Separating quantum and classical computing: rigorous proof and practical application

Marshall, S.C.

CHAPTER 5

All this for one Qubit?

5.1 Introduction

The near-term deployment of advantageous quantum algorithms is currently constrained by available hardware. Among other limitations, modern machines simply do not have enough qubits for the most interesting algorithms. In an attempt to augment modern machines, several cutting schemes [3–7, 7–11] have been proposed that partition a given quantum circuit into smaller blocks. Each block can be run independently and then combined to simulate the output of the original circuit. We refer to these techniques collectively as “Circuit Cutting" (CC). One application of these circuit-cutting schemes was given in the previous chapter.

While the potential value of these schemes is clear, their practical value is limited by their computational cost: the number of circuit evaluations required grows exponentially with the number of two-qubit gates between partitioned blocks. For many algorithms, each qubit requires a polynomial number of two-qubit gates connecting it to the rest of the circuit, preventing useful application of CC to these cases. Applications of CC are instead relegated to a secondary role, such as augmenting connectivity by adding virtual connections [5]. Broader application of CC can therefore only be achieved if the number of terms can be reduced, one such route is by a computational cost of a scheme that scales in e.g. number of qubits

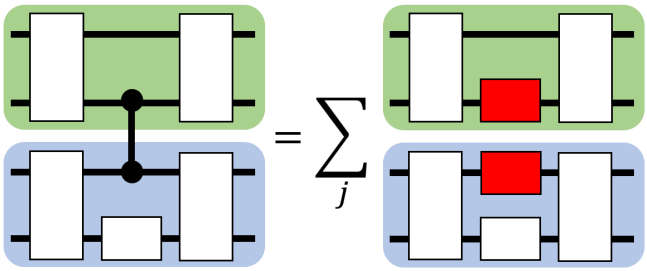

Figure 5.1: Circuit cutting schemes can be used to simulate computations with more qubits than a user has access to. Here a circuit cutting scheme rewrites two-qubit gates as sums of single-qubit gates, allowing the 4-qubit circuit to be expressed as a sum of tensor product two-qubit circuits, those two-qubit circuits can then be evaluated on a smaller machine. This chapter focuses only on "cut-local" schemes, i.e. ones that only modify the circuit at the sites of partition-crossing two-qubit gates, replacing the gates with a sum of single-qubit unitaries.

removed, instead of two qubit gates cut.

While it is clear that some partitions will require a super-polynomial number of circuit evaluations if quantum computers provide a computational advantage, **BPP≠BQP** (a simple example of this is given in the footnote [1]), this super-polynomial requirement could still produce achievable runtimes if it scaled in a different parameter, such as the number of qubits in the blocks of the partition. In some ways, the size of the largest block is a more natural scaling parameter, for instance, the limited Hilbert space dimension limits the maximum amount of entanglement between blocks (a key ingredient in quantum advantage [87, 88]).

The size of the Hilbert space is a basis for the proofs of circuit cutting scaling requirements in [7, 9], where it is shown that an exponential number of terms are needed to simulate large and highly entangled states.

While this argument bounds circuit-cutting techniques which scale with number of two-qubit gates, it does not, for example, bound CC schemes which take into account fixed inputs or scale in other parameters [8].

[1] Cutting the circuit in half, then repeatedly cutting the subcircuits generated by this cut in half would require only $\lceil(\log_2(n))\rceil$ rounds to reduce an n-qubit circuit to a combination of 1-qubit circuits. If each round produces at most A subcircuits only $A^{\log(n)}$ circuit evaluations are needed to reduce the n-qubit circuit can be reduced to a polynomial-sized set of classically simulatable 1-qubit circuits. If the cutting procedure produced less than some pseudo-polynomial number of terms each time, A, then a classic simulator exists using pseudo-polynomial time.

This chapter investigates if there could exist such a circuit cutting scheme: one whose computational overhead would scale polynomially with the number of inter-partition gates, at the cost of an exponential scaling in the size of the smallest partitioned block. We show that when the circuit cutting scheme is limited to local modifications (i.e. it can only modify partition-crossing two-qubit gates; removing this assumption would make any formal statements dramatically more difficult to prove [2]) any circuit cutting scheme that can efficiently remove a single qubit (create a $(1, (n-1))$ partition) would imply **BPP**=**BQP**. These results demonstrate that circuit-cutting schemes can not be broadly efficient, regardless of what tricks are applied, and that they do not represent a shortcut to quantum advantage. In relation to this thesis, this chapter demonstrates that the heuristic circuit-cutting-esque machine learning algorithm given in the previous chapter cannot work on all circuits (when phrased as machine learning problems).

