*, 4:254, 2020.

[BE20] Andreas Bärtschi and Stephan Eidenbenz. Grover mixers for QAOA: Shifting complexity from mixer design to state preparation. In *Proc. IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*, pages 72–82, 2020.

[BFM+22] 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). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[BGMZ22] Joao Basso, David Gamarnik, Song Mei, and Leo Zhou. Performance and limitations of the QAOA at constant levels on large sparse hypergraphs and spin glass models. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022.

[BH22] Guy Bresler and Brice Huang. The algorithmic phase transition of random k-SAT for low degree polynomials. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 298–309, 2022.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics*, pages 53–74. American Mathematical Society, 2002.

[BK21] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is NP-hard. *Physical Review Letters*, 127(12), 2021.

[BKKT20] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters*, 125(26), 2020.

[BL24] 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.

[BM22a] Boaz Barak and Kunal Marwaha. Classical algorithms and quantum limitations for maximum cut on high-girth graphs. In *13th Innovations in*

Theoretical Computer Science Conference (ITCS 2022). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[BM22b] Sami Boulebnane and Ashley Montanaro. Solving boolean satisfiability problems with the quantum approximate optimization algorithm, 2022.

[BMO+15] Boaz Barak, Ankur Moitra, Ryan O'Donnell, Prasad Raghavendra, Oded Regev, David Steurer, Luca Trevisan, Aravindan Vijayaraghavan, David Witmer, and John Wright. Beating the random assignment on constraint satisfaction problems of bounded degree, 2015.

[BMWV+23] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E. O'Brien. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A*, 107(3), 2023.

[Bri15] C.H̃all Brian. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222 of *Graduate Texts in Mathematics*. Springer, 2015.

[Cam97] John Campbell. On a law of combination of operators bearing on the theory of continuous transformation groups. *Proceedings of the London Mathematical Society*, s1-28(1):381–390, 1897.

[CBE20] Jeremy Cook, Andreas Bärtschi, and Stephan Eidenbenz. The quantum alternating operator ansatz on maximum k-vertex cover. In *Proc.IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*, pages 83–92, 2020.

[CCZZ24] Lixue Cheng, Yu-Qin Chen, Shi-Xin Zhang, and Shengyu Zhang. Quantum approximate optimization via learning-based adaptive optimization. *Communications Physics*, 7(1), 2024.

[CGPR19] Wei-Kuo Chen, David Gamarnik, Dmitry Panchenko, and Mustazee Rahman. Suboptimality of local algorithms for a class of max-cut problems. *The Annals of Probability*, 47(3), 2019.

[CLSS22a] Chi-Ning Chou, Peter J. Love, Juspreet Singh Sandhu, and Jonathan Shi. Limitations of local quantum algorithms on random Max-k-XOR and beyond, 2022.

[CLSS22b] Chi-Ning Chou, Peter J. Love, Juspreet Singh Sandhu, and Jonathan Shi. Limitations of local quantum algorithms on random Max-k-XOR and beyond, 2022.

[CO10] Amin Coja-Oghlan. A better algorithm for random k-SAT. *SIAM Journal on Computing*, 39(7):2823–2864, 2010.

[COP16] Amin Coja-Oghlan and Konstantinos Panagiotou. The asymptotic k-SAT threshold. *Advances in Mathematics*, 288:985–1068, 2016.

[Cro18] Gavin E Crooks. Performance of the quantum approximate optimization algorithm on the maximum cut problem, 2018.

[CRU24] Ernesto Campos, Daniil Rabinovich, and Alexey Uvarov. Depth scaling of unstructured search via quantum approximate optimization, 2024.

[DB97] O Dubois and Y Boufkhad. A general upper bound for the satisfiability threshold of randomr-SAT formulae. *Journal of Algorithms*, 24(2):395–420, 1997.

[Deu85] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A*, 400(1818):97–117, 1985.

[Dev16] Simon J. Devitt. Performing quantum computing experiments in the cloud. *Physical Review A*, 94(3), 2016.

[Die17] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2017.

[DN05] Christopher M. Dawson and Michael A. Nielsen. The solovay–kitaev algorithm. *Quantum Information and Computation*, 6(1), 2005.

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[DSS21] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k, 2021.

[EMW21] Daniel J. Egger, Jakub Mareček, and Stefan Woerner. Warm-starting quantum optimization. *Quantum*, 5:479, 2021.

