Quasi-polynomial Time Approximation of Output Probabilities of Constant-depth, Geometrically-local Quantum Circuits

Nolan J. Coble*†, Matthew Coudron‡

December 11, 2020

Abstract

We present a classical algorithm that, for any 3D geometrically-local, constant-depth quantum circuit C, and any bit string $x \in \{0,1\}^n$, can compute the quantity $|\langle x| C |0^{\otimes n}\rangle|^2$ to within any inverse-polynomial additive error in quasi-polynomial time. It is known that it is $\#P$-hard to compute this same quantity to within 2^{-n^2} additive error [Mov20]. The previous best known algorithm for this problem used $O(2^{n^{1/3}}\mathsf{poly}(1/\epsilon))$ time to compute probabilities to within additive error ϵ [BGM20]. Notably, the [BGM20] paper included an elegant polynomial time algorithm for the same estimation task with 2D circuits, which makes a novel use of 1D Matrix Product States carefully tailored to the 2D geometry of the circuit in question. Surprisingly, it is not clear that it is possible to extend this use of MPS to address the case of 3D circuits in polynomial time. This raises a natural question as to whether the computational complexity of the 3D problem might be drastically higher than that of the 2D problem. In this work we address this question by exhibiting a quasi-polynomial time algorithm for the 3D case. We believe that our algorithm extends naturally to any fixed dimension D by induction on the dimension, but we focus on the 3D case, as the simplest unresolved case, for concreteness. Furthermore, we show that, under a natural, polynomial-time-checkable condition on the circuit C, our algorithm runs in polynomial time. This highlights the possibility that the super-polynomial worst-case time bound on our algorithm might be due to limitations of our analysis. In order to surpass the technical barriers encountered by previously known techniques we are forced to pursue a novel approach: Constructing a recursive sub-division of the given 3D circuit using carefully designed block-encodings.

Our algorithm has a Divide-and-Conquer structure, demonstrating how to approximate the desired quantity via several instantiations of the same problem type, each involving 3D-local circuits on at most half the number of qubits as the original. This division step is then applied recursively, expressing the original quantity as a weighted sum of smaller and smaller 3D-local quantum circuits. A central technical challenge is to control correlations arising from the entanglement that may exist between the different circuit "pieces" produced this way. We believe that the division step, which makes a novel use of block-encodings [GSLW19], together with an Inclusion-Exclusion style argument to reduce error, may be of interest for future research on low depth quantum circuits.

*Authors are listed alphabetically.

†ncoble@terpmail.umd.edu

‡mcoudron@umd.edu - Corresponding Author

1 Introduction

Many schemes for obtaining a quantum computational advantage with near-term quantum hardware are motivated by mathematical results proving the computational hardness of sampling from near-term quantum circuits. In this work we consider quantum circuits which are geometrically local and have constant circuit-depth. It is known to be $\#P$-hard to compute output probabilities of n-qubit circuits of this type to within 2^{-n^2} additive error [Mov20], a result which builds on an extensive line of research focusing on the hardness of sampling from quantum circuits [AA11, BJS11, BMS17, NSC+17, BFNV19]. It has even been shown, under several computational assumptions, that there is no classical polynomial time algorithm which, given a constant-depth, geometrically-local quantum circuit, K, can produce samples whose distribution lies within a constant, in the ℓ_1 distance, of the output distribution of K in the computational basis [BVHS+18].

On the other hand, a series of works on the classical complexity of sampling from near-term quantum circuits, and related tasks, highlights the subtle nature of identifying an actual quantum advantage based on these tasks [DHKLP20, HZN+20, NPD+20]. These results frame the significance of the algorithm presented as Theorem 5 in [BGM20], which estimates output probabilites of 2D-local constant depth circuits to inverse polynomial additive error in polynomial time. In fact, the original algorithm in [BGM20], actually estimates quantities of the form $\langle 0^{\otimes n}| C^\dagger \left(\otimes_{i=1}^n P_i\right) C |0^{\otimes n}\rangle$, where each $P_i \in \{X, Y, Z, I\}$ is a single-qubit Pauli observable operator. However, it is straightforward to convert their algorithm to compute the quantity $\langle 0^{\otimes n}| C^\dagger \left(\otimes_{i=1}^n |x_i\rangle\langle x_i|\right) C |0^{\otimes n}\rangle = |\langle x| C |0^{\otimes n}\rangle|^2$, $x \in \{0,1\}^n$, instead. Theorem 5 of [BGM20] constitutes a pertinent observation. While it is hard to sample from constant-depth quantum circuits, it is still unresolved whether it is hard to estimate any property of such a circuit which could have been computed using a polynomial number of samples from the output of the quantum circuit itself. In particular: A polynomial number of samples from a 2D-local, constant-depth quantum circuit only allows one to estimate output probabilites of that circuit to inverse polynomial additive error. But, it is shown in Theorem 5 of [BGM20] that this same task can be done in classical polynomial time! One might ask: Is there a well-defined Decision problem which can be solved using only a polynomial number of samples from such a quantum circuit, together with classical post-processing, and yet cannot also be efficiently solved using classical computing alone? This is unknown.

We note, at this point, some basic facts about the task of computing the quantity $|\langle 0^{\otimes n}| C |0^{\otimes n}\rangle|^2$ which explain why we can focus on this task WLOG, and may motivate our interest in it:

- If there is an algorithm to estimate the quantity $|\langle 0^{\otimes n}| C |0^{\otimes n}\rangle|^2$, for any 3D-local constant-depth circuit C, then that algorithm can be used to estimate $|\langle x| C |0^{\otimes n}\rangle|^2$ for any $x \in \{0,1\}^n$. The reason is that $|\langle x| C |0^{\otimes n}\rangle|^2 = |\langle 0^{\otimes n}| G |0^{\otimes n}\rangle|^2$ where G is taken to be the 3D-local constant-depth circuit $G \equiv C\left(\otimes_{i=1}^n X^{x_i}\right)$. Here X represents the single qubit Pauli operator σ_X.
- Any such algorithm can also estimate $|\langle 0^{\otimes n}| CZ^n C^\dagger |0^{\otimes n}\rangle|^2$, which is the magnitude of the expected bias of the Parity of the output bits of C, when measured in the computational basis. This is true by virtue of the fact that $CZ^nC^\dagger$ is, itself, a 3D local, constant-depth circuit. So, this type of computational problem allows us to study the power of constant-depth, geometrically-local, quantum circuits combined with certain limited types of classical post-processing, like the Parity function.
- The algorithm we present in this work can easily be modified to approximate marginal probabilities (e.g., the probability that $x_1 = 1$ for $x \in \{0,1\}^n$ sampled from the given circuit,

etc). Consequently, it is straightforward to use this algorithm to search for all $x \in \{0,1\}^n$ which have probability at least δ in the output distribution of a given constant-depth, geometrically-local circuit C. That is, searching for all of the "δ-heavy" strings of C. When $\delta = 1/\mathsf{poly}(n)$ there can be at most $\mathsf{poly}(n)$ such strings and our algorithm can find them all in quasi-polynomial time.

The algorithm for 2D circuits presented in Theorem 5 of [BGM20] makes a novel use of 1D Matrix Product States, carefully tailored to the 2D geometry of the circuit in question. However, the authors of [BGM20] point out that it is not clear that it is possible to extend this use of MPS to address the case of 3D circuits in polynomial time. Instead they provide a sub-exponential time algorithm for the 3D case, which has time complexity $O(2^{n^{1/3}}\mathsf{poly}(1/\epsilon))$ for computing the desired quantity to within additive error ϵ. In this work we introduce a new set of techniques culminating in a divide-and-conquer algorithm which solves the 3D case in quasi-polynomial time.

Our algorithm has a divide-and-conquer structure with the goal being to divide the circuit C into pieces, and reduce the original problem to a small number of new 3D-circuit problems involving circuits on only a fraction of the number of qubits as the original. This division step requires the ability to construct Schmidt vectors of the state $C\,|0^{\otimes n}\rangle$, across a given cut, via a constant-depth geometrically-local quantum circuit, so that the new subproblems can be expressed as smaller instantiations of the original problem type. We accomplish this through the use of block-encodings, a technique designed for quantum algorithms [GSLW19], but used here as a subroutine of a classical simulation algorithm instead. However, to date, we are only able to construct, as a block-encoding circuit, the *leading* Schmidt vector across certain "heavy" cuts. Due to this restriction we are forced to use a novel division step in our Divide-and-Conquer approach. Instead of dividing about a single cut and constructing many of its Schmidt vectors as constant-depth geometrically local block-encodings, we must divide across many cuts and construct only their leading Schmidt vectors. Interestingly, this process can still lead to low approximation error via an Inclusion-Exclusion style argument, as shown in Lemma 18.

These techniques culminate in a worst-case quasi-polynomial time algorithm for 3D circuits, which is our main result:

Theorem 1. *Let C be any depth-d, 3D geometrically local quantum circuit on n qubits. Algorithm 1, $\mathcal{A}_{full}(C,\mathcal{B},\delta)$ (where $\mathcal{B}$ is the base case algorithm, chosen to be the algorithm in Theorem 5 of [BGM20]) will produce the scalar quantity $|\langle 0^{\otimes n}|\,C\,|0^{\otimes n}\rangle|^2$ to within δ error in time*

$$T(n) = 2^{\mathsf{polylog}(n)(1/\delta)^{1/\log^2(n)}} \cdot 2^{d^3} \tag{1}$$

(See technical abstract for the precise definition of Algorithm 1, $\mathcal{A}_{full}(C,\mathcal{B},\delta)$.)

Note that, for any $\delta = \Omega(1/n^{\log(n)})$, we have $(1/\delta)^{1/\log^2(n)} = O(1)$, and thus $T(n)$ is quasi-polynomial. In particular, for any $\delta(n)$ which scales inverse-polynomially (or even for some inverse-quasi-polynomial scaling), the algorithm runs in quasi-polynomial time. Furthermore, under a natural, polynomial-time-checkable assumption on the circuit C (see Assumption 33), we obtain a polynomial time algorithm:

Theorem 2. *Let C be any depth-d, 3D geometrically local quantum circuit on n qubits. If C satisfies Assumption 33, then Algorithm 3, $\mathcal{A}_{const}(C,\mathcal{B},\delta)$ (where $\mathcal{B}$ is the base case algorithm chosen to be as in Theorem 5 of [BGM20]), will approximate the scalar quantity $|\langle 0^{\otimes n}|\,C\,|0^{\otimes n}\rangle|^2$ to within δ additive error in time*

$$T(n) = \mathsf{poly}(n, 2^{(1/\delta)^{1/\log^2(n)}}) \cdot 2^{d^3}. \tag{2}$$

(See technical abstract for the precise definition of Algorithm 3, $\mathcal{A}_{const}(C,\mathcal{B},\delta)$.)

Note that, for any $\delta = \Omega(1/n^{\log(n)})$, we have $(1/\delta)^{1/\log^2(n)} = O(1)$, and thus $T(n)$ is polynomial.

2 Dividing the Cube: Some Notation

Given a 3D-local, constant-depth circuit C, we wish to estimate the quantity $|\langle 0^{\otimes n}| C |0^{\otimes n}\rangle|^2$. To begin our divide-and-conquer approach we will divide the circuit C in half via a cut through the center as shown in Figure 1. The width of the cut is a constant, and we will discuss how to select the constant below. To begin with, we make the constant large enough to have the non-empty sets F, M, and B defined below.

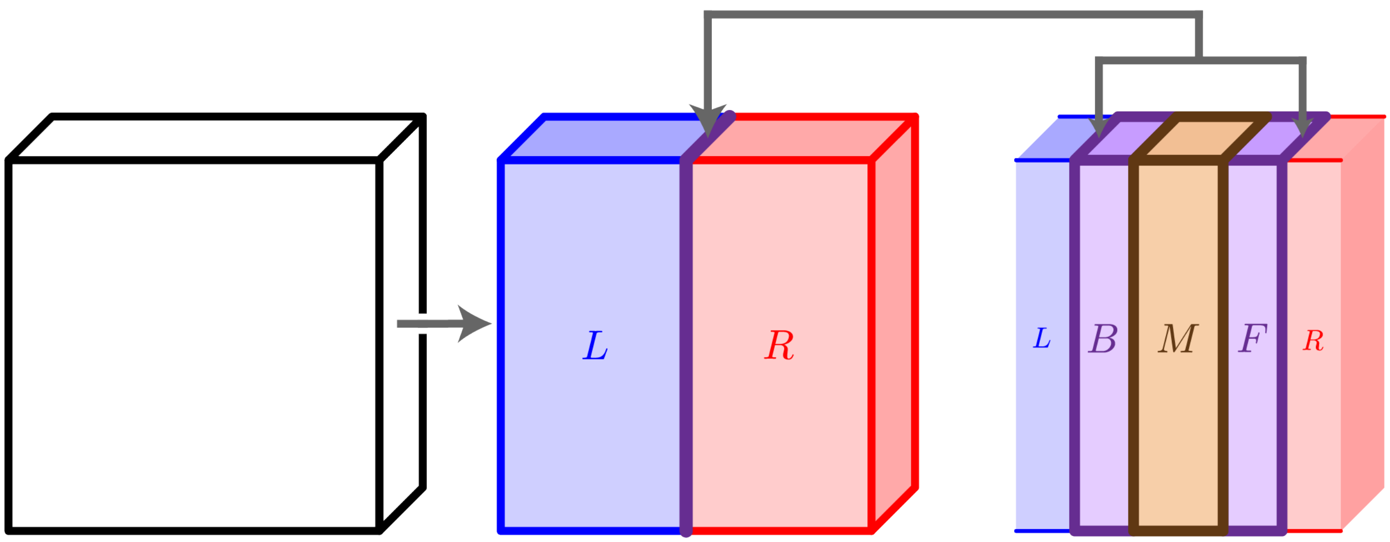

Figure 1: Cutting the Cube: (left) 3D cube of qubits. (center) Choose a location to cut the qubits. The qubits to the left and right of the cut are denoted by L and R, respectively. (right) Within the cut there are three regions: a center region M and regions to the left and right of M, denoted by B and F, respectively.

Definition 3 (M, B, F, R, and L (see Figure 1))**.** Let M be the set of all qubits in the "Middle of the cut" (the middle part of the cut which is not in the lightcone of qubits from outside the cut. Let B be the set of all qubits within the cut which are to the left of M. Let F be the set of all qubits within the cut which are to the right of M. We will choose the width of M to be a constant such that the lightcones of B and F are disjoint. We will choose the widths of B and F to be constants such that the lightcone of M is contained in $B \cup M \cup F$. For concreteness we set the width of each of B, M, F to be $10d$, where d is the (constant) depth of the given circuit C. Since C is geometrically-local, this is sufficiently large width to satisfy the above conditions on lightcones.

Let L be all qubits outside the cut which are to the left of the cut(that is, to the left of B). The set L is colored blue. Let R be the set of all qubits outside to the right of the cut (that is, to the right of F). The set R is colored red.

We will now define a 2D geometrically local, constant-depth circuit $C_{B\cup M\cup F}$ which can be thought of as the sub-circuit of C which lies within the light-cone of M. Intuitively this circuit captures all of the local information that must be accounted for in the division step across this particular slice in our divide-and-conquer algorithm.