5.2 Background

Circuit cutting (CC) schemes [3–7, 7–11] (sometimes referred to as circuit partitioning or circuit knitting) are a class of methods designed to reduce the demands on a quantum computer when trying to implement a large quantum circuit. Existing schemes either make multiple calls to the device [3–6] or link multiple devices with classical communication [7, 9] to simulate the larger circuit.

Each of these circuit-cutting schemes works slightly differently, for simplicity, we will focus on a generalization of the formalism presented in [3]. This generalization involves decomposing a unitary into a number of smaller unitaries. Here, decomposing means expressing a given n-qubit unitary as a sum of tensor products of two fewer-qubit unitaries:

$$U = \sum_{i}^{L} \alpha_i U_i' \otimes U_i'', \tag{5.1}$$

We discuss how our results generalize to alternative schemes (e.g. ones that decompose tensor-networks [4] or superoperators [5]) later in this chapter.

[2] As we discuss later, local schemes effectively mean we do not allow significant semantic-preserving circuit rewritings; allowing circuit rewritings is more powerful, but also leads to the (NP-hard) problems of finding minimal or otherwise simplest circuits, making proofs exceptionally difficult. Hence we focus on the simpler case here.

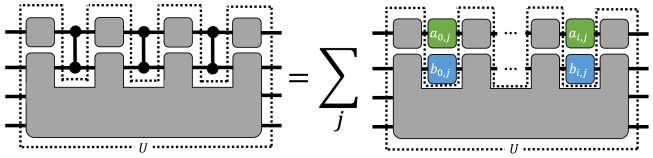

Figure 5.2: Figure depicts a cut-local circuit cutting scheme being applied to our quantum comb formalism. The comb is designed to specify the whole circuit except for some "gaps", which are then filled by input gates, in this case, Controlled-Z gates, that can add connectivity between the first qubit and the other $(n-1)$-qubits. When a cut-local circuit cutting scheme is applied to the circuit, the connecting two-qubit gates in the gaps become single-qubit operators. The resulting unitary is now a tensor-product allowing the first qubit and the other $(n-1)$-qubits to be run separately, reducing the required width of quantum computer. To go beyond cut-local schemes, one would allow U to depend on j.

All these existing approaches to circuit cutting have a drawback: they incur a super-polynomial scaling in the number of connections, k, crossing the partition, i.e. $L \geq c^k$ for some $c \geq 2$. This chapter addresses an intriguing question: could there exist a scheme capable of partitioning an n-qubit circuit into an m-qubit block and an $(n-m)$-qubit block with super-polynomial scaling only in m, while remaining polynomial in k and n, in other words, we seek a scaling $L \in O(c'^m \times \text{poly}(n,k))$. We find that even considering the simplest case, where $m = 1$, is sufficient to show it is not possible.

Conflicting intuitions surround the possibility of a $\text{poly}(n,k)$ scheme for the $m = 1$ case. On one hand, the addition of a qubit doubles the dimension of the relevant Hilbert space, making it unclear how to simulate the larger Hilbert space with access to only the smaller one. On the other hand, existing limitations of circuit cutting rely on simulating states with substantial entanglement between the blocks to show that $L \geq 2^k$. However, in the $m = 1$ case, this entanglement is heavily limited, breaking the assumptions behind these limitations. Notably, if we simplify the goal to expressing equation 5.1 as a sum of *arbitrary linear operators*, it clearly only requires 4 terms to satisfy equation 5.1 [3]. This chapter resolves these uncertainties by showing that when the circuit cutting scheme is only allowed to make local modifications, an exponential number of terms in k is necessary, regardless of how many qubits are removed.

[3] This is the upper bound on the Schmidt rank when partitioning a 2-dimensional subspace.

This brings us to a key property shared by all existing circuit cutting schemes, which we refer to as "cut-locality". The idea behind cut-locality is that when modifying a gate the resulting subcircuit only differs at the site of the removed gate. In [3] this is a two-qubit gate replaced locally as the sum of one-qubit gates (see Figure 5.1).