[ESS19] Alexander Engel, Graeme Smith, and E.P̃arker Scott˙Quantum algorithm for the vlasov equation. *Physical Review A*, 100:062315, 2019.

[FB99] Ehud Friedgut and Jean Bourgain. Sharp thresholds of graph properties, and the k-sat problem. *Journal of the American Mathematical Society*, 12(4):1017–1054, 1999.

[FGG+01] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292(5516):472–475, 2001.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[FGG15] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem, 2015.

[FGG20] Edward Farhi, David Gamarnik, and Sam Gutmann. The quantum approximate optimization algorithm needs to see the whole graph: A typical case, 2020.

[FGGZ22] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The quantum approximate optimization algorithm and the sherrington–kirkpatrick model at infinite size. *Quantum*, 6:759, 2022.

[FGRV25] Edward Farhi, Sam Gutmann, Daniel Ranard, and Benjamin Villalonga. Lower bounding the MaxCut of high girth 3-regular graphs using the QAOA, 2025.

[FH19] Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm, 2019.

[FS96] Alan Frieze and Stephen Suen. Analysis of two simple heuristics on a random instance ofk-sat. *Journal of Algorithms*, 20(2):312–355, 1996.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, USA, 2009.

[Gam21] David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41), 2021.

[GGF+23] 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, 2023.

[GJW22] David Gamarnik, Aukosh Jagannath, and Alexander S. Wein. Hardness of random optimization problems for boolean circuits, low-degree polynomials, and langevin dynamics, 2022.

[GM19] G. G. Guerreschi and A. Y. Matsuura. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Scientific Reports*, 9(1), 2019.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 212–219. ACM, 1996.

[GS13] David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs, 2013.

[GS17] David Gamarnik and Madhu Sudan. Performance of sequential local algorithms for the random NAE-k-SAT problem. *SIAM Journal on Computing*, 46(2):590–619, 2017.

[GV20] Vijay Ganesh and Moshe Y. Vardi. On the unreasonable effectiveness of SAT solvers. In *Beyond the Worst-Case Analysis of Algorithms*, 2020.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[Has96] Johan Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC)*, pages 627–636, 1996.

[Has01] Johan Hastad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[Has19] M. B. Hastings. Classical and quantum bounded depth approximation algorithms, 2019.

[Hau06] Felix Hausdorff. Die symbolische exponentialformel in der gruppentheorie. *Berichte über die Verhandlungen der Königlich-Sächsischen Gesellschaft der Wissenschaften zu Leipzig, Mathematisch-Physikalische Klasse*, 58:19–48, 1906.

[HHS+24] Tianyi Hao, Zichang He, Ruslan Shaydulin, Jeffrey Larson, and Marco Pistoia. End-to-end protocol for high-quality QAOA parameters with few shots, 2024.

[HLZ04] Eran Halperin, Dror Livnat, and Uri Zwick. MAX CUT in cubic graphs. *Journal of Algorithms*, 53(2):169–185, 2004.

[HS22] Brice Huang and Mark Sellke. Tight lipschitz hardness for optimizing mean field spin glasses, 2022.

[HSN+21] Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Sean Demura, Andrew Dunsworth, Daniel Eppens, Austin Fowler, Brooks Foxen, Craig Gidney, Marissa Giustina, Rob Graff,

Steve Habegger, Alan Ho, Sabrina Hong, Trent Huang, L. B. Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Cody Jones, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Seon Kim, Paul V. Klimov, Alexander N. Korotkov, Fedor Kostritsa, David Landhuis, Pavel Laptev, Mike Lindmark, Martin Leib, Orion Martin, John M. Martinis, Jarrod R. McClean, Matt McEwen, Anthony Megrant, Xiao Mi, Masoud Mohseni, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Charles Neill, Florian Neukart, Murphy Yuezhen Niu, Thomas E. O'Brien, Bryan O'Gorman, Eric Ostby, Andre Petukhov, Harald Putterman, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Andrea Skolik, Vadim Smelyanskiy, Doug Strain, Michael Streif, Marco Szalay, Amit Vainsencher, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Leo Zhou, Hartmut Neven, Dave Bacon, Erik Lucero, Edward Farhi, and Ryan Babbush. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics*, 17(3):332–336, 2021.