Definition 4 ($C_{B\cup M\cup F}$)**.** Now, let us begin with the all zeroes state on all the qubits $|0\rangle_{L\cup B\cup M\cup F\cup R} = |0_{ALL}\rangle$, and apply the minimum number of gates from the circuit C such that every gate on the qubits within M has been applied. We will call this unitary $C_{B\cup M\cup F}$. Note that this unitary does not act on any qubits outside of $B\cup M\cup F$. This is because the lightcone of M is contained in $B\cup M\cup F$ by Definition 3. Note that $C_{B\cup M\cup F}$ can be thought of as a 2D (not 3D) geometrically local, constant depth circuit, since the third dimension of the circuit is constant-width and does not grow in size with n.

We define $C_{L\cup R}$ to be the unitary composed of the remainder of the gates of C not yet applied in $C_{B\cup M\cup F}$, so that $C = C_{L\cup R}\circ C_{B\cup M\cup F}$. We define C_L (resp. C_R) to be the unitaries composed of the remainder of the gates of C not yet applied in $C_{B\cup M\cup F}$ and which lie to the left (resp. right) of the M. Note that $C_L\circ C_R = C_{L\cup R}$ since none of the circuits $C_L, C_R, C_{L\cup R}$ act non-trivially on M. See Figure 2 for an illustration of these unitaries with a 1D geometrically-local circuit, and Figure 3 for an illustration in a 2D circuit.

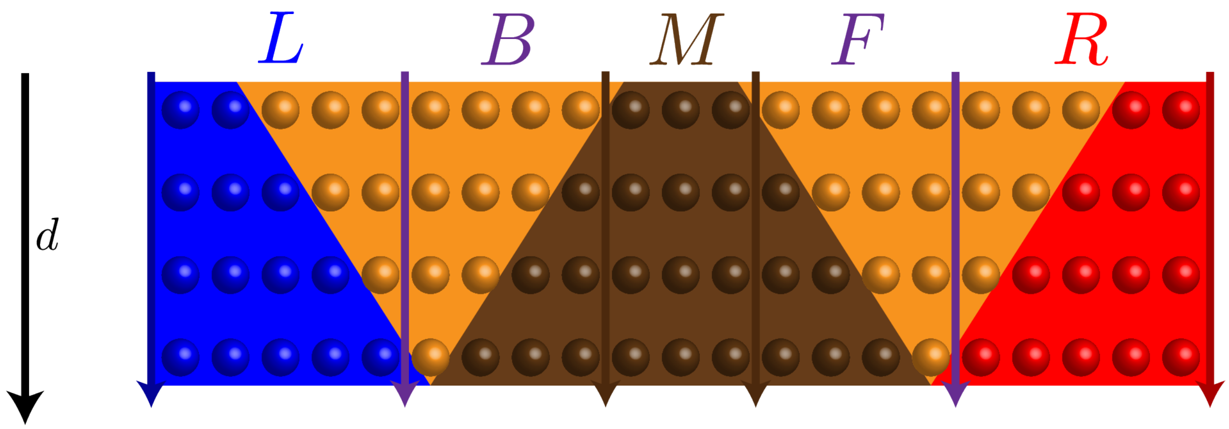

Figure 2: Block depiction of the unitaries defined in Definition 4 and Definition 6 for the case of a 1D geometrically-local constant-depth circuit C. Here the vertical dimension represents the depth of the circuit, so the rectangle has the same dimensions as the circuit diagram would. $C_{B\cup M\cup F}$ consists of all gates that act within the inner trapezoidal block. The unitaries defined in Definition 6 are also depicted here: $C_{wrap} \equiv C_{L-Wrap}\otimes C_{R-Wrap}$ consists of all gates that act within the triangular blocks. C_{L-Wrap} (resp. C_{R-Wrap}) consists of just those gates within the left (resp. right) triangular block. Furthermore, C_L' (resp. C_R') denote the unitaries consisting of all gates that act within the left (resp. right) trapezoidal block. We also illustrate the 2D case in Figure 3 below. We do not have an analogous figure for the 3D circuits that we are, in fact, analyzing in this work, because it would require 4 dimensions to illustrate. However, we believe the reader will understand the definitions from the 1D and 2D diagrams.

The sub-normalized quantum state produced by $C_{B\cup M\cup F}$, defined below, is the state whose Schmidt decomposition we consider in our division step.

Definition 5. Let $|\psi\rangle_{B\cup F} \equiv \langle 0|_M\, C_{B\cup M\cup F}\, |0\rangle_{B\cup M\cup F}$.

Note that, $\langle 0|_{\mathrm{ALL}}\, C_{L\cup R}\, |0\rangle_{L\cup R}\otimes|\psi\rangle_{B\cup F} = \langle 0|_{\mathrm{ALL}}\, C\, |0\rangle_{\mathrm{ALL}}$.

(Throughout this document, the notation $|0_{ALL}\rangle$ will refer to the zero state on all unmeasured qubits for a given state. It's meaning will be clear from context.)

Definition 6 (C_{Wrap} **).** Define a new unitary C_{Wrap} which consists of all the gates from C which are in the reverse light-cone of $B \cup M \cup F$, but not in $C_{B\cup M\cup F}$ itself. That is, let C_{L-Wrap} (resp. C_{R-Wrap}) be the unitary consisting of all the of the gates in C which are in the reverse light-cone of B (resp. F), but not in $C_{B\cup M\cup F}$ itself, and let $C_{Wrap} \equiv C_{L-Wrap} \circ C_{R-Wrap}$. Therefore,

$$C^\dagger_{Wrap} \circ C = C'_L \circ C_{B\cup M\cup F} \circ C'_R \tag{3}$$

Where $C'_L \equiv C^\dagger_{L-Wrap} \circ C_L$ (see Definition 4 for the definition of C_L) is a unitary acting only or L (the remaining, untouched gates of C within L), and $C'_R \equiv C^\dagger_{R-Wrap} \circ C_R$ (see Definition 4 for the definition of C_R) is a unitary acting only on R (the remaining, untouched gates of C within R). Since C is constant depth it is clear that every qubit in the non-trivial support of C_{Wrap} lies within some constant distance of M. Let R^{Wrap} (resp. L^{Wrap}) be the subset of qubits in R (resp. L) that lie in the non-trivial support of C_{Wrap}. In other words, R^{Wrap} (resp. L^{Wrap}) is the non-trivial support of C_{R-Wrap} (resp. C_{L-Wrap}). See Figure 2 for an illustration of these unitaries with a 1D geometrically-local circuit, and Figure 3 for an illustration in a 2D circuit.

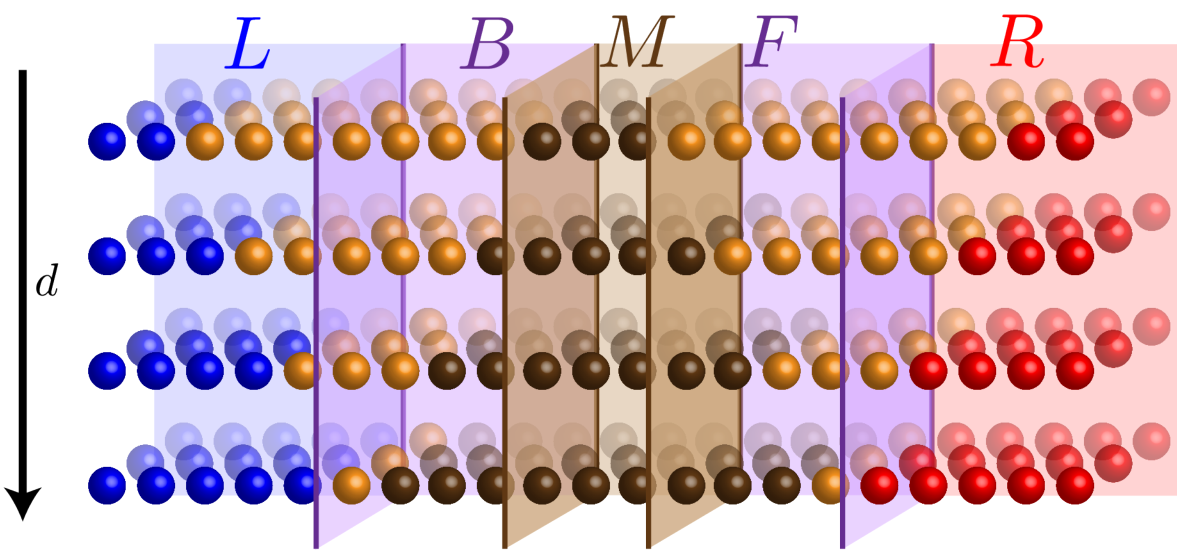

Figure 3: Geometric depiction of the unitaries defined in Definition 4 and Definition 6 for the case of a 2D grid of qubits. Here the vertical dimension represents the depth of the circuit, so the rectangular prism has the same dimensions as the circuit diagram would.

3 Divide and Conquer: Schmidt Vectors and Block Encodings

In this section we will show how to construct a constant depth, geometrically local circuit for the largest Schmidt vector of the unnormalized state $\langle 0_M| C |0_{ALL}\rangle$ across the cut M, in the case that the largest Schmidt coefficient is very large. Let us begin, however, by outlining the intuition behind our divide and conquer approach, which explains why we are interested in constructing Schmidt vectors via circuits in the first place. Consider expanding the quantity $\langle 0_{ALL}| C |0_{ALL}\rangle = \langle 0|_{\text{ALL}} C_{L\cup R} |0\rangle_{L\cup R} \otimes |\psi\rangle_{B\cup F}$ as a sum over the Schmidt decomposition of $|\psi\rangle_{B\cup F}$ across the cut M. Suppose, that $|\psi\rangle_{B\cup F}$ has almost all of its weight on the top polynomially many Schmidt vectors (In Section 4 we will show that, in fact, we can restrict this part of the analysis WLOG to cases where $|\psi\rangle_{B\cup F}$ has a large fraction of its weight on λ_1). Then $|\psi\rangle_{B\cup F} \approx \sum_{i=1}^{p(n)} \lambda_i |v_i\rangle_B \otimes |w_i\rangle_F$, and we have:

$$\langle 0|_{\text{ALL}} C_{L\cup R} |0\rangle_{L\cup R} \otimes |\psi\rangle_{B\cup F} \approx \sum_{i=1}^{p(n)} \lambda_i \langle 0|_{\text{ALL}} C_{L\cup R} |0\rangle_{L\cup R} \otimes |v_i\rangle_B \otimes |w_i\rangle_F \tag{4}$$

$$= \sum_{i=1}^{p(n)} \lambda_i \langle 0|_{L\cup B} C_L |0\rangle_L \otimes |v_i\rangle_B \cdot \langle 0|_{F\cup R} C_R |0\rangle_R \otimes |w_i\rangle_F, \tag{5}$$

where ALL $\equiv L \cup B \cup F \cup R$.

Suppose we could produce approximations for the Schmidt vectors $|v_i\rangle_B$ and $|w_i\rangle_F$ via constant depth, 2D geometrically local circuits. In this case, the quantity in Equation 5 would be a sum of polynomially many scalar quantities, each of which is the product of output probabilities of two new 3D geometrically local circuit problems (C_L and C_R). Furthermore, these new 3D circuit problems involve about half the number of qubits as the original problem we were trying to solve. This leads to a divide and conquer recursion which can yield a more efficient runtime for the original problem. The base case in this divide-and-conquer algorithm, when the width of the recursively divided circuits is reduced down to a constant, can be handled by Theorem 5 of [BGM20]. This Theorem shows that the output probabilities of 2D constant-depth circuits (or, in this case, of 3D constant-depth circuits with constant width in one dimension) can be efficiently computed.

However, this strategy only works if we can produce explicit approximations for the Schmidt vectors $|v_i\rangle_B$ and $|w_i\rangle_F$ via constant depth, 2D geometrically local circuits. In the case when λ_1 is sufficiently large, it turns out that we can at least produce $|v_1\rangle_B$ and $|w_1\rangle_F$ in this way. (Note, we will also need to compute λ_1 efficiently, and this can also be done using Theorem 5 of [BGM20], as described in Definition 25.) Surprisingly, we will see in Section 4 that this is already sufficient to produce a divide-and-conquer algorithm for the whole estimation problem. The key is to combine the above reasoning with an additional expansion trick, expressed in Lemma 18.

The approach for explicitly constructing $|w_1\rangle_F$ is based on a tool called a "block-encoding", which aims to generate a unitary whose top left corner contains the Hermitian matrix $\rho_F \equiv \text{tr}_B(|\psi\rangle\langle\psi|_{B\cup F})$, or the integer powers ρ_F^K for $K = O(\mathsf{polylog}(n))$. In fact, under an assumption that λ_1 is sufficiently large, $\frac{1}{\lambda_1^K}\rho_F^K$ is already very close to a projector onto $|w_1\rangle_F$ (see Lemma 16 for the explicit scaling).

Lemma 7 (Lemma 45 of [GSLW19]). *The following is a 2D-local constant-depth circuit which gives a block encoding for $\rho_F \equiv \text{tr}_B(|\psi\rangle\langle\psi|_{B\cup F})$:*

$$(C^\dagger_{B\cup M\cup F} \otimes I_{F'})(I_{B\cup M} \otimes SWAP_{FF'})(C_{B\cup M\cup F} \otimes I_{F'})$$

Note: In Lemma 7 and throughout the paper we make use of a slight abuse of notation. When we say that a circuit is 2D-local we are including, within that definition, circuits which are 3D local but have a constant width (a constant number of qubits) in one of the dimensions. This is the case for the circuits in Lemmas 7, and 8, for example. The reason that this is a reasonable use of terminology in the context of this paper is that the base-case algorithm that we will use to compute properties of 2D-local circuits (Theorem 5 of [BGM20]) can also handle these "constant-width 3D local" circuits with the same time complexity.

Proof. From Lemma 45 of [GSLW19] it follows that the circuit $(C^\dagger_{B\cup M\cup F} \otimes I_{F'})(I_{B\cup M} \otimes \text{SWAP}_{FF'})(C_{B\cup M\cup F} \otimes I_{F'})$ is a block-encoding of ρ_F. Here F' is a fresh register which is identical in size to F. Note that SWAP is not geometrically local a priori, but if we interleave the qubits of F and F' in the geometrically appropriate way, which we are free to do, then the $\text{SWAP}_{FF'}$ can be implemented in a geometrically local, depth-1 manner. Thus the entire block-encoding is still given by a constant-depth 2D local circuit.

One additional subtlety: We are neglecting to measure the M register in the $|0\rangle$ basis here, but this is still a block-encoding for ρ_F nonetheless. The reason is that that measurement can be absorbed into the definition of block-encoding. □

Following Lemma 53 of [GSLW19], we can now create a block encoding for the K^{th} power of ρ_F by creating K distinct F registers $F_1, ..., F_K$ (interwoven in the geometrically appropriate way just as in the proof of Lemma 7), and multiplying K different block encodings for ρ_F, each using a different one of the registers F_i, as so:

$$\prod_{i=1}^{K}(C^\dagger_{B\cup M\cup F} \otimes I_{F_1,...,F_K})(I_{B\cup M} \otimes \text{SWAP}_{FF_i})(C_{B\cup M\cup F} \otimes I_{F_1,...,F_K}) \tag{6}$$

We therefore have the following Lemma.