To formally treat cut-local schemes it is useful to use a variation of the quantum comb formalism [89]. We are only interested in using the formalism to partition a single qubit from our circuit, thus we will slightly modify the definition of a quantum comb. Informally our quantum comb is a unitary with G "gaps" where gates can be plugged in, the unitary does not have connections between the first qubit and the rest of the qubits but by putting in two-qubit gates to these gaps we can create a connected unitary. An example is shown in Figure 5.2. We define the quantum comb as a map, $U(\cdot)$, taking in two-qubit unitaries, G_i, and returning a fixed n-qubit unitary with G_i in the gaps as described.

Our ultimate goal is to bound cut-local schemes, which would transform a quantum comb with entangling two-qubit gate arguments into the sum of quantum combs with tensor product arguments:

$$U(\ldots, G_j, \ldots) = \sum_{i=0}^{L} \alpha_i U(\ldots, a_{i,j} \otimes b_{i,j}, \ldots) \tag{5.2}$$

where $a_{i,j}, b_{i,j} \in SU(2)$ (single qubit unitaries), $\alpha_i \in \mathbb{C}$ (some coefficient) and $G_j \in SU(4)$ (two-qubit gates). If a given quantum comb with given two-qubit unitary inputs can be represented as the sum of quantum combs with tensor product inputs in at most L terms, as in equation 5.2, we say there exists an *L-term partitioned quantum comb* representation.

We have yet to define whether the quantum comb is equal to a specific computation, with known input and observable (i.e. classically specified and fixed), or to the unitary itself (so the decomposition does not benefit from considering specific input states or measurements, and must correctly apply to all inputs and measurements). The following paragraphs will address both of these cases; we show the unitary case is simple via linear algebra, the fixed-input-output case is more challenging, requiring a complexity-theoretic argument.

5.3 Bounds on the optimal scheme

Our results are structured into two groups: The first group, Lemma 5.3.1 and Theorem 5.3.1, shows that when the input and observable are fixed

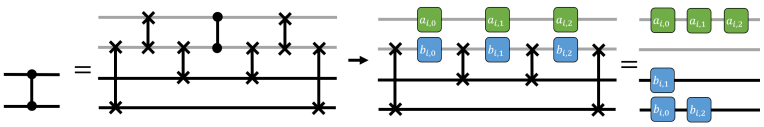

Figure 5.3: Demonstration of a gadget used in proofs of theorems 5.3.1 and 5.3.2. The gadget adds two ancilla qubits (shown in grey) to move any non-SWAP two-qubit gate to these two ancilla qubits. Applying any local circuit cutting scheme to separately partition the first qubit then leaves a circuit of only SWAP and single-qubit gates. This can then be classically rewritten in polynomial time as a sum of circuits of single-qubit unitaries. To evaluate this circuit classically requires only polynomial time, this is used to reach a contradiction to prove theorem 5.3.1. Polynomial sums of single-qubit unitaries can only have polynomial Schmidt rank, this is used to reach a contradiction to prove theorem 5.3.2.

it is technically possible to partition a single qubit from the rest of the circuit in polynomially many terms, but that if such a decomposition could be found in polynomial time then **BPP**=**BQP**. Our second group, centered on Theorem 5.3.2 shows that even with unlimited runtime a local circuit cutting scheme cannot find a decomposition of a given circuit in polynomially many terms if we force the same decomposition to apply for every input and observable.

Our first result, showing that a decomposition can be found, is only true in a technical sense and not informative to practical use.

Lemma 5.3.1
Given a circuit expressed as a quantum comb with fixed gates, $U(G_1, \ldots)$, a known (i.e. there is a known succinct classical description) input, $|\phi\rangle$ and known observable, M, the associated expectation value can be expressed with a 1-term partitioned quantum comb in the same expectation value:

$$\langle\phi| U^\dagger(G_1, \ldots) M U(G_1, \ldots) |\phi\rangle = \tag{5.3}$$

$$\alpha_0 \langle\phi| U^\dagger(a_{0,0} \otimes b_{0,0}) M U(a_{0,0} \otimes b_{0,0}) |\phi\rangle\,, \tag{5.4}$$

where $a_{0,0}, b_{0,0} \in SU(2)$ (single qubit unitaries), $\alpha_0 \in \mathbb{C}$ (coefficient).

Proof. The outcome of equation 5.3 is equal to some real number, γ. If $\gamma = 0$, set α_0 to 0 and $a_{0,0}$ and $b_{0,0}$ to any single qubit gates. The equality holds proving this case.