[HWO+19] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.

[IBM21] IBM Quantum. IBM quantum platform. `https://quantum.ibm.com/`, 2021.

[IMM18] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*, pages 428–437, 2018.

[Jen67] G.F. Jenks. The data model concept in statistical mapping. *International Yearbook of Cartography*, 7:186–190, 1967.

[JL03] Richard Jozsa and Noah Linden. On the role of entanglement in quantum computational speed-up. *Proceedings of the Royal Society of London A*, 459(2036):2011–2032, 2003.

[JMSS23] Chris Jones, Kunal Marwaha, Juspreet Singh Sandhu, and Jonathan Shi. Random max-csps inherit algorithmic hardness from spin glasses. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

[JN17] J.S̃akurai J. and Jim Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 2017.

[Joz98] Richard Jozsa. Entanglement and quantum computation. In *The Geometric Universe: Science, Geometry, and the Work of Roger Penrose*. Oxford University Press, 1998.

[JRS07] Sabine Jansen, Markus B. Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48:102–111, 2007.

[JSW+24] Stephen Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, and Ryan Babbush. Optimization by decoded quantum interferometry, 2024. Google Research publication page.

[JSW+25] Stephen P. Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, Tanuj Khattar, and Ryan

Babbush. Optimization by decoded quantum interferometry, 2025. arXiv v4, 22 May 2025.

[JV03] Han Kim Jeong and H.Ṽu Van˙ Generating random regular graphs. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 213–222, 2003.

[Kar72] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[KKKS98] Lefteris M. Kirousis, Evangelos Kranakis, Danny Krizanc, and Yannis C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Struct. Algorithms*, 12(3):253–269, 1998.

[KL21] William M. Kirby and Peter J. Love. Variational quantum eigensolvers for sparse hamiltonians. *Physical Review Letters*, 127(11), 2021.

[KMRT$^+$07] Florent Krzakala, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborova. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

[LC19] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[Lin87] Nathan Linial. Distributive graph algorithms—global solutions from local data. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 331–335, 1987.

[Lin92] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[LLL20] Daniel Liang, Li Li, and Stefan Leichenauer. Investigating quantum approximate optimization algorithms under bang-bang protocols. *Physical Review Research*, 2:033402, 2020.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[LMS24] Jeffrey Larson, Matt Menickelly, and Jiahao Shi. A novel noise-aware classical optimizer for variational quantum algorithms, 2024.

[Loc22] Owen Lockwood. An empirical review of optimization techniques for quantum variational circuits, 2022.

[LSCA21] Xinwei Lee, Yoshiyuki Saito, Dongsheng Cai, and Nobuyoshi Asai. Parameters fixing strategy for quantum approximate optimization algorithm. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021.

[LSS+23] Danylo Lykov, Ruslan Shaydulin, Yue Sun, Yuri Alexeev, and Marco Pistoia. Fast simulation of high-depth QAOA circuits. In *Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, SC-W 2023. ACM, 2023.

[LTE25] Quinn Langfitt, Reuben Tate, and Stephan Eidenbenz. Phantom edges in the problem hamiltonian: A method for increasing performance and graph visibility for QAOA. In *Proceedings of the 2025 IEEE International Conference on Quantum Communications, Networking, and Computing (QCNC)*, pages 40–49, 2025.

[LTW+24] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and M. Cerezo. A review of barren plateaus in variational quantum computing, 2024.

[Luc14] Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2, 2014.

[MMZ05a] Stephan Mertens, Marc Mezard, and Riccardo Zecchina. Threshold values of random k-SAT from the cavity method, 2005.

[MMZ05b] M. Mezard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Phys. Rev. Lett.*, 94:197205, 2005.

[MS24] Raimel A. Medina and Maksym Serbyn. A recursive lower bound on the energy improvement of the quantum approximate optimization algorithm, 2024.

[MSFB25] Thorge Müller, Ajainderpal Singh, K.W̃ilhelm Frank and Tim Bode. Limitations of quantum approximate optimization in solving generic higher-order constraint-satisfaction problems. *Physical Review Research*, 7:023165, 2025.

[MSL92] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, page 459–465. AAAI Press, 1992.

[Nad23] Alexander Nadel. Solving huge instances with intel SAT solver. In *26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023)*, 2023.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011.

[Net23] NetworkX developer team. Networkx, 2023.