Lemma 8 (Lemma 53 of [GSLW19])**.** *For any constant integer $K > 0$, the following is a 2D-local circuit which gives a block encoding for ρ_F^K, and has depth $O(K)$:*

$$\prod_{i=1}^{K}(C^\dagger_{B\cup M\cup F} \otimes I_{F_1,...,F_K})(I_{B\cup M} \otimes SWAP_{FF_i})(C_{B\cup M\cup F} \otimes I_{F_1,...,F_K}) \tag{7}$$

Proof. The fact that Equation (7) gives a block encoding of ρ_F^K follows by repeated application of Lemma 53 of [GSLW19]. The circuit in Equation (7) has depth $O(K)$ because it is a composition of $3K$ constant-depth circuits. The circuit can be made 2D-local because we choose, WLOG, for the F_j registers to be interleaved with the other qubits in a manner that matches the 2D geometry. Since there are now K different F_j registers, this can increase the depth of our circuit by another factor of K (adding SWAP gates to ensure that every gate is exactly 2D-local). So the depth of the geometrically-local version of the circuit is $O(K^2)$. □

Stated concretely, the fact that the circuit in Equation (7) is a block encoding for ρ_F^K simply means that, if we define $|0_{ancilla}\rangle = |0_{F_1,...F_k,M,B}\rangle$, then:

$$\rho_F^K = \langle 0_{ancilla}| \prod_{i=1}^{K}(C^\dagger_{B\cup M\cup F} \otimes I_{F_1,...,F_K})(I_{B\cup M} \otimes \text{SWAP}_{FF_i})(C_{B\cup M\cup F} \otimes I_{F_1,...,F_K}) |0_{ancilla}\rangle \,. \tag{8}$$

4 Divide and Conquer: Splitting Over Heavy Slices

In this section we will prove a set of results which will allow us to precisely define and analyze the division step in our divide-and-conquer algorithm. The process begins by identifying slices of the circuit C which are appropriate division points. Those are the slices which have "heavy weight" as defined below.

Consider a set of constant-width 2D slices $K = \{K_i\}$ of the qubits of C, where each slice K_i is parallel to the cut $B \cup M \cup F$ shown in Figure 1, and is made up of three analogous sections B_i, M_i, F_i. Let the the slices in K be evenly spaced at a constant distance apart, where the constant is chosen to be large enough that the light cones of K_i and K_j are disjoint when $i \neq j$. For concreteness we will say that the distance between slices K_i is equal to $10d$, where we recall that d is the depth of the circuit C. We will also set the width of each of the sections B_i, M_i, F_i to be $10d$, just as discussed in Definition 3. This ensures that the properties stipulated by 3 are satisfied by B_i, M_i, F_i.

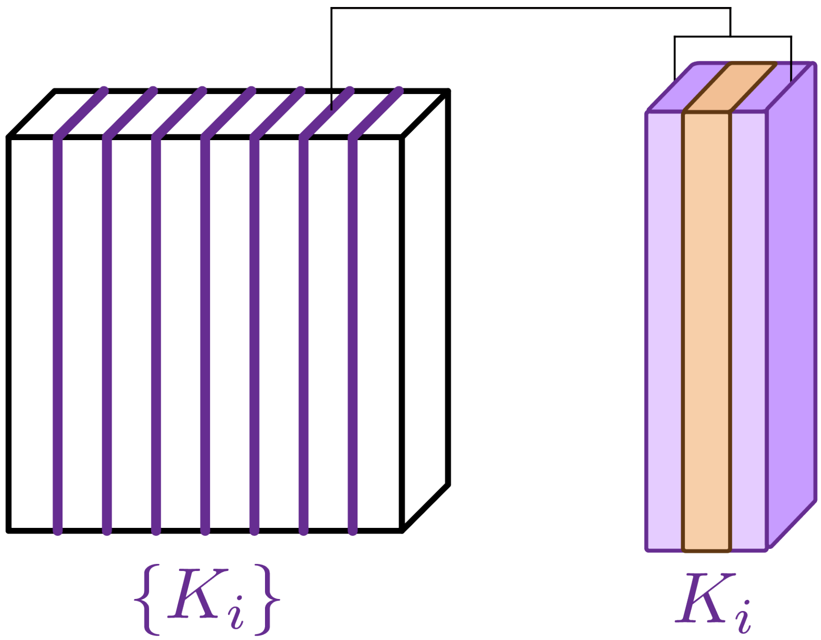

Figure 4: Set of slices $\{K_i\}$

Definition 9. Let $I[M_i = 0]$ be the indicator random variable for the event that all of the qubits in K_i collapse to 0 when measured in the computational basis. Here joint probabilities are defined according the probability distribution p_{total} produced by measuring $C\,|0^{\otimes n}\rangle$ in the computational basis. Let $p_{M_i=0} := \mathbb{E}_{p_{total}}[I[M_i = 0]]$ be the probability that all of the bits in K_i evaluate to 0 according to the distribution p_{total}.

Lemma 10. *The $I[M_i = 0]$ are independent random variables. Therefore,*

$$p_{total}(M_i = 0\ \forall i) = \prod_i p_{total}(M_i = 0).$$

Proof. The $I[M_i = 0]$ are independent random variables because the cuts K_i are light-cone separated by definition. The desired results follows. $\square$

Note that the variables $I[M_i = 0]$ may well be *conditionally* dependent when conditioned on the outcomes of measuring the qubits in between the K_i slices. Indeed, that's what makes the global problem non-trivial in the first place. But, when measuring the K_i slices alone we see that the $I[M_i = 0]$ are independent as stated in Lemma 10.

Lemma 11. *If* $|\langle 0^{\otimes n}| C |0^{\otimes n}\rangle| > |1/q(n)|$, *then, for any* $0 \leq h \leq 1$, $h|K|$ *of the slices* K_i *in* K *have the property that:*

$$p_{total}(K_i = 0) \geq (|1/q(n)|)^{\frac{1}{(1-h)|K|}} \tag{9}$$

We will let K_{heavy} *be the subset of* K *consisting of those* K_i *satisfying Equation (72).*

Proof. The proof of Lemma 11 is given in Appendix A. □

Definition 12. We define any particular slice K_i as $K_i = B_i \cup M_i \cup F_i.$, where B_i, M_i, F_i are the analogous regions to B, M, F (respectively) in Figure 1. These slices are depicted in Figure 4. Let $|\psi\rangle_{B_i \cup F_i}$ be analogous to $|\psi\rangle_{B\cup F} \equiv \langle 0|_M C_{B\cup M \cup F} |0\rangle_{B\cup M\cup F}$. Let L_i and R_i be sets of qubits analogous to the sets L and R. We define the unitaries $C_{B_i \cup M_i \cup F_i}$, C_{wrap_i}, $C'_{L_i} \equiv C^{\dagger}_{L-Wrap_i} \circ C_{L_i}$, and $C'_{R_i} \equiv C^{\dagger}_{R-Wrap_i} \circ C_{R_i}$ exactly as given in Definition 6 for the case of a single cut.

Lemma 13. *For any slice* $K_i \in K_{heavy}$ *satisfying:*

$$p_{total}(M_i = 0) \geq 1 - e(n), \tag{10}$$

the top Schmidt coefficient of $|\psi\rangle_{B_i \cup F_i}$ *satisfies* $\lambda_1^i \geq 1 - O(e(n))$. *(Where the Schmidt decomposition is taken across the partition* B_i, F_i*.)*

Proof. The proof of Lemma 13 is given in Appendix A. □

The proof of Lemma 13 (See Appendix A) suggests a way to perform a division step, dividing the original computational problem into the product of two new problems, but at the cost of an additive error that scales like $\Theta(e(n))$. But, note that, if we want, say, $1/n^d$ additive error for $d \geq 2$, then this additive error term is way too large (in some cases $e(n)$ scales like $1/\log(n)$). This means that, a priori, we cannot even afford to make use of Lemma 13 one single time! However, Lemma 20 below shows how we can use this type of division step to divide the circuit at Δ different, light-cone separated cuts, K_i, simultaneously, and thereby achieve additive error that scales like $e(n)^{\Delta}$.

Definition 14. For any K_i define the following two operators inspired by the block-encoding approach in Section 3:

$$P^K_{F_i} \equiv \frac{1}{\lambda_1^K} \left\langle 0^{B_i, M_i, F_i^1, \ldots F_i^k} \right| \prod_{j=1}^{K} (C^{\dagger}_{B_i \cup M_i \cup F_i} \otimes I_{F_i^1, \ldots, F_i^K})(I_{B_i \cup M_i} \otimes \mathrm{SWAP}_{F_i F_i^j})(C_{B_i \cup M_i \cup F_i} \otimes I_{F_i^1, \ldots, F_i^K}) \left| 0^{B_i, M_i, F_i^1, \ldots F_i^k} \right\rangle$$

and (11)

$$P^K_{B_i} \equiv \frac{1}{\lambda_1^K} \left\langle 0^{F_i, M_i, B_i^1, \ldots B_i^K} \right| \prod_{j=1}^{K} (C^{\dagger}_{B_i \cup M_i \cup F_i} \otimes I_{B_i^1, \ldots, B_i^K})(I_{F_i \cup M_i} \otimes \mathrm{SWAP}_{B_i B_i^j})(C_{B_i \cup M_i \cup F_i} \otimes I_{B_i^1, \ldots, B_i^K}) \left| 0^{F_i, M_i, B_i^1, \ldots B_i^K} \right\rangle$$

Here the first equation gives a linear operator on F_i, and the second equation gives a linear operator on B_i. The registers F_i^j (resp. B_i^j) are dummy registers that are used to create K block encodings of the density matrix of the F_i (resp. B_i) register of the state $C_{B_i \cup M_i \cup F_i} |0^{F_i, M_i, B_i}\rangle$. These K block encodings are then composed (multiplied) with each other in such a manner that they produce the block encoding of the K^{th} power of the density matrix, as described in Section 3.

Definition 15. Define $\Pi^K_{F_i} \equiv C^{\dagger}_{Wrap_i} P^K_{F_i} C_{Wrap_i}$.

Note that the operator $\Pi^K_{F_i}$ is in tensor product with $\left|0_{M_i}\right\rangle$ (it acts as the identity on the M_i register since neither C_{Wrap_i} or $P^K_{F_i}$ act non-trivially on that register).

Lemma 16. *For any $K_i \in K_{heavy}$,*

$$\|P^K_{F_i} - |w_1\rangle\langle w_1|_{F_i}\|_1 \leq f(n) \tag{12}$$

and

$$\|P^K_{B_i} - |v_1\rangle\langle v_1|_{B_i}\|_1 \leq f(n) \tag{13}$$

where $f(n) \equiv \left(\frac{1-\lambda^i_1}{\lambda^i_1}\right)^K$, and $|w_1\rangle\langle w_1|_{F_i}$, $|v_1\rangle\langle v_1|_{B_i}$ are the projectors onto the top Schmidt vectors of $|\psi\rangle_{B_i\cup F_i}$ in F_i and B_i respectively.

Proof. The proof of Lemma 16 is given in Appendix A. □

Definition 17. Let $\sigma \in \mathcal{P}[\Delta] \setminus \varnothing$ where $[\Delta] = \{1,\ldots,\Delta\}$. Define the unormalized states

$$|\Psi_\sigma\rangle = \otimes_{j\in\sigma}\Pi^K_{F_j} \otimes_{i\in[\Delta]} \left\langle 0_{M_i}\right| C\,|0_{ALL}\rangle$$

And,

$$|\Psi_\varnothing\rangle = \otimes_{i\in[\Delta]} \left\langle 0_{M_i}\right| C\,|0_{ALL}\rangle$$

Lemma 18. *Consider a set K_{heavy} of slices such that, for every $K_i \in K_{heavy}$, $|\psi\rangle_{B_i\cup F_i}$ satisfies $\lambda^i_1 \geq 1 - e(n)$, and such that for any $K_i, K_j \in K_{heavy}$, the operators $\Pi^K_{F_i}$ and $\Pi^K_{F_j}$ are light-cone separated whenever $i \neq j$. Then, for any set of Δ slices, $\{K_i\}_{i\in[\Delta]} \subseteq K_{heavy}$, we have that:*

$$\left\|\sum_{\sigma\in\mathcal{P}([\Delta])}(-1)^{|\sigma|}|\Psi_\sigma\rangle\langle\Psi_\sigma|\right\| = \left\||\Psi_\varnothing\rangle\langle\Psi_\varnothing| - \sum_{\sigma\in\mathcal{P}([\Delta])\setminus\varnothing}(-1)^{|\sigma|+1}|\Psi_\sigma\rangle\langle\Psi_\sigma|\right\| = (2e(n)+2f(n))^\Delta, \tag{14}$$

where $f(n) \equiv \left(\frac{1-\lambda^i_1}{\lambda^i_1}\right)^K$

Proof. The proof of Lemma 18 is given in Appendix A. □

We now define three new states that are dependent on a particular choice of K_i.

Definition 19. Given K_i, define the states:

$$|\Omega_i\rangle = \Pi^K_{F_i}\left\langle 0_{M_i}\right| C\,|0_{ALL}\rangle \tag{15}$$

$$\left|\Xi_{L_i}\right\rangle = P^K_{F_i}\left\langle 0_{M_i}\right| C_{L_i}C_{B_i\cup M_i\cup F_i}\left|0_{L_i\cup B_i\cup M_i\cup F_i}\right\rangle \tag{16}$$

$$\left|\Xi_{R_i}\right\rangle = P^K_{B_i}\left\langle 0_{M_i}\right| C_{R_i}C_{B_i\cup M_i\cup F_i}\left|0_{R_i\cup B_i\cup M_i\cup F_i}\right\rangle \tag{17}$$

At this point it is pertinent to state Lemma 20:

Lemma 20. *For any $K_i \in K_{heavy}$ such that $|\psi\rangle_{B_i \cup F_i}$ satisfies $\lambda_1^i \geq 1 - e(n)$, the state $|\Omega_i\rangle\langle\Omega_i|$ is within $(2e(n))^K$ of an unnormalized product state about M_i, described as follows:*

$$\left\| |\Omega_i\rangle\langle\Omega_i| - 1/\lambda_1^i \operatorname{tr}_{F_i}\left(|\Xi_{L_i}\rangle\langle\Xi_{L_i}|\right) \otimes \operatorname{tr}_{B_i}\left(|\Xi_{R_i}\rangle\langle\Xi_{R_i}|\right) \right\| = O((2e(n))^K) \tag{18}$$

Proof. The proof of Lemma 20 is given in Appendix A. □

Definition 21 (Synthesis)**.** We say that an unnormalized quantum state ϕ is *synthesized* by a quantum circuit Γ, if Γ has three registers of qubits L, M, N such that:

$$\phi = \phi_{(\Gamma,L,M,N)} = \operatorname{tr}_{L\cup M}(\langle 0_M| \Gamma |0_{L\cup M\cup N}\rangle \langle 0_{L\cup M\cup N}| \Gamma^\dagger |0_M\rangle). \tag{19}$$

In this case we say that the circuit Γ together with a specification of the registers L, M, N constitutes a *synthesis* of ϕ. When ϕ is implicit we will call this collection (Γ, L, M, N) a *synthesis*.

When Γ is a 3D geometrically-local, constant-depth circuit, and the register N is one contiguous cubic subset of the qubits that Γ acts on, with L, and M only containing qubits on the "edges", we call (Γ, L, M, N) a 3D geometrically-local, constant-depth *synthesis*.

Definition 22. [The Circuits $\Gamma_{i,j}$, $\Gamma_{L,i}$, $\Gamma_{R,j}$] Recall, from Definition 4, that $\Gamma_{B_k\cup M_k\cup F_k}$ ($k \in \{i,j\}$) is defined to be the circuit containing the minimal number of gates of Γ such that every gate acting on M_k is included. Taking this definition for both $k = i$ and $k = j$, we now define $\Gamma_{i,j}$ to be a sub-circuit of Γ consisting of the minimal number of gates of Γ such that $\Gamma_{i,j} \circ \Gamma_{B_i\cup M_i\cup F_i} \circ \Gamma_{B_j\cup M_j\cup F_j}$ contains all of the gates of Γ that lie between M_i and M_j. Similarly define $\Gamma_{L,i}$ (resp. $\Gamma_{R,j}$) to be te sub-circuit of Γ consisting of the minimal number of gates of Γ such that $\Gamma_{L,i} \circ \Gamma_{B_i\cup M_i\cup F_i}$ (resp. $\Gamma_{R,j} \circ \Gamma_{B_j\cup M_j\cup F_j}$) contains all of the gates of Γ that lie between M_i (resp. M_j)and the left-hand side (resp. right-hand side) of the Cube.