If $\gamma \neq 0$, by [3] it is known that there must exist some input $a_{0,0}$ and $b_{0,0}$ that produces a non-zero output of equation 5.4. This non-zero output

can then be rescaled with some $\alpha_0 \in \mathbb{C}$ to achieve the equality. □

It is easy to see that if a scheme existed to produce this 1-term partitioned quantum comb, then we could iteratively apply it to the whole circuit and obtain a classical algorithm with polynomial runtime to compute any quantum circuit, implying **BQP** = **BPP**. The following theorem then demonstrates that *finding* this partition must in some way contain the hardness of **BQP**, indeed it shows that the existence of any polynomial-time circuit cutting algorithm capable of finding this polynomial termed partitioned quantum comb would imply **BQP** = **BPP**.

Theorem 5.3.1
If there exists a polynomial time classical algorithm that takes an arbitrary input circuit expressed as a quantum comb with fixed gates, $U(G_1, \ldots)$, a known input $|\phi\rangle$ and a known observable M, and returns the arguments of an L-term partitioned quantum comb, $a_{i,j}, b_{i,j} \in SU(2), \alpha_i \in \mathbb{C}$ for $L \in poly(k, n)$, such that:

$$\langle\phi| U^\dagger(G_1, \ldots) M U(G_1, \ldots) |\phi\rangle =$$
$$\langle\phi| \sum_i^L \bar{\alpha_i} U^\dagger(a_{i,0} \otimes b_{i,0}, \ldots) M \sum_i^L \alpha_i U(a_{i,0} \otimes b_{i,0}, \ldots) |\phi\rangle$$

then **BQP** = **BPP**.

Proof. Given an algorithm that can partition a single qubit as described (call this algorithm $\mathcal{A}$) we provide an efficient classical simulator.

Given an n-qubit input circuit V written in some standard gate set (e.g. H, CNOT, T), replace every existing two-qubit gate anywhere in V with the gadget shown in Figure 5.3, creating a circuit of only swap gates and single-qubit gates everywhere except for arbitrary 2 qubit gates between the first 2 qubits.

Applying $\mathcal{A}$ to separate the top qubit produces a circuit of only SWAP and single qubit gates, which is classically simulatable in poly(n) time [1]. □

This is our main result: put simply, local schemes cannot partition even a single qubit without paying an exponential cost somewhere. Extending this result to the impossibility of separating l qubits is just a matter of "padding" the gadget with $l - 1$ qubits which do not interact with the rest of the circuit. Note that this result transfers to the task of approximating (rather than exactly recreating) the output with an L-term partitioned quantum comb as **BQP** is robust to approximations of outputs. We will

also describe how this result can be extended to other local circuit cutting schemes in following paragraphs.

While the condition **BQP** $\neq$ **BPP** is a reasonable requirement, it is not clear if it is necessary. We show that by forcing one decomposition to apply for all inputs and measurements (which is equivalent to demanding the unitaries are the same, and thus could be derived from existing work such as [7]) we can show unconditionally that it is impossible to partition one qubit from an n qubit circuit with only polynomially many terms.

Theorem 5.3.2
There exist quantum circuits expressible as a quantum comb with input gates, $U(G_1, \ldots)$, such that for all $L \in O(poly(n))$ and inputs $\alpha_i \in \mathbb{C}, a_{i,j}, b_{i,j} \in SU(2)$,

$$U(G_1, \ldots) \neq \sum_{i}^{L} \alpha_i U(a_{i,0} \otimes b_{i,0}, \ldots).$$

Proof. Define C as a circuit that generates $n/2$ Bell pairs from the $|0\rangle^{\otimes n}$ state (if n is odd, generate $n-1$ bell pairs). We can apply the rewriting gadget in Figure 5.3 to C, call this new circuit C'.

Assume towards contradiction that there exists a polynomial-L term quantum comb implementing the same unitary as C', by separating the first qubit we have created a sum of L tensor product circuits. As shown in Figure 5.3, each of these circuits acts locally on every qubit, thus each individual circuit has an operator Schmidt rank of 1 across any partition. Summing over L tensor product terms produces an operator of Schmidt rank at most L. Applying this circuit to the Schmidt rank 1 input state, $|0\rangle^{\otimes n}$, we produce an output state of Schmidt rank at most L, but $n/2$ Bell pairs require a Schmidt rank of at least $2^{n/2}$ [7], which is a contradiction. □

As with the previous theorem, the proof of Theorem 5.3.2 also extends to the case of *approximating* a unitary easily; the fidelity between the closest $poly(n)$-Schmidt rank state and the $n/2$-Bell-pairs state decays exponentially in n. This implies that the operator distance (and diamond-norm distance) between U and any polynomial sum approximating U also becomes maximal in n.