[PAA+23] Brenda Praggastis, Sinan Aksoy, Dustin Arendt, Mark Bonicillo, Cliff Joslyn, Emilie Purvine, Madelyn Shapiro, and Ji Young Yun. Hypernetx: A python package for modeling complex network data as hypergraphs, 2023.

[PBGE24] Elijah Pelofske, Andreas Bartschi, John Golden, and Stephan Eidenbenz. Scaling whole-chip QAOA for higher-order ising spin glass models on heavy-hex graphs. *npj Quantum Information*, 10:109, 2024.

[PJMS+24] Aidan Pellow-Jarman, Shane McFarthing, Ilya Sinayskiy, Daniel K. Park, Anban Pillay, and Francesco Petruccione. The effect of classical optimizers and ansatz depth on QAOA performance in noisy devices. *Scientific Reports*, 14(1), 2024.

[PMS+14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), 2014.

[Pow94] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*, 1994.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[Qis23] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

[Ric11] P.S̃tanley Richard˙ *Enumerative Combinatorics, Volume 1*. Cambridge University Press, 2011.

[Ric12] P.S̃tanley Richard˙ *Enumerative Combinatorics, Volume 2*. Cambridge University Press, 2012.

[SAT+24] Francesca Schiavello, Edoardo Altamura, Ivano Tavernelli, Stefano Mensa, and Benjamin Symons. Evolving a multi-population evolutionary-QAOA on distributed qpus, 2024.

[Sch01] Fabio Schoen. *Stochastic global optimization two-phase methods Two-Phase Methods*, pages 2475–2479. Springer US, Boston, MA, 2001.

[SG76] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society, 1994.

[SHS+24] Shree Hari Sureshbabu, Dylan Herman, Ruslan Shaydulin, Joao Basso, Shouvanik Chakrabarti, Yue Sun, and Marco Pistoia. Parameter setting in quantum approximate optimization of weighted problems. *Quantum*, 8:1231, 2024.

[SN99] Angelika Steger and C.W̃ormald Nicholas˙ Generating random regular graphs quickly. *Combinatorics, Probability and Computing*, 8(4):377–396, 1999.

[SP23] Ruslan Shaydulin and Marco Pistoia. QAOA with $n \cdot p \geq 200$, 2023.

[SSC+22] Vishal Sharma, Nur Shahidee Bin Saharan, Shao-Hen Chiew, Ezequiel Ignacio Rodríguez Chiacchio, Leonardo Disilvestro, Tommaso Federico Demarie, and Ewan Munro. Openqaoa – an SDK for QAOA, 2022.

[Str68] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5(3):506–517, 1968.

[Suz90] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many body theories. *Phys. Lett. A*, 146(6):319–323, 1990.

[SW20] Shay Solomon and Nicole Wein. Improved dynamic graph coloring. *ACM Transactions on Algorithms*, 16(3):41:1–41:24, 2020.

[Sze11] Richard Szeliski. Computer vision algorithms and applications, 2011.

[TCC+22] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[Tro59] H. F. Trotter. On the product of semigroups of operators. *Proc. Amer. Math. Soc.*, 10(4):545–551, 1959.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H.

Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[vR95] G. van Rossum. Python. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995.

[WHJR18] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A*, 97(2), 2018.

[WL21] Jonathan Wurtz and Peter Love. MaxCut quantum approximate optimization algorithm performance guarantees for p ¡ 1. *Physical Review A*, 103(4), 2021.

[WL24] Elisabeth Wybo and Martin Leib. Vanishing performance of the parity-encoded quantum approximate optimization algorithm applied to spin-glass models. *Quantum*, 8:1554, 2024.

[WSL25] Xiaoyang Wang, Yuexin Su, and Tongyang Li. Performance guarantees of light-cone variational quantum algorithms for the maximum cut problem. *arXiv preprint*, 2025.

[WZ82] William K. Wootters and Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

[ZLZ24] Hao-Kai Zhang, Shuo Liu, and Shi-Xin Zhang. Absence of barren plateaus in finite local-depth circuits with long-range entanglement. *Physical Review Letters*, 132:150603, 2024.

[ZTB+22] Linghua Zhu, Ho Lun Tang, George S. Barron, F. A. Calderon-Vargas, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. An adap-

tive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer, 2022.

[ZWC+20] 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), 2020.