Definition 23. Let $S = (\Gamma, G, H, N)$ be a 3D local, constant-depth synthesis, and let K_i, K_j ($i < j$) be two slices on the register N, as described in Definition 12. Let M_i, F_i, B_i, and M_j, F_j, B_j be the subregisters of slices K_i and K_j respectively, as defined in Definition 12. Recall, from Definition 21, that the state synthesized by S is:

$$\phi_S = \operatorname{tr}_{G\cup H}(\langle 0_H| \Gamma |0_{G\cup H\cup N}\rangle \langle 0_{G\cup H\cup N}| \Gamma^\dagger |0_H\rangle).$$

We define three new pure states as follows:

$$\begin{aligned}
|\varphi_{\mathsf{L},i}\rangle &= (\lambda_1^i)^K P_{F_i}^K \langle 0_{M_i,H}| \Gamma_{L_i} \Gamma_{B_i\cup M_i\cup F_i} |0_{L_i\cup B_i\cup M_i\cup F_i\cup G\cup H}\rangle \\
|\varphi_{j,\mathsf{R}}\rangle &= (\lambda_1^j)^K P_{B_j}^K \left\langle 0_{M_j,H}\right| \Gamma_{R_j} \Gamma_{B_j\cup M_j\cup F_j} \left|0_{R_j\cup B_j\cup M_j\cup F_j\cup G\cup H}\right\rangle \\
|\varphi_{i,j}\rangle &= (\lambda_1^i \lambda_1^j)^K P_{B_i}^K \circ P_{F_j}^K \left\langle 0_{M_i,M_j,H}\right| \Gamma_{i,j} \circ \Gamma_{B_i\cup M_i\cup F_i} \circ \Gamma_{B_j\cup M_j\cup F_j} \left|0_{N_{i,j}\cup B_i\cup M_i\cup F_i\cup B_j\cup M_j\cup F_j\cup G\cup H}\right\rangle
\end{aligned} \tag{20}$$

Here $P_{F_i}^K, P_{B_j}^K$ are defined as in Definition 14. In the above the notation $\Gamma_{L_i}, \Gamma_{R_i}, \Gamma_{B_i\cup M_i\cup F_i}$ are defined just like the notation $C_{L_i}, C_{R_i}, C_{B_i\cup M_i\cup F_i}$ in Definition 4, except with the circuit Γ, instead of the circuit C. Finally, $\Gamma_{i,j}$ is defined as in Definition 22, and $N_{i,j}$ is defined to be the sub-register of N containing all of the qubits between F_i and B_j.

From these, we define three new synthesized states (with corresponding syntheses) as follows:

$$
\begin{aligned}
\phi_{\mathsf{L},i} &= \mathrm{tr}_{F_i \cup M_i \cup G \cup H}\left(\left|\varphi_{\mathsf{L},i}\right\rangle \left\langle \varphi_{\mathsf{L},i}\right|\right) \\
\phi_{j,\mathsf{R}} &= \mathrm{tr}_{B_j \cup M_j \cup G \cup H}\left(\left|\varphi_{j,\mathsf{R}}\right\rangle \left\langle \varphi_{j,\mathsf{R}}\right|\right) \\
\phi_{i,j} &= \mathrm{tr}_{B_i \cup M_i \cup M_j \cup F_j \cup G \cup H}\left(\left|\varphi_{i,j}\right\rangle \left\langle \varphi_{i,j}\right|\right)
\end{aligned}
\tag{21}
$$

We can now write out the explicit synthesis for each of these synthesized states as follows: Recalling, from Definition 14 that,

$$
P^K_{F_i} \equiv \frac{1}{(\lambda^i_1)^K} \left\langle 0^{B_i,M_i,F_i^1,\dots F_i^K} \right| \prod_{j=1}^{K} (\Gamma^\dagger_{B_i \cup M_i \cup F_i} \otimes I_{F_i^1,\dots,F_i^K})(I_{B_i \cup M_i} \otimes \mathrm{SWAP}_{F_i F_i^j})(\Gamma_{B_i \cup M_i \cup F_i} \otimes I_{F_i^1,\dots,F_i^K}) \left| 0^{B_i,M_i,F_i^1,\dots F_i^K} \right\rangle
\tag{22}
$$

We have that the explicit synthesis corresponding to $\phi_{\mathsf{L},i}$ is:

$$
\begin{aligned}
S_{\mathsf{L},i} \equiv \quad & \Big(\Gamma_{P^K_{F_i}} \circ \Gamma_{L_i} \circ \Gamma_{B_i \cup M_i \cup F_i}, (F_i \cup G), (M_i \cup M_i' \cup B_i' \cup F_i^1, \dots \cup F_i^K \cup H) \\
& , (L_i \cup B_i \cup M_i \cup F_i \cup M_i' \cup B_i' \cup G \cup H \cup F_i^1, \dots \cup F_i^K) \Big),
\end{aligned}
$$

where $\Gamma_{P^K_{F_i}}$ is defined as

$$
\Gamma_{P^K_{F_i}} \equiv \prod_{j=1}^{K} (\Gamma^\dagger_{B_i' \cup M_i' \cup F_i} \otimes I_{F_i^1,\dots,F_i^K})(I_{B_i' \cup M_i'} \otimes \mathrm{SWAP}_{F_i F_i^j})(\Gamma_{B_i' \cup M_i' \cup F_i} \otimes I_{F_i^1,\dots,F_i^K}),
\tag{23}
$$

where $\Gamma_{B_i' \cup M_i' \cup F_i}$ is the same as $\Gamma_{B_i \cup M_i \cup F_i}$ except that it does not act on registers B_i or M_i at all, but instead, acts on dummy registers B_i' and M_i' in their place.

Symmetrically, for the explicit synthesis for $\left|\phi_{j,\mathsf{R}}\right\rangle$, recall that:

$$
P^K_{B_i} \equiv \frac{1}{(\lambda^i_1)^K} \left\langle 0^{F_i,M_i,B_i^1,\dots B_i^K} \right| \prod_{j=1}^{K} (C^\dagger_{B_i \cup M_i \cup F_i} \otimes I_{B_i^1,\dots,B_i^K})(I_{F_i \cup M_i} \otimes \mathrm{SWAP}_{B_i B_i^j})(C_{B_i \cup M_i \cup F_i} \otimes I_{B_i^1,\dots,B_i^K}) \left| 0^{F_i,M_i,B_i^1,\dots B_i^K} \right\rangle,
\tag{24}
$$

and, therefore, we have that the explicit synthesis corresponding to $\left|\phi_{j,\mathsf{R}}\right\rangle$ is:

$$
\begin{aligned}
S_{j,\mathsf{R}} \equiv \quad & \Big(\Gamma_{P^K_{B_j}} \circ \Gamma_{R_j} \circ \Gamma_{B_j \cup M_j \cup F_j}, (B_j \cup G), (M_j \cup M_j' \cup F_j' \cup B_j^1, \dots \cup B_j^K \cup H) \\
& , (R_j \cup B_j \cup M_j \cup F_j \cup M_j' \cup F_j' \cup G \cup H \cup B_j^1, \dots \cup B_j^K) \Big),
\end{aligned}
$$

where $\Gamma_{P^K_{B_j}}$ is defined as

$$
\Gamma_{P^K_{B_j}} \equiv \prod_{l=1}^{K} (\Gamma^\dagger_{B_j' \cup M_j' \cup F_j} \otimes I_{F_j^1,\dots,F_j^K})(I_{B_j' \cup M_j'} \otimes \mathrm{SWAP}_{F_j F_j^l})(\Gamma_{B_j' \cup M_j' \cup F_j} \otimes I_{F_j^1,\dots,F_j^K}),
\tag{25}
$$

where $\Gamma_{B_j \cup M_j' \cup F_j'}$ is the same as $\Gamma_{B_j \cup M_j \cup F_i}$ except that it does not act on registers F_j or M_j at all, but instead, acts on dummy registers F_j' and M_j' in their place.

Finally, reusing Equations (22),(23) (24), and (25) , the explicit synthesis for $\left|\phi_{i,j}\right\rangle$ can be written as (See Equation 20 for definition of $\left|\phi_{i,j}\right\rangle$):

$$\begin{aligned} S_{i,j} \equiv \quad & \Big(\Gamma_{P^K_{B_i}} \circ \Gamma_{P^K_{F_j}} \circ \Gamma_{i,j} \circ \Gamma_{B_i \cup M_i \cup F_i} \circ \Gamma_{B_j \cup M_j \cup F_j}, (B_i \cup F_j \cup G), \\ & (M_i \cup M_i' \cup F_i' \cup B_i^1, \dots \cup B_i^K \cup M_j \cup M_j' \cup B_j' \cup F_j^1, \dots \cup F_j^K \cup H) \\ & , (N_{i,j} \cup F_i \cup B_j \cup B_i \cup F_j \cup G \cup M_i \cup M_i' \cup F_i' \cup B_i^1, \dots \cup B_i^K \cup M_j \cup M_j' \cup B_j' \cup F_j^1, \dots \cup F_j^K \cup H)\Big), \end{aligned}$$

where $N_{i,j}$ is defined in the same manner as before: the register containing all of the qubits between K_i and K_j.

Definition 24. Define syntheses

$$\begin{aligned} \Lambda_1^{j,T} \equiv \quad & \Big(\Gamma_{P^T_{B_j}} \circ \Gamma_{B_j \cup M_j \cup F_j}, (B_j \cup F_j), (M_j \cup M_j' \cup F_j' \cup B_j^1, \dots \cup B_j^T) \\ & , (B_j \cup M_j \cup F_j \cup M_j' \cup F_j' \cup B_j^1, \dots \cup B_j^T)\Big), \end{aligned}$$

$$\begin{aligned} Z_j^T \equiv \quad & \Big(\Gamma_{P^T_{B_j}}, (B_j), (M_j' \cup F_j' \cup B_j^1, \dots \cup B_j^T) \\ & , (B_j \cup M_j' \cup F_j' \cup B_j^1, \dots \cup B_j^T)\Big), \end{aligned}$$

Note that these two objects are, in this case, scalars (see Definition 21 to understand why). In fact,

$$Z_j^T = \operatorname{tr}\left(\rho^T_{B_j}\right), \tag{26}$$

and

$$\Lambda_1^{j,T} = \operatorname{tr}\left(\rho^T_{B_j} \left|\psi\right\rangle \left\langle\psi\right|_{B_j \cup M_j \cup F_j} \rho^T_{B_j}\right) \tag{27}$$

where

$$\rho^K_{B_i} \equiv \left\langle 0^{F_i, M_i, B_i^1, \dots B_i^K} \right| \prod_{j=1}^{K} (C^\dagger_{B_i \cup M_i \cup F_i} \otimes I_{B_i^1, \dots, B_i^K})(I_{F_i \cup M_i} \otimes \mathrm{SWAP}_{B_i B_i^j})(C_{B_i \cup M_i \cup F_i} \otimes I_{B_i^1, \dots, B_i^K}) \left| 0^{F_i, M_i, B_i^1, \dots B_i^K} \right\rangle$$

as in Lemma 8.

We write the scalars Z_j^T, $\Lambda_1^{j,T}$ as 2D geometrically local, constant-depth syntheses is to emphasize that they can be computed by any algorithm which computes the probability of zero being output by a 2D geometrically local, constant-depth synthesis. In the following analysis we will use the 2D algorithm in Theorem 5 of [BGM20] to compute these quantities to inverse polynomial additive error.

Now we will give a definition of a scalar quantity κ^i_{T,ϵ_2} which is meant to be an approximation for the quantity λ^i_1 that we can compute using the "base case" algorithm $\mathcal{B}$ described below. We need this because we want to use λ^i_1 to normalize terms in Algorithm 2 below. We will use its approximation, κ^i_{T,ϵ_2}, as a substitute, since it is a quantity that we can compute in quasi-polynomial time. The quality of this approximation is the subject of Lemma 26.

Definition 25.

$$\kappa^i_{T,\epsilon_2} \equiv \frac{\mathcal{B}(\Lambda^{j,T}_1, \epsilon_2)}{\mathcal{B}(Z^{2T}_j, \epsilon_2)} = \frac{\operatorname{tr}\left(\rho^T_{B_j} \left|\psi\right\rangle \left\langle\psi\right|_{B_j \cup M_j \cup F_j} \rho^T_{B_j}\right) \pm \epsilon_2}{\operatorname{tr}\left(\rho^{2T}_{B_j}\right) \pm \epsilon_2}$$

Here the notation $\mathcal{B}(\Lambda^{j,T}_1, \epsilon_2)$ (resp. $\mathcal{B}(Z^{2T}_j, \epsilon_3)$) denotes a use of algorithm $\mathcal{B}$, which we define to be the algorithm from Theorem 5 of [BGM20], to compute the scalar quantity $\Lambda^{j,T}_1$ (resp. Z^{2T}_j) to within additive error ϵ_2 (resp. ϵ_3). We will elaborate further on this computational task (time complexity, etc) in the analysis of Algorithm 2.

Lemma 26. *If $\lambda^i_1 \geq 1 - e(n)$ then $|\kappa^i_{T,\epsilon_2} - \lambda^i_1| \leq O\left(\frac{(e(n))^{2T} + \epsilon_2}{(\lambda^i_1)^{2T+1}}\right)$.*

Proof. Starting with the definition:

$$\begin{aligned}
\kappa^i_{T,\epsilon_2} &\equiv \frac{\mathcal{B}(\Lambda^{j,T}_1, \epsilon_2)}{\mathcal{B}(Z^{2T}_j, \epsilon_2)} = \frac{\operatorname{tr}\left(\rho^T_{B_j} \left|\psi\right\rangle \left\langle\psi\right|_{B_j \cup M_j \cup F_j} \rho^T_{B_j}\right) \pm \epsilon_2}{\operatorname{tr}\left(\rho^{2T}_{B_j}\right) \pm \epsilon_2} \\
&= \frac{(\lambda^i_1)^{2T+1} + O(e(n)^{2T}) + O(\epsilon_2)}{(\lambda^i_1)^{2T} + O(e(n)^{2T}) + O(\epsilon_2)} = \frac{(\lambda^i_1)^{2T+1}(1 + O(\frac{e(n)^{2T}+\epsilon_2}{(\lambda^i_1)^{2T+1}}))}{(\lambda^i_1)^{2T}(1 + O(\frac{e(n)^{2T}+\epsilon_2}{(\lambda^i_1)^{2T}}))} \\
&= \lambda^i_1 \frac{(1 + O(\frac{e(n)^{2T}+\epsilon_2}{(\lambda^i_1)^{2T+1}}))}{(1 + O(\frac{e(n)^{2T}+\epsilon_2}{(\lambda^i_1)^{2T}}))} = \lambda^i_1 \left(1 + O\left(\frac{e(n)^{2T} + \epsilon_2}{(\lambda^i_1)^{2T+1}}\right)\right) \\
&= \lambda^i_1 + O\left(\frac{e(n)^{2T} + \epsilon_2}{(\lambda^i_1)^{2T+1}}\right) \qquad (28)
\end{aligned}$$

The desired result follows. $\square$

Definition 27. For any natural number Δ, we define $[\Delta] \equiv \{1, ...\Delta\}$. We define $\mathcal{P}([\Delta])$ to be the set of all subsets of $[\Delta]$, that is, the power set of $[\Delta]$. For any set $\sigma \in \mathcal{P}([\Delta])$, we let σ_{max} denote the largest element of σ. We let $|\sigma|$ denote the size of the set σ, and for any $0 < i \leq |\sigma|$ we let $\sigma(i)$ denote the i^{th} smallest element of σ.