5.4 Generalizations to other schemes

Our results have bounded how any locally acting unitary-based circuit cutting schemes can perform, but we have said relatively little about how a general scheme (one which can express circuits as the sum of other circuits of any form) may perform. It is therefore important to determine how broadly our results apply. In this section, we generalize our framework to encompass other locally acting circuit cutting schemes and discuss how apparently promising routes to generalize to non-local circuit cutting schemes do not work out.

To generalize our technique to other locally acting circuit cutting schemes note that the choice to decompose a unitary into other unitaries, while useful for illustration, was not maximally general. Instead, we can consider decomposing the *channel* associated to that unitary, $\mathcal{U}$, into other channels:

$$\mathcal{U} = \sum_{i}^{L} \alpha_i \mathcal{C}_i \tag{5.5}$$

$\mathcal{C}_i$ now respect an analogous cut locality condition. If a scheme obeys this locality condition (as [5] does) then the gadget can be applied to convert a connected circuit into a tensor product, allowing for classical simulation and extending Theorem 5.3.1 to this case. Even classical augmentation of the channel (e.g. with classical communication [7, 9]) would not break this simulability argument, further extending our results to this case.

It is less clear how the circuit cutting schemes that cut qubits time-wise (i.e. decompose identity channels [4, 9]) fit into this framework. The time-like circuit cutting schemes can be used to create partitioned blocks by cutting qubits that appear in two otherwise disconnected blocks. The choice of which qubits to cut is not immediately clear in our problem (which is to reduce the hardware requirements by just one qubit), instead we must try and find the analogous problem. If we only allow modifications outside the quantum comb, but still require blocks of at most $(n-1)$-qubits then the only option is to decompose local channels on the 2^{nd} qubit. In this case, our results extend.

Extending these results even further, to non-local (i.e. unrestricted) circuit cutting schemes, generally becomes much more challenging. The question now runs into issues of deciding the minimum circuit size necessary to implement a given function, related to the famously opaque minimum circuit size problem and its quantum analog [90]. Fortunately, existing schemes operate using only local cuts, making this question less relevant.

Finally, we wish to address an ostensible link between bounds on non-

local circuit cutting and the one clean qubit model [91]. The one clean qubit model is a restricted computational model consisting of an arbitrary circuit taking input of one clean qubit in some fiducial state and $(n-1)$ maximally mixed qubits. If one applies the types of circuit cutting methods discussed in this chapter to achieve an $(1, n-1)$ partition, one may come to the conclusion that the $(n-1)$-qubit computations will be acting on the maximally mixed states, which is classically simulatable. In this case, if the circuit cutting results in just polynomially many terms, the entire circuit cutting computation would be weakly simulatable (we can sample the output of the circuit), which would collapse the polynomial hierarchy, **PH**=**AM** [92], which is widely believed not to hold. This would constitute a rather elegant general no-go result for circuit cutting methods achieving a sub-exponential number of terms. However, the argument fails as circuit cutting does not necessarily apply a sum of just unitary channels to the maximally mixed input (e.g. in [3] different unitaries might be multiplied to the left and right side of the state, which is not a unitary channel and doesn't preserve the classical simulability of the maximally mixed state) or necessarily compute a $(n-1)$-qubit circuit on a subsystem of just the original (maximally mixed) input. Indeed this argument can be modified to show a slightly more general result: that all circuit cutting schemes must apply non-unital channels (which contain unitary channels), regardless of the number of terms generated (exponential or otherwise).

Corollary 5.4.1
No circuit cutting scheme can decompose any given unitary, U, on a given partition into a finite sum of only unital channels.

The proof (provided in the supplementary material) functions by using two SWAP gates to swap a clean qubit into a maximally mixed block.

5.5 Conclusion

In this chapter we have analysed the limits of locally acting circuit cutting schemes' ability to partition whole qubits. We found that for polynomially many two-qubit gates between the single qubit and the rest of the circuit, no locally acting scheme can achieve polynomial efficiency. We discussed how these results can be extended into other locally acting circuit cutting schemes, such as the superoperator or tensor network formalisms and suggested that they may apply to all local schemes.

This chapter suggests a clear future research direction: to either generalize these results to non-local circuit cutting schemes, or to attempt to

utilise some of the intuitions generated here to produce an efficient scheme for removing a single qubit from a circuit.