5 Estimating Amplitudes in Quasi-polynomial Time

In this section we define and analyze our algorithm for computing $|\left\langle 0^{\otimes n}\right| C \left|0^{\otimes n}\right\rangle|^2$.

Algorithm 1: $\mathcal{A}_{full}(C, \mathcal{B}, \delta)$: Driver for Algorithm 2

Input : 3D Geometrically-Local, Constant-Depth circuit C, base-case algorithm $\mathcal{B}$, approximation error δ

Output: An approximation of $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ to within additive error δ.

`/* We begin by handling the case in which` δ `is so small that it trivializes our runtime, and the case in which` δ `is so large that it causes meaningless errors: */`

1 **if** $\delta \leq 1/n^{\log^2(n)}$ **then**

2   **return** The value $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ computed with zero error by a "brute force" $2^{O(n)}$-time algorithm.

3 **if** $\delta \geq 1/2$ **then return** $1/2$

`/* Here begins the non-trivial part of the algorithm: */`

4 Let N be the register containing all of the qubits on which C acts. Since these qubits are arranged in a cubic lattice, one of the sides of the cube N must have length at most $n^{\frac{1}{3}}$. We will call the length of this side the "width" and will now describe how to "cut" the cube N, and the circuit C, perpendicular to this particular side.

5 Select $\frac{1}{10d}n^{\frac{1}{3}}$ light-cone separated slices K_i of constant width in N, with at most $10d$ distance between adjacent slices. Let $h(n) = \log^7(n)$. Use the base case algorithm $\mathcal{B}$ to check if at least $\frac{1}{10d}n^{\frac{1}{3}} - h(n)$ of the slices obey:

6

$$\left|\operatorname{tr}\left(\langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^{\dagger} |0_{M_i}\rangle\right)\right| \geq 2^{\frac{\log(\delta)}{h(n)}}.$$

OR, there are fewer than $\frac{1}{10d}n^{\frac{1}{3}} - h(n)$ slices that obey:

7

$$\left|\operatorname{tr}\left(\langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^{\dagger} |0_{M_i}\rangle\right)\right| \geq 2^{\frac{\log(\delta)}{2h(n)}}.$$

`/* Note that one of the two conditions above must hold. See the runtime analysis in the proof of Theorem 1 for a detailed explanation of how the base case algorithm` $\mathcal{B}$ `can efficiently distinguish between these two cases. */`

8 **if** Fewer than $\frac{1}{10d}n^{\frac{1}{3}} - h(n)$ of the slices obey Line 7 **then return** 0

9 **if** *At least* $\frac{1}{10d}n^{\frac{1}{3}} - h(n)$ *of the slices obey Line 7* **then**

10   We will denote the set of these slices by K_{heavy}. Note that the maximum amount of width between any two adjacent slices in K_{heavy} is $10d \cdot h(n)$. Furthermore, the maximum amount of width collectively between Δ slices in K_{heavy} is $10d\Delta + 10d \cdot h(n)$. Now that the set K_{heavy} has been defined, we will use this fixed set in the recursive algorithm, Algorithm 2.

11   Define the constant-depth, geometrically local synthesis $S \equiv (C, L, M, N)$, where $L = M = \emptyset$, are empty registers, and N is the entire input register for the circuit C.

12   **return** $\mathcal{A}(S, \eta = \frac{\log(n)}{3\log(4/3)}, \Delta = \log(n), \epsilon = \delta 2^{-10\log(n)\log(\log(n)))}, h(n) = \log^7(n), K_{heavy}, \mathcal{B})$

Algorithm 2: $\mathcal{A}(S, \eta, \Delta, \epsilon, h(n), K_{heavy}, \mathcal{B})$: η-tier Divide and Conquer algorithm with base-case algorithm $\mathcal{B}$

Input : 3D Geometrically-Local, Constant-Depth synthesis S, number of iterations η, number of cuts Δ, stopping width w_0, positive base-case error bound $\epsilon > 0$, base-case algorithm $\mathcal{B}$, a set of heavy slices K_{heavy}

Output: An approximation of the quantity $\langle 0_N | \phi_S | 0_N \rangle$ where ϕ_S is the un-normalized mixed state specified by the constant-depth, 3D geometrically-local synthesis S, and $|0_N\rangle$ is the 0 state on the entire N register of that synthesis. The approximation error is bounded in the analysis below.

Given the constant-depth, geometrically local synthesis $S = (\Gamma, L, M, N)$, let us ignore the registers L and M as they have already been measured or traced-out.

Let d be the depth of the part of the circuit C that acts on N.

Let ℓ be the width of the N register of the synthesis S. Define the stopping width $w_0 \equiv 10d(\Delta + h(n) + 2)$.

if $\ell < w_0 = 10d(\Delta + h(n) + 2)$ *OR* $\eta < 1$ **then**

  Use the base-case algorithm $\mathcal{B}$ to compute the quantity $\langle 0_N | \phi_S | 0_N \rangle$ to within error ϵ.

  return $\mathcal{B}(S, \epsilon)$

else

  We will "slice" the 3D geometrically-local, constant-depth synthesis S in Δ different locations, as follows:

  Since N is 3D we define a region $Z \subset N$ to be the sub-cube of N which has width $10d(\Delta + h(n) + 2)$, and is centered at the halfway point of N width-wise (about the point $\ell/2$ of the way across N). Since the maximum amount of width collectively between Δ slices in K_{heavy} is $10d\Delta + 10d \cdot h(n)$ (see Algorithm 1), we are guaranteed that the region Z will contain at least Δ slices, $K_1, K_2, \ldots, K_\Delta$, from K_{heavy}. For any two slices $K_i, K_j \in K_{heavy}$, let the un-normalized states $|\varphi_{\mathsf{L},i}\rangle, |\varphi_{i,j}\rangle, |\varphi_{j,\mathsf{R}}\rangle$, and corresponding sub-syntheses $S_{\mathsf{L},i}, S_{i,j}, S_{j,\mathsf{R}}$ be as defined in Definition 23, with $K = \log^3(n)$. We will use these to describe the result of our division step below.

  For each $K_i \in K_{heavy}$ pre-compute the quantity κ^i_{T,ϵ_2}, with $T = \log^3(n)$, and $\epsilon_2 = \delta 2^{-10\log(n)\log(\log(n)))}$.

  return

$$\sum_{i=1}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{i,R}, \eta - 1) \tag{29}$$

$$- \sum_{i=1}^{\Delta} \sum_{j=i+1}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{B}(S_{i,j}, \epsilon) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \tag{30}$$

$$+ \sum_{i=1}^{\Delta} \sum_{j=i+2}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \tag{31}$$

$$\cdot \left[\sum_{\sigma \in \mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \mathcal{B}\left(\left(\otimes_{k\in\sigma} \Pi^K_{F_k} \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k\in\sigma} |0_{M_k}\rangle \Pi^K_{F_k} \right), \frac{\epsilon}{2^\Delta} \right) \right] \tag{32}$$

  `/* In the above` $\mathcal{B}\left(\left(\otimes_{k\in\sigma} \Pi^K_{F_k} \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k\in\sigma} |0_{M_k}\rangle \Pi^K_{F_k} \right), \frac{\epsilon}{2^\Delta} \right)$ `denotes an` $\frac{\epsilon}{2^\Delta}$ `approximation of the quantity` $\left(\langle 0_{ALL} | \otimes_{k\in\sigma} \Pi^K_{F_k} \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \Pi^K_{F_k} |0_{ALL}\rangle \right)$ `obtained via base case Algorithm` $\mathcal{B}$`. Note that for brevity it is implied that` $\mathcal{A}(S, \eta) = \mathcal{A}(S, \eta, \Delta, \epsilon, h(n), K_{heavy}, \mathcal{B})$`. */`

5.1 Global Run-Time and Error Analysis for Driver Algorithm 1

Theorem (Restatement of Theorem 1)**.** *Let C be any depth-d, 3D geometrically local quantum circuit on n qubits. Algorithm 1, $\mathcal{A}_{full}(C, \mathcal{B}, \delta)$, where $\mathcal{B}$ is the base case algorithm specified in Theorem 5 of [BGM20], will produce the scalar quantity $|\langle 0^{\otimes n}| C |0^{\otimes n}\rangle|^2$ to within δ error in time*

$$T(n) = 2^{\mathsf{polylog}(n)(1/\delta)^{1/\log^2(n)}} \cdot 2^{d^3} \tag{33}$$

Proof. The proof proceeds in two parts, the first bounding the approximation error obtained by the algorithm, and the second bounding the runtime.

Approximation Error: The analysis of the approximation error obtained by $\mathcal{A}_{full}(C, \mathcal{B}, \delta)$ can be broken into four cases according to the IF statements on Lines 1, 3, 8, and 9 of Algorithm 1. The first three cases are easy. If the condition in Line 1 is satisfied, then the specified additive error δ is so small that we can compute the desired quantity, $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$, exactly, by brute force, in $2^{O(n)}$ time, and this will still take less time than the guaranteed runtime:

$$T(n) = 2^{\mathsf{polylog}(n)(1/\delta)^{1/\log^2(n)}} 2^{d^3}.$$

So, if the condition in Line 1 is satisfied, then we are done. If not, we proceed.

Next, if the condition in Line 3 is satisfied, then $\delta \geq 1/2$, in which case, outputting 0 is clearly a δ additive approximation of $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$, since $0 \leq |\langle 0_{ALL}| C |0_{ALL}\rangle|^2 \leq 1$. So, if this is the case, we are done, otherwise we proceed.

Next, if the condition in Line 8 is satisfied, then, either $\delta \geq 1/2$ (in which case, outputting $1/2$ is clearly a δ additive approximation of $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$, since $0 \leq |\langle 0_{ALL}| C |0_{ALL}\rangle|^2 \leq 1$), OR there is a set of slices $K_{lightweight}$ of size at least $h(n)$, such that every slice $K_i \in K_{lightweight}$ satisfies:

$$\left|\mathrm{tr}_{K_i^c}\left(\langle 0_{ALL}| C \left|0_{K_i}\right\rangle\right)\right| < 2^{\frac{\log(\delta)}{2h(n)}}.$$

In this case, since, for all $K_i, K_j \in K_{lightweight}$ with $K_i \neq K_j$ we know that K_i is lightcone separated from K_j. It follows that:

$$|\langle 0_{ALL}| C |0_{ALL}\rangle| \leq \Pi_{K_i \in K}\, \mathrm{tr}_{K_i^c}\left(\langle 0_{ALL}| C \left|0_{K_i}\right\rangle\right) \leq \left(2^{\frac{\log(\delta)}{2h(n)}}\right)^{h(n)} = 2^{\log(\delta)/2} = \sqrt{\delta}. \tag{34}$$

So,

$$|\langle 0_{ALL}| C |0_{ALL}\rangle|^2 \leq \delta^2. \tag{35}$$

Therefore, in this case, Algorithm 1 returns the quantity 0 as an answer , which is trivially a δ-additive error approximation of $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ by Equation 34.

On the other hand, if the IF statement on Line 8 of Algorithm 1 is not satisfied, then that means that the IF statement of Line 9 must be satisfied, by definition. In that case Algorithm 1 returns the quantity:

$$\mathcal{A}(S, \eta = \frac{\log(n)}{3\log(4/3)}, \Delta = \log(n), \epsilon = \delta 2^{-10\log(n)\log(\log(n))}, h(n) = log^7(n), K_{heavy}, \mathcal{B})$$

which we know is an $f(S,\eta,\Delta,\epsilon)$-additive error approximation of $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$, and by Lemma 31 we know that:
$\eta = \frac{\log(n)}{3\log(4/3)}$

$$
\begin{aligned}
f(S,\eta,\Delta,\epsilon) &\le \eta 20^{\eta}\Delta^{3\eta}E_3(n,K,T,\epsilon_2,\epsilon) \\
&= \eta 20^{\eta}\Delta^{3\eta} O\left(2^{\Delta}(2e(n))^K + 2^{\Delta}K\left(e(n)^{2T}+\epsilon_2\right)+\epsilon\right) \\
&= 1/3\log(4/3)\cdot\log(n)20^{\frac{\log(n)}{3\log(4/3)}}(\log(n))^{3\frac{\log(n)}{3\log(4/3)}} O\left(2^{\log(n)}(2(1-2^{\frac{\log(\delta)}{\log^7(n)}}))^{\log^3(n)}\right. \\
&\left. +2^{\log(n)}\log^3(n)\left((1-2^{\frac{\log(\delta)}{\log^7(n)}})^{2\log^3(n)}+\epsilon_2\right)+\delta 2^{-10\log(n)\log(\log(n))}\right) \\
&\le (\log(n))^{2\log(n)}\cdot \mathsf{poly}(n)\cdot\left((2(1-2^{\frac{\log(\delta)}{\log^7(n)}}))^{\log^3(n)}+\epsilon_2+\delta 2^{-10\log(n)\log(\log(n))}\right) \\
&\le (\log(n))^{2\log(n)}\cdot \mathsf{poly}(n)\cdot\left(\left(O\left(\frac{1}{\log^4(n)}\right)\right)^{\log^3(n)}+2\cdot\delta 2^{-10\log(n)\log(\log(n))}\right) \\
&\le 2^{2\log(n)\log(\log(n))}\cdot \mathsf{poly}(n)\cdot\left(O\left(\frac{1}{\log^4(n)}\right)\right)^{\log^3(n)}+\delta 2^{-8\log(n)\log(\log(n))} \\
&\le o(1)\cdot\delta + o(1)\cdot\delta = o(1)\cdot\delta
\end{aligned}
\tag{36}
$$

where the first inequality follows from Lemma 31 and the rest follows by calculation, recalling that $e(n) \le (1-2^{\frac{\log(\delta)}{\log^7(n)}}) = O(1/\log^4(n))$ (since $\delta \ge n^{-\log^2(n)} = 2^{-\log(n)^3}$ as verified in the driver algorithm, Algoritm 1), $K=\log^3(n)$, $T=\log^3(n)$, and $\epsilon_2 = \delta 2^{-10\log(n)\log(\log(n))}$. The final inequality, which claims $2^{2\log(n)\log(\log(n))}\cdot\mathsf{poly}(n)\cdot\left(O\left(\frac{1}{\log^2(n)}\right)\right)^{\log^2(n)} = o(1)\cdot\delta$, again follows because $\delta \ge n^{-\log^2(n)}$ as verified in the driver algorithm, Algorithm 1.

Runtime: The runtime analysis of Algorithm 1, $\mathcal{A}_{full}(C,\mathcal{B},\delta)$, proceeds by considering the same four cases in the IF statements on Lines 1, 3, 8, and 9, just as in the error analysis above. Just as before, the first three cases are easy. If the IF statement Line 1 is satisfied, then the specified additive error δ is so small that we can compute the desired quantity, $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$, exactly, by brute force, in $2^{O(n)}$ time, and this will still take less time than the guaranteed runtime:

$$T(n) = 2^{\mathsf{polylog}(n)(1/\delta)^{1/\log^2(n)}}2^{d^3}.$$

So, if the IF statement on Line 1 is satisfied, then we are done.

If the the IF statement in Line 3 is satisfied then the algorithm outputs 1/2, which is a constant time operation, and we are done.

On the other hand, in the case that these first two IF statements are not satisfied, we must bound the running time of Line 5. Line 5 can clearly be done in polynomial time, which is an additive cost that is significantly less than our ultimate quasi-polynomial running time upper bound, so we can absorb it into the $O(\cdot)$ notation, and continue without explicitly tracking it.

Line 5 calls for the use of the base case algorithm $\mathcal{B}$ (which we have specified to be the algorithm from Theorem 5 of [BGM20]) to estimate, for every slice K_i, the quantity:

$$\left|\mathrm{tr}_{K_i^c}\left(\langle 0_{ALL}|\, C\, |0_{K_i}\rangle\right)\right|.$$

In particular we use B to estimate this quantity to within additive error $\tilde{\epsilon} \equiv 2^{\frac{\log(\delta)}{2h(n)}-1} - 2^{\frac{\log(\delta)}{h(n)}-1}$, and we count only those slices for which the approximation output by B is at least $2^{\frac{\log(\delta)}{h(n)}} + \tilde{\epsilon}$. So, all the slices accepted by this count will necessarily have weight at least $2^{\frac{\log(\delta)}{h(n)}} + \tilde{\epsilon} - \tilde{\epsilon} = 2^{\frac{\log(\delta)}{h(n)}}$. Furthermore, any slice with weight at least $2^{\frac{\log(\delta)}{h(n)}} + 2\tilde{\epsilon} = 2^{\frac{\log(\delta)}{2h(n)}}$ will certainly be counted by this process. Therefore, this procedure is able to determine which of Line 5 and Line 6 is true. (As noted in a comment in the Algorithm, one of these two must be the case.) It remains to bound the running time cost of these uses of algorithm $\mathcal{B}$. The key observation here is that the quantity:

$$\left|\mathrm{tr}_{K_i^c}\left(\langle 0_{ALL}|\, C\, |0_{K_i}\rangle\right)\right|,$$

only depends on the part of the circuit C that lies in the lightcone of slice K_i. Since K_i is a 2D slice with constant thickness in third dimension, so too is the (slightly larger) part of the circuit C that lies in it's lightcone. While Theorem 5 of [BGM20] only explicitly shows that algorithm $\mathcal{B}$ can compute such quantities for circuits that are exactly 2D local (with no "thickness" in the third dimension), it is straightforward to adapt their techniques to handle the cases where the circuit has constant thickness in the third dimension, while only increasing the runtime by a constant factor in the exponent. This is done by simply increasing the bond dimension in the Matrix Product States used in Theorem 5 of [BGM20] by a constant factor. It follows that the base case algorithm $\mathcal{B}$ can compute the quantity:

$$\left|\mathrm{tr}_{K_i^c}\left(\langle 0_{ALL}|\, C\, |0_{K_i}\rangle\right)\right|,$$

to within additive error $\tilde{\epsilon} \equiv 2^{\frac{\log(\delta)}{2h(n)}-1} - 2^{\frac{\log(\delta)}{h(n)}-1}$ in time $O(\mathsf{poly}(n)/\tilde{\epsilon}^2) = O(\mathsf{poly}(n)/(1-\delta^{1/2h(n)})^2) = O(2^{\mathsf{polylog}(n)})$, where the final equality follows by straightforward calculation whenever $\delta \leq 1/2$, which we know is true because we previously established that the IF statement in Line 3 is not true.

Next, if the IF statement in Line 8 is satisfied, then our Algorithm 1 returns the quantity 0 as an answer, which is a constant time operation, and we are done.

On the other hand, if the IF statement on Line 8 of Algorithm 1 is not satisfied, then that means that the IF statement of Line 9 must be satisfied, by definition. In that case Algorithm 1 returns the quantity:

$$\mathcal{A}(S, \eta = 1/3\log(n), \Delta = \log(n), \epsilon = \delta 2^{-10\log(n)\log(\log(n)))}, h(n) = log^3(n), K_{heavy}, \mathcal{B})$$

which we know is an $f(S, \eta, \Delta, \epsilon)$-additive error approximation of $|\langle 0_{ALL}|\, C\, |0_{ALL}\rangle|^2$, and takes $T(n) < O(2^{\mathsf{polylog} n})$ time to compute. This time bound follows directly from the runtime bound on Algorithm 2, which is given in Theorem 32 of Subsection 5.3. Note that, within Algorithm 2 we already have that $\delta \geq n^{-\log^2(n)}$ because that was verified by Algorithm 1 before it ever calls Algorithm 2. This is the reason that that this runtime bound does not have an explicit δ dependence. In fact it's dependence is $\mathsf{poly}(1/\delta)$, but that is dominated by other terms given the bound on δ.

All together, regardless of which IF statements are true, no step of Algorithm 1 exceeds a running time of $2^{\mathsf{polylog}(n)(1/\delta)^{1/\log^2(n)}} \cdot 2^{d^3}$, and so we are done.

□

5.2 Recursive Error Analysis for Subroutine Algorithm 2

In this subsection we will derive, by induction, an error bound on the estimate produced by Algorithm 2, $\mathcal{A}(S, \eta, \Delta, \epsilon, \mathcal{B}, K_{heavy}, r)$. We will only pursue an error analysis of $\mathcal{A}$ under the assumption that the driver algorithm, Algorithm 1, has actually called Algorithm 2, and has thus constructed the set K_{heavy} according to specification. This is because, if Algorithm 1 does not call Algorithm 2, that means that it has already found an easier approximation to the answer, and the output of $\mathcal{A}$ (Algorithm 2) is not relevant. Recall that, given a synthesis S, the goal of Algorithm 2 is to compute the quantity $|\langle 0_N|\phi\rangle\langle\phi|0_N\rangle|^2$ where $|\phi\rangle$ is the state synthesized by synthesis S, and N is the active register, as defined in Definition 21. The Algorithm $\mathcal{A}(S, \eta, \Delta, \epsilon, \mathcal{B}, K_{heavy}, r)$ is a recursive algorithm and, since the variables $\Delta, \epsilon, \mathcal{B}, K_{heavy}$, and r remain unchanged throughout, we will use a simplification $\mathcal{A}(S, \eta, \Delta, \epsilon, \mathcal{B}, K_{heavy}, r) = \mathcal{A}(S, \eta)$ throughout this analysis. The output of Algorithm 2 is the scalar quantity:

$$\begin{aligned}
\mathcal{A}(S,\eta) \equiv & \sum_{i=1}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta-1) \cdot \mathcal{A}(S_{i,R}, \eta-1) \\
& - \sum_{i=1}^{\Delta} \sum_{j=i+1}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta-1) \cdot \mathcal{B}(S_{i,j}, \epsilon) \cdot \mathcal{A}(S_{j,R}, \eta-1) \\
& + \sum_{i=1}^{\Delta} \sum_{j=i+2}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta-1) \cdot \mathcal{A}(S_{j,R}, \eta-1) \\
& \cdot \left[\sum_{\sigma \in \mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \mathcal{B}\left(\left(\otimes_{k\in\sigma} \Pi^K_{F_k} \langle 0_{M_k}| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} |0_{M_k}\rangle \Pi^K_{F_k} \right), \frac{\epsilon}{2^\Delta} \right) \right]
\end{aligned} \tag{37}$$

Here the notation $\mathcal{B}(T, \epsilon)$ denotes the use of the "base case" algorithm to estimate the quantity $|\langle 0_{ALL}|\phi_T\rangle\langle\phi_T|0_{ALL}\rangle|^2$ to within a desired additive error ϵ, for the specified synthesis T. For us the "base case" algorithm will be defined to be the algorithm from Theorem 5 of [BGM20]. This algorithm can be used for middle sections of the circuit since these sections have a 2D geometry with a "thickness" of at most a polylogarithmic number of qubits in the third dimension. The algorithm from Theorem 5 of [BGM20] can compute an ϵ-additive-error approximation of output probabilities of such syntheses in $O(2^{\mathsf{polylog}(n)}\mathsf{poly}(\frac{1}{\epsilon}))$ time. Note this is not stated explicitly in [BGM20], which technically only handles true 2D circuits (in other words, circuits with "thickness" exactly 1 in the third dimension), but their techniques can be extended to the case of polylogarithmic thickness in a straightforward manner (to do so, increase the bond dimension of their Matrix Product States to polylogarithmic size account for the added "thickness" of qubits).

Since we have assumed that Algorithm 1 has called Algorithm 2, we know that every slice K_i in the input set K_{heavy} to $\mathcal{A}(S, \eta, \Delta, w_0, \epsilon, \mathcal{B}, K_{heavy}, r) = \mathcal{A}(S, \eta)$ satisfies:

$$\left| \mathrm{tr}\left(\langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger |0_{M_i}\rangle \right) \right| \geq 2^{\frac{\log(\delta)}{h(n)}} = 2^{\frac{-\log(1/\delta)}{h(n)}}.$$

Equivalently, $p_{total}(M_i = 0) \geq 1 - e(n)$, where we define:

$$e(n) \equiv (1 - 2^{\frac{-\log(1/\delta)}{h(n)}}). \tag{38}$$

It follows from Lemma 13 that, $\forall K_i \in K_{heavy}$, $\lambda_1^i \geq 1 - O(e(n))$.
We know, by Lemma 18 that,

$$\begin{aligned}
&\left\| \sum_{\sigma \in \mathcal{P}([\Delta])} (-1)^{|\sigma|} \left|\Psi_\sigma\right\rangle \left\langle\Psi_\sigma\right| \right\| = \left\| \left|\Psi_\varnothing\right\rangle \left\langle\Psi_\varnothing\right| - \sum_{\sigma \in \mathcal{P}([\Delta]) \setminus \varnothing} (-1)^{|\sigma|+1} \left|\Psi_\sigma\right\rangle \left\langle\Psi_\sigma\right| \right\| = (2e(n) + 2f(n))^\Delta \\
&\leq \left(2e(n) + 2\left(\frac{e(n)}{1 - O(e(n))} \right)^K \right)^\Delta,
\end{aligned} \tag{39}$$

where $f(n) \equiv \left(\frac{1-\lambda_1^i}{\lambda_1^i} \right)^K \leq \left(\frac{e(n)}{1-O(e(n))} \right)^K$, and the states $\left|\Psi_\sigma\right\rangle$ are defined as:

$$\left|\Psi_\sigma\right\rangle = \otimes_{j \in \sigma} \Pi_{F_j}^K \otimes_{i \in [\Delta]} \left\langle 0_{M_i} \right| C \left|0_{ALL}\right\rangle,$$

$$\left|\Psi_\varnothing\right\rangle = \otimes_{i \in [\Delta]} \left\langle 0_{M_i} \right| C \left|0_{ALL}\right\rangle.$$

Note that, $\langle 0_{ALL}|\Psi_\varnothing\rangle \langle\Psi_\varnothing|0_{ALL}\rangle$ is exactly the quantity that we wish for Algorithm 2 to output! So, the error between the returned output of Algorithm 2, (defined on Line 11 of that algorithm), which we will denote by $\mathcal{A}$ for short, and the desired output quantity $\langle 0_{ALL}|\Psi_\varnothing\rangle \langle\Psi_\varnothing|0_{ALL}\rangle$ is:

$$f(S, \eta, \Delta, \epsilon) \leq \left\| \langle 0_{ALL}|\Psi_\varnothing\rangle \langle\Psi_\varnothing|0_{ALL}\rangle - \mathcal{A} \right\| \tag{40}$$

$$\leq \left\| \langle 0_{ALL}|\Psi_\varnothing\rangle \langle\Psi_\varnothing|0_{ALL}\rangle - \sum_{\sigma \in \mathcal{P}([\Delta]) \setminus \varnothing} (-1)^{|\sigma|+1} \langle 0_{ALL}|\Psi_\sigma\rangle \langle\Psi_\sigma|0_{ALL}\rangle \right\| \tag{41}$$

$$+ \left\| \sum_{\sigma \in \mathcal{P}([\Delta]) \setminus \varnothing} (-1)^{|\sigma|+1} \langle 0_{ALL}|\Psi_\sigma\rangle \langle\Psi_\sigma|0_{ALL}\rangle - \mathcal{A} \right\| \tag{42}$$

$$\begin{aligned}
&\leq \left(2e(n) + 2\left(\frac{e(n)}{1 - O(e(n))} \right)^K \right)^\Delta + \left\| \sum_{\sigma \in \mathcal{P}([\Delta]) \setminus \varnothing} (-1)^{|\sigma|+1} \langle 0_{ALL}|\Psi_\sigma\rangle \langle\Psi_\sigma|0_{ALL}\rangle - \mathcal{A} \right\| \\
&= \left(2e(n) + 2\left(\frac{e(n)}{1 - O(e(n))} \right)^K \right)^\Delta + \left\| \sum_{\sigma \in \mathcal{P}([\Delta]) \setminus \varnothing} (-1)^{|\sigma|+1} \langle 0_{ALL}|\Psi_\sigma\rangle \langle\Psi_\sigma|0_{ALL}\rangle \right. \\
&- \left(\sum_{i=1}^{\Delta} \frac{1}{(\kappa_{T,\epsilon_2}^i)^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{i,R}, \eta - 1) \right. \\
&- \sum_{i=1}^{\Delta} \sum_{j=i+1}^{\Delta} \frac{1}{(\kappa_{T,\epsilon_2}^i \kappa_{T,\epsilon_2}^j)^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{B}(S_{i,j}, \epsilon) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \\
&+ \sum_{i=1}^{\Delta} \sum_{j=i+2}^{\Delta} \frac{1}{(\kappa_{T,\epsilon_2}^i \kappa_{T,\epsilon_2}^j)^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \\
&\left.\left. \cdot \left[\sum_{\sigma \in \mathcal{P}(\{j-1, \ldots, i+1\})} (-1)^{|\sigma|+1} \mathcal{B}\left(\left(\otimes_{k \in \sigma} \Pi_{F_k}^K \left\langle 0_{M_k} \right| \right) \phi_{i,j} \left(\otimes_{k \in \sigma} \left|0_{M_k}\right\rangle \Pi_{F_k}^K \right), \frac{\epsilon}{2^\Delta} \right) \right] \right) \right\|
\end{aligned} \tag{43}$$

Grouping analogous terms and using triangle inequality gives:

$$
\begin{aligned}
f(S,\eta,\Delta,\epsilon) &\leq \left(2e(n)+2\left(\frac{e(n)}{1-O(e(n))}\right)^{K}\right)^{\Delta} \\
&+ \Bigg\| \sum_{i=1}^{\Delta}\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{i,R},\eta-1)-\left\langle 0_{ALL}\middle|\Psi_{\{i\}}\right\rangle\left\langle\Psi_{\{i\}}\middle|0_{ALL}\right\rangle\right) \\
&- \sum_{i=1}^{\Delta}\sum_{j=i+1}^{\Delta}\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2}\kappa^{j}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{B}(S_{i,j},\epsilon)\cdot\mathcal{A}(S_{j,R},\eta-1)-\left\langle 0_{ALL}\middle|\Psi_{\{i,j\}}\right\rangle\left\langle\Psi_{\{i,j\}}\middle|0_{ALL}\right\rangle\right) \\
&+ \sum_{i=1}^{\Delta}\sum_{j=i+2}^{\Delta}\sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})}\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2}\kappa^{j}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1)\right. \\
&\cdot(-1)^{|\sigma|+1}\mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^{K}_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^{K}_{F_k}\right),\frac{\epsilon}{2^{\Delta}}\right) \\
&\left.-\left\langle 0_{ALL}\middle|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\middle|0_{ALL}\right\rangle\right)\Bigg\| \\
&\leq \left(2e(n)+2\left(\frac{e(n)}{1-O(e(n))}\right)^{K}\right)^{\Delta} \\
&+ \sum_{i=1}^{\Delta}\left\|\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{i,R},\eta-1)-\left\langle 0_{ALL}\middle|\Psi_{\{i\}}\right\rangle\left\langle\Psi_{\{i\}}\middle|0_{ALL}\right\rangle\right)\right\| \\
&+ \sum_{i=1}^{\Delta}\sum_{j=i+1}^{\Delta}\left\|\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2}\kappa^{j}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{B}(S_{i,j},\epsilon)\cdot\mathcal{A}(S_{j,R},\eta-1)-\left\langle 0_{ALL}\middle|\Psi_{\{i,j\}}\right\rangle\left\langle\Psi_{\{i,j\}}\middle|0_{ALL}\right\rangle\right)\right\| \\
&+ \sum_{i=1}^{\Delta}\sum_{j=i+2}^{\Delta}\Bigg\|\sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})}(-1)^{|\sigma|+1}\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2}\kappa^{j}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1)\right. \\
&\cdot\mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^{K}_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^{K}_{F_k}\right),\frac{\epsilon}{2^{\Delta}}\right) \\
&\left.-\left\langle 0_{ALL}\middle|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\middle|0_{ALL}\right\rangle\right)\Bigg\|
\end{aligned}
\tag{44}
$$

We bound the three absolute values in Equation 44 as follows:

Lemma 28.

$$
\begin{aligned}
&\left\|\left(\frac{1}{(\kappa^{i}_{T,\epsilon_2})^{4K+1}}\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{i,R},\eta-1)-\left\langle 0_{ALL}\middle|\Psi_{\{i\}}\right\rangle\left\langle\Psi_{\{i\}}\middle|0_{ALL}\right\rangle\right)\right\| \\
&\leq E_1(n,K,T,\epsilon_2)+2f(S,\eta-1,\Delta,\epsilon),
\end{aligned}
$$

where $E_1(n,K,T,\epsilon_2)\equiv 10K(e(n)^{T}+(2e(n))^{K}+\epsilon_2)$.

Proof. The proof of this Lemma is a simpler special case of the proof of Lemma 30 below. It is simpler in that it follows by using Lemma 26, and Lemma 20, and does not require the use of

Lemma 18 as the proof of Lemma 30 does. For succinctness, instead of writing out this entire proof, we refer the reader to the proof of Lemma 30 in the Appendix, of which the proof of this Lemma is a special case. □

Lemma 29.

$$\left\| \left(\frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{B}(S_{i,j},\epsilon)\cdot\mathcal{A}(S_{j,R},\eta-1) - \left\langle 0_{ALL} \middle| \Psi_{\{i,j\}} \right\rangle \left\langle \Psi_{\{i,j\}} \middle| 0_{ALL} \right\rangle \right) \right\| \\ \leq E_2(n,K,T,\epsilon_2,\epsilon) + 2f(S,\eta-1,\Delta,\epsilon), \tag{45}$$

where $E_2(n,K,T,\epsilon_2,\epsilon) \equiv 10K(e(n)^T + (2e(n))^K + \epsilon_2) + \epsilon$

Proof. The proof of this Lemma is a simpler special case of the proof of Lemma 30 below. It is simpler in that it follows by using Lemma 26, and Lemma 20, and does not require the use of Lemma 18 as the proof of Lemma 30 does. For succinctness, instead of writing out this entire proof, we refer the reader to the proof of Lemma 30 in the Appendix, of which the proof of this Lemma is a special case. □

Lemma 30.

$$\begin{aligned} &\Bigg\| \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} (-1)^{|\sigma|+1} \left(\frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \right. \\ &\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^K_{F_k} \langle 0_{M_k}| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} |0_{M_k}\rangle \Pi^K_{F_k} \right), \frac{\epsilon}{2^\Delta} \right) \\ &\left. - \left\langle 0_{ALL} \middle| \Psi_{\{i,j\}\cup\sigma} \right\rangle \left\langle \Psi_{\{i,j\}\cup\sigma} \middle| 0_{ALL} \right\rangle \right) \Bigg\| \\ &\leq E_3(n,K,T,\epsilon_2,\epsilon) + 16f(S,\eta-1,\Delta,\epsilon), \end{aligned} \tag{46}$$

where

$$E_3(n,K,T,\epsilon_2,\epsilon) \equiv O\left(2^\Delta(2e(n))^K + 2^\Delta K\left(e(n)^{2T} + \epsilon_2 \right) + \epsilon \right)$$

Proof. The proof follows by two uses Lemma 20, Lemma 26, AND (unlike the previous two Lemmas) Lemma 18. See Appendix A for a full proof. □

Using all three of the above Lemmas, we have

$$\begin{aligned} f(S,\eta,\Delta,\epsilon) \leq &\left(2e(n) + 2\left(\frac{e(n)}{1-O(e(n))} \right)^K \right)^\Delta + \Delta\left(E_1(n,K,T,\epsilon_2,\epsilon) + 2f(S,\eta-1,\Delta,\epsilon) \right) \\ &+ \Delta^2\left(E_2(n,K,T,\epsilon_2,\epsilon) + 2f(S,\eta-1,\Delta,\epsilon) \right) + \Delta^2\left(E_3(n,K,T,\epsilon_2,\epsilon) + 16f(S,\eta-1,\Delta,\epsilon) \right) \\ &\leq 4\Delta^2 E_3(n,K,T,\epsilon_2,\epsilon) + 20\Delta^2 f(S,\eta-1,\Delta,\epsilon), \end{aligned} \tag{47}$$

where the final inequality follows because $E_3(n, K, T, \epsilon_2, \epsilon) \geq E_2(n, K, T, \epsilon_2, \epsilon) \geq E_1(n, K, T, \epsilon_2, \epsilon)$, and $E_3(n, K, T, \epsilon_2, \epsilon) \geq \left(2e(n) + 2\left(\frac{e(n)}{1-O(e(n))}\right)^K\right)^\Delta$.

We note that, when $\eta = 0$ we have $f(S, \eta, \Delta, \epsilon) = f(S, 0, \Delta, \epsilon) \leq \epsilon$. This is because, in the definition of Algorithm 2, if $\eta < 1$ the algorithm the calls subroutine $\mathcal{B}(S, \epsilon)$ and computes the final desired quantity with error ϵ. This gives us the base case that we need to bound $f(S, \eta, \Delta, \epsilon)$ via standard recursive analysis:

Lemma 31. *The error function $f(S, \eta, \Delta, \epsilon)$ obeys the following bound:*

$$f(S, \eta, \Delta, \epsilon) \leq \eta 20^\eta \Delta^{2\eta} E_3(n, K, T, \epsilon_2, \epsilon) \tag{48}$$

Proof. The Lemma follows by using standard analysis of the recursion in Equation 47, and with the base case $f(S, 0, \Delta, \epsilon) \leq \epsilon \leq E_3(n, K, T, \epsilon_2, \epsilon)$. □

5.3 Recursive Run-Time Analysis for Subroutine Algorithm 2

In this section we will derive a bound on the run-time for Algorithm 2. Recall that, given a synthesis S, the goal of Algorithm 2 is to compute the quantity $|\langle 0_N| \phi_S |0_N\rangle|^2$ where ϕ_S is the state synthesized by synthesis S, and N is the active register, as defined in Definition 21. In the algorithm, we define a center section, Z, along the third dimension of the cube which we will use to choose Δ cuts from. We denote the width of this region as $|Z|$. To approximate $|\langle 0_N| \phi_S |0_N\rangle|^2$ our Algorithm 2 recursively approximates the quantities $|\langle 0_N| \phi_{L,i} |0_N\rangle|^2$, $|\langle 0_N| \phi_{i,j} |0_N\rangle|^2$, and $|\langle 0_N| \phi_{j,R} |0_N\rangle|^2$ as defined in Definition 23, for each pair of K_i, K_j cuts. In other words, at each recursive level we will need to approximate $C(\Delta, 2) = O(\Delta^2)$ sub-quantities via the same number of recursive calls to Algorithm 2. Notice that we have two types of subproblems. We define edge problems as those that lie partially outside of Z (every $\phi_{L,i}$ and $\phi_{j,R}$) and middle problems as those that lie entirely within Z (every $\phi_{i,j}$). See Figure 5.3 for a depiction of the above definitions.

Recall that ℓ is defined to be the width of the N register for our input synthesis S. Since there are Δ total cuts, we have exactly 2Δ edge problems at each recursive level. And since all of the cuts occur within Z, the largest possible edge problem will have size $\ell_{\text{edge}} < \frac{\ell + |Z|}{2}$.

There are fewer than Δ^2 middle problems. Since all of the cuts occur within Z, the largest possible middle problem will have size $\ell_{\text{middle}} < |Z|$.

We now begin our recursive time cost analysis: We use $T(\ell)$ to denote the run-time bound for our algorithm on a synthesis with an N register of width ℓ. From the above arguments, at each recursive level we will produce 2Δ edge problems and less than Δ^2 middle problems all with widths smaller than ℓ. In particular, we know that

$$T(\ell) < 2\Delta T(\ell_{\text{edge}}) + \Delta^2 T(\ell_{\text{middle}}) + \mu(\ell) \tag{49}$$

$$T(\ell) < 2\Delta T\left(\frac{\ell + |Z|}{2}\right) + \Delta^2 T(|Z|) + \mu(\ell) \tag{50}$$

where $\mu(\ell)$ represents the time complexity of preparing and recombining the subproblems, plus the time complexity of executing the base case algorithm in Line 10 , and Equation 69 of Algorithm 2. The first task, which consists of simply identifying where to divide the circuit to form the subproblems, is a polynomial time computation. The second task requires using the base case algorithm $\mathcal{B}$ as specified and discussed in Definition 25. Note that this can be

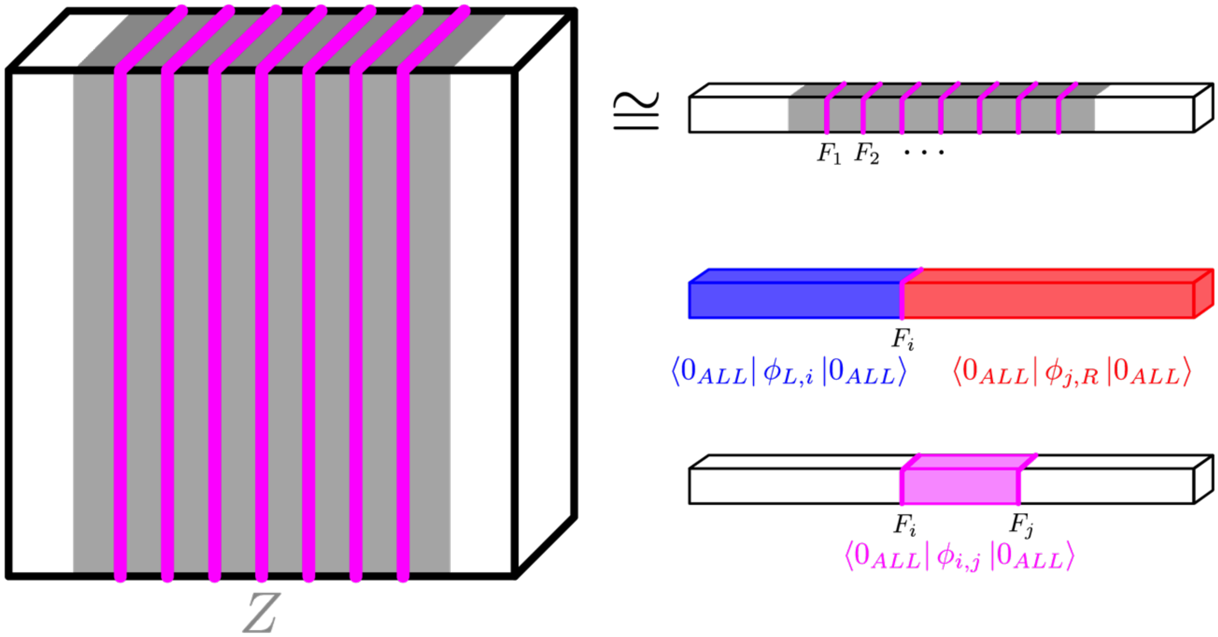

Figure 5: Cuts from Z

done in quasi-polynomial time, by Theorem 5 of [BGM20] because we have chosen, in Algorithm 2, to set $\epsilon_2 = \delta 2^{-10\log(n)\log(\log(n))}$, and because the exponent T in Definition 25 is set to $\log^3(n)$ in Algorithm 2, which means that the 2D circuits on which we are using the base-case algorithm $\mathcal{B}$ have depth at most $\log^6(n)$. The third task requires computing the quantity $\mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\langle 0_{M_k}|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}|0_{M_k}\rangle\Pi^K_{F_k}\right), \frac{\epsilon}{2^\Delta}\right)$, with $\epsilon = \delta 2^{-10\log(n)\log(\log(n))}$, $\Delta = \log(n)$, and $K = \log^3(n)$. The circuit $\phi_{i,j}$ is a state synthesized by a 3D circuit with thickness at most $O(\log^7(n))$ in the third dimension, and with depth at most $K^2 = \log^6(n)$. Furthermore, the circuit $\Pi^K_{F_k}$ do not increase this depth beyond $\log^6(n)$ (because they only act on the part of $\phi_{i,j}$ where the synthesizing circuit still has constant depth, the only place where it has higher depth $\log^6(n)$ is at cuts i, j). With all of this information together, we know that Theorem 5 of [BGM20] (or, actually, a very straightforward modification of that analysis, as discussed earlier) allows us to use base case algorithm $\mathcal{B}$ to compute the quantity $\langle 0_{ALL}|\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\langle 0_{M_k}|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}|0_{M_k}\rangle\Pi^K_{F_k}\right)|0_{ALL}\rangle$ to additive error $\frac{\epsilon}{2^\Delta}$, in $2^{\mathsf{polylog}(n)}$ time. Therefore, $\mu(\ell) \leq 2^{\mathsf{polylog}(n)}$. The very same argument shows us that, in the case that Line 6 of Algorithm 2 is invoked, the runtime of that line is also at most $2^{\mathsf{polylog}(n)}$. So we can write a recursive bound on the runtime $T(\ell)$ as follows:

$$T(\ell) < 2\Delta T\left(\frac{\ell + |Z|}{2}\right) + \Delta^2 2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)} \tag{51}$$

In Algorithm 2 we define $|Z| = 10d(\log n + \log^7 n + 2)$ where d is the depth of the circuit C and n is the total number of qubits in the original problem. Note that $n = \ell^3$. From this point on we will refer to the run-time in terms of $n^{\frac{1}{3}}$ instead of the synthesis width ℓ. For sufficiently large n, the width of edge problems will be bounded above as

$$\frac{n^{\frac{1}{3}} + |Z|}{2} < \frac{3}{4}\cdot n^{\frac{1}{3}} \tag{52}$$

so that we can write our recursive run-time as

$$T(n) < 2\Delta \cdot T\left(\frac{3}{4} \cdot n^{\frac{1}{3}}\right) + \Delta^2 2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)} \tag{53}$$

On line of Algorithm 2, the base case algorithm $\mathcal{B}$ is used on the middle subproblems. This yields

$$T(n) < 2\Delta \cdot T\left(\frac{3}{4} \cdot n^{\frac{1}{3}}\right) + \Delta^2 2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)} \tag{54}$$

where $R(x)$ represents the time-complexity of the base-case algorithm on a synthesis with an N register of size x. Recall that x is exactly the depth of the synthesis in the third dimension. The base-case algorithm that we will use for our algorithm comes from Theorem 5 of [BGM20], and will approximate our base-case 3D circuit to any error ϵ in time

$$R(x) \leq 2^{\mathsf{polylog}(n)} \cdot \epsilon^{-2} \cdot 2^{x+d^3} \tag{55}$$

where n is the total number of qubits in the synthesis, and d is the depth of the circuit given.

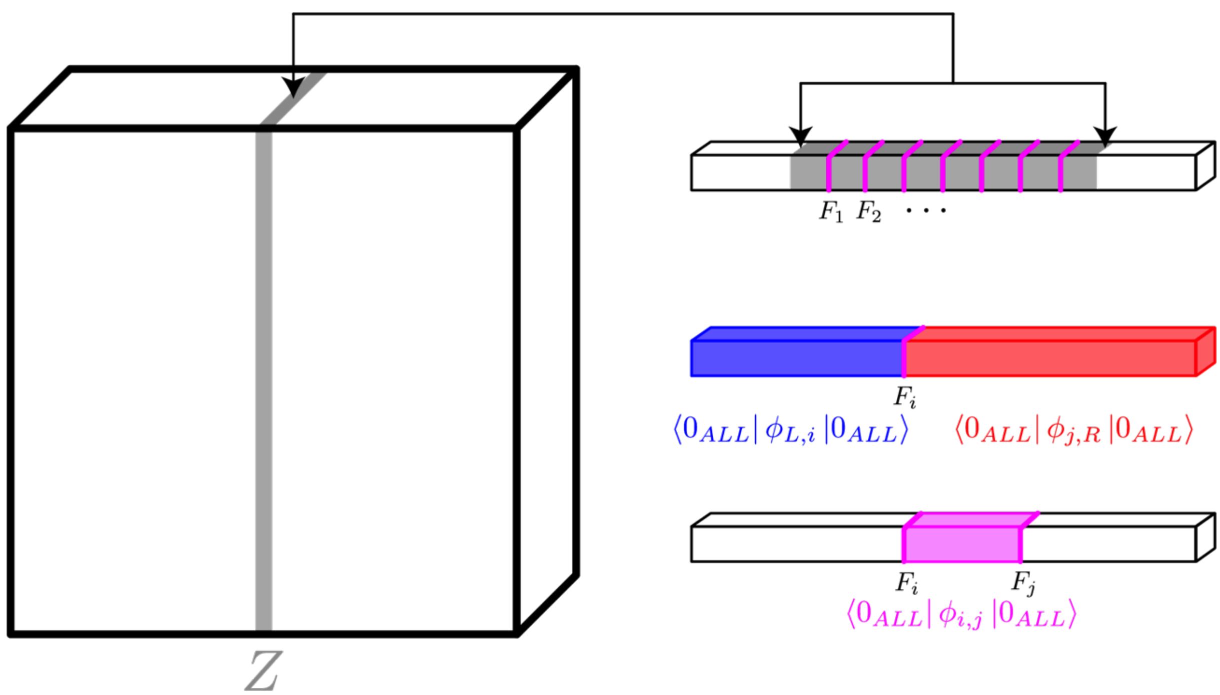

Figure 6: Cuts from a constant-width Z

Note that Equation 54 is a recursive run-time whereby, at each level, we have at most 2Δ subproblems, each with size at most $\frac{3}{4}$ of the original problem. This is a common formula, and we can use the Master Theorem for divide-and-conquer algorithms to determine an upper bound for our run-time as

$$T(n) < (2\Delta)^{\eta} \cdot R\left(\left(\frac{3}{4}\right)^{\eta} n^{\frac{1}{3}}\right) + \sum_{i=0}^{\eta}(2\Delta)^{i} \cdot \left(\Delta^{2}2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)}\right) \tag{56}$$

$$< (2\Delta)^{\eta} \cdot R\left(\left(\frac{3}{4}\right)^{\eta} n^{\frac{1}{3}}\right) + \frac{(2\Delta)^{\eta+1}-1}{(2\Delta)-1} \cdot \left(\Delta^{2}2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)}\right) \tag{57}$$

$$< (2\Delta)^{\eta} \cdot R\left(\left(\frac{3}{4}\right)^{\eta} n^{\frac{1}{3}}\right) + (2\Delta)^{\eta+1} \cdot \left(\Delta^{2}2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)}\right) \tag{58}$$

$$T(n) < (2\Delta)^{\eta}\left[R\left(\left(\frac{3}{4}\right)^{\eta} n^{\frac{1}{3}}\right) + 2\Delta^{3}2^{\mathsf{polylog}(n)} + 2\Delta 2^{\mathsf{polylog}(n)}\right] \tag{59}$$

where η is the depth of our recursive calls. Note that the first $R(x)$ term denotes uses of the base case algorithm on syntheses in the η-th level of recursion.

Theorem 32. *Suppose $\eta = \frac{\log(n)}{3\log(4/3)}$ and $\Delta = \log n$. Given these values, the run-time for Algorithm 2 will be bounded by*

$$T(n) < 2^{\mathsf{polylog}(n)} \tag{60}$$

Proof. Theorem 32 follows directly from the above calculations, but we also include an explicit proof here for completeness:

Using the values $\eta = \frac{\log(n)}{3\log(4/3)}$ and $\Delta = \log n$ in Equation 59 yields:

$$T(n) < (2\log n)^{\frac{-1}{3\log\left(\frac{3}{4}\right)}\log n}\left[R\left(\left(\frac{3}{4}\right)^{\frac{-1}{3\log\left(\frac{3}{4}\right)}\log n} n^{\frac{1}{3}}\right) + 2\log^{3} n \cdot 2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)}\right] \tag{61}$$

$$= \mathrm{poly}(n) \cdot 2^{-\frac{\log^{2} n}{3\log\frac{3}{4}}}\left[R\left(n^{-\frac{1}{3}}n^{\frac{1}{3}}\right) + 2^{\mathsf{polylog}(n)} + 2^{\mathsf{polylog}(n)}\right] \tag{62}$$

$$= \mathrm{poly}(n) \cdot 2^{-\frac{\log^{2} n}{3\log\frac{3}{4}}}\left[R\left(1\right) + 2^{\mathsf{polylog}(n)}\right] \tag{63}$$

$$T(n) < \mathrm{poly}(n) \cdot O\left(2^{\log^{2} n}\right)\left[R\left(1\right) + 2^{\mathsf{polylog}(n)}\right] \leq 2^{\mathsf{polylog}(n)} \tag{64}$$

From Equation 55, $R(1) = \mathrm{poly}(n) \cdot \epsilon^{-2}$, where we have set $\epsilon = \delta 2^{-10\log(n)\log(\log(n))}$.

□

6 A Polynomial Time Algorithm under a Polynomial-Time-Checkable assumption

In Section 5 we show that the quantity $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ can be approximated in quasi-polynomial time by use of a divide-and-conquer strategy about certain "heavy" slices of the circuit, which produce states with large leading Schmidt coefficients. Naturally, one may ask what factors limit the performance of Algorithm 2. An inspection of Line 9 provides a partial answer to this question: in order for Algorithm 2 to guarantee that $\Delta = O(\log(n))$ heavy slices exist within the center region Z, the width of Z must scale like $|Z| = O(\log(n)^3)$. This width means that our applications of the base case algorithm $\mathcal{B}$ will automatically require quasi-polynomial time, forcing the quasi-polynomial run-time of the whole algorithm. The reason that Z must have polylogarithmic width, resulting in a quasi-polynomial run time, is that we make no assumption whatsoever about the positions of the heavy slices within the circuit C. This gives a worst-case performance guarantee for our algorithm. However, it seems natural to consider how we might improve the efficiency of our algorithm in the case that the heavy slices of C can be assumed to be more-or-less evenly spaced. In this Section we show how, under the assumption that heavy slices are evenly spaced, we can design a polynomial time algorithm for this problem.

To begin, let us imagine, as before, that we choose $\Omega(n^{\frac{1}{3}})$ light-cone separated slices $\{K_i\}$ of constant width. For the sake of reducing the run-time, we will need to increase, from Algorithm 2, the required "weight" of each slice K_i. Namely, we will denote the set $K_{heavy} \subset \{K_i\}$ as the $O(n^{\frac{1}{3}})$ slices which obey

$$\left|\mathrm{tr}_{K_i^c}\left(\langle 0_{ALL}| C \left|0_{K_i}\right\rangle\right)\right| \geq 1 - \frac{h\log(n)}{n^{\frac{1}{3}}} \tag{65}$$

where h is some constant. The trade-off of increasing the weight of slices is that we can no longer use the reasoning in Lemma 11 to guarantee that we can find many heavy slices in the small-width center region Z. At least not if the slices are positioned adversarially. But, Lemma 11 does guarantee the existence of sufficiently many heavy slices that, on average,the slices must be spaced only a constant distance apart. If we simply assume that this is not only the case on average, but actually the case for every single pair of adjancent heavy slices, then we are able to conclude that any constant width region Z will have a proportional, constant number of heavy slices within it, and improve the efficiency of our algorithm. The required assumption is defined here:

Assumption 33. Every pair of adjacent slices $K_i, K_{i+1} \in K_{heavy}$ are separated by at most a constant number of qubits, r.

Note that it is possible to *verify* Assumption 33, given the input circuit C, in polynomial time using the base case algorithm $\mathcal{B}$ (which is specified to be the algorithm from Theorem 5 of [BGM20]) and the procedure described in the runtime analysis in the proof of Theorem 1! So, in the case that Assumption 33 is true, we can efficiently verify that it is true.

Given this assumption, we now specificy two new algorithms analogous to Algorithms 1 and 2, but modified in order to leverage Assumption 33 to improve efficiency to polynomial time. After stating the algorithms we will outline how to modify the analysis of Algorithms 1 and 2 to obtain an improved, polynomial, runtime bound (and an unchanged error bound) for these new algorithms. The key difference between our new algorithms below and the original Algorithms 1, and 2 is that, because we have chosen slices that have much higher "weight", and we now know, via Assumption 33, that these slices are spaced at most a constant distance apart, we no longer need our algorithm to cut the circuit at $\Delta = \log(n)$ locations at each division step in order to obtain small error. Instead, it is sufficient for the algorithm to cut at Δ locations where Δ is a constant that just depends on the degree of the desired approximation error. This greatly diminishes the required runtime of the algorithm.

Algorithm 3: $\mathcal{A}_{const}(S, \eta, \Delta, \epsilon, \mathcal{B})$: Driver for Algorithm 4

/* This algorithm computes the quantity $\langle 0_{ALL}| C |0_{ALL}\rangle$ to within δ additive error, where C is a constant-depth, 3D geometrically-local quantum circuit. */

Input : 3D Geometrically-Local, Constant-Depth quantum circuit C, base-case algorithm $\mathcal{B}$, approximation error δ

Output: An approximation of $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ to within additive error δ.

/* We begin by handling the case in which δ is so small that it trivializes our runtime, and the case in which δ is so large that it causes meaningless errors: */

1 **if** $\delta \leq 1/n^{\log^2(n)}$ **then**

2   **return** The value $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ computed with zero error by a "brute force" $2^{O(n)}$-time algorithm.

3 **if** $\delta \geq 1/2$ **then return** $1/2$

/* Here begins the non-trivial part of the algorithm: */

4 Let N be the register containing all of the qubits on which C acts. Since these qubits are arranged in a cubic lattice, one of the sides of the cube N must have length at most $n^{\frac{1}{3}}$. We will call the length of this side the "width" and will now describe how to "cut" the cube N, and the circuit C, perpendicular to this particular side.

5 Select $\frac{1}{20d} n^{\frac{1}{3}}$ light-cone separated slices K_i of constant width in N, with at most $10d$ distance between adjacent slices.

6 Let $h(n) = \frac{1}{50d} n^{\frac{1}{3}}$. Use the base case algorithm $\mathcal{B}$ to check if at least $\frac{1}{40d} n^{\frac{1}{3}}$ of the slices obey

7
$$\left|\mathrm{tr}\left(\langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^{\dagger} |0_{M_i}\rangle\right)\right| \geq \delta^{1/h(n)}.$$

OR, there are fewer than $\frac{1}{40d} n^{\frac{1}{3}}$ slices that obey:

8
$$\left|\mathrm{tr}\left(\langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^{\dagger} |0_{M_i}\rangle\right)\right| \geq \delta^{1/2h(n)}.$$

/* Note that one of the two conditions above must hold. See the runtime analysis in the proof of Theorem 1 for a detailed explanation of how the base case algorithm $\mathcal{B}$ can efficiently distinguish between these two cases. Algorithm 3 handles this in the same way. */

9 **if** Fewer than $\frac{1}{40d} n^{\frac{1}{3}}$ of the slices obey Line 8 **then return** 0

10 **if** *At least* $\frac{1}{40d} n^{\frac{1}{3}}$ *of the slices obey Line 7* **then**

11   We will denote the set of these slices by K_{heavy}. Here, we will assume that the slices obey Assumption 33. Now that the set K_{heavy} has been defined, we will use this set in the recursive Algorithm 4.

12   Define the constant-depth, geometrically local synthesis $S \equiv (C, L, M, N)$, where $L = M = \emptyset$, are empty registers, and N is the entire input register for the circuit C.

13   **return**
$$\mathcal{A}^{rec}_{const}\left(S, \eta = \frac{\log(n)}{3\log(4/3)}, \Delta = \log(\delta)/\log(n), \epsilon = \delta/\mathrm{poly}(n), h(n) = \frac{1}{40d} n^{\frac{1}{3}}, K_{heavy}, \mathcal{B}\right)$$

Algorithm 4: $\mathcal{A}^{rec}_{const}(S, \eta, \Delta, \epsilon, r, K_{heavy}, \mathcal{B})$: η-tier Divide and Conquer algorithm with base-case algorithm $\mathcal{B}$

Input : 3D Geometrically-Local, Constant-Depth synthesis S, number of iterations η, number of cuts Δ, positive base-case error bound $\epsilon > 0$, base-case algorithm $\mathcal{B}$, $\Theta(n^{\frac{1}{3}})$ heavy slices K_{heavy} obeying Assumption 33

Output: An approximation of the quantity $\langle 0_N | \phi_S | 0_N \rangle$.

Given the constant-depth, geometrically local synthesis $S = (\Gamma, L, M, N)$, let us ignore the registers L and M as they have already been measured or traced-out.

Let d be the depth of the part of the C that acts on N.

Let ℓ be the width of the N register of the synthesis S. Let $w_0 \equiv 10\Delta(r + d + 2)$.

if $\ell < w_0 = 10\Delta(r + d + 2)$ *OR* $\eta < 1$ **then**

  Use the base-case algorithm $\mathcal{B}$ to compute the quantity $\langle 0_N | \phi_S | 0_N \rangle$ to within error ϵ.

  return $\mathcal{B}(\phi_S, \epsilon)$

else

  We will "slice" the 3D geometrically-local, constant-depth synthesis S in Δ different locations, as follows:

  Since N is 3D we define a region $Z \subset N$ to be the sub-cube of N which has width $10\Delta(r + d + 2)$, and is centered at the halfway point of N width-wise (about the point $\ell/2$ of the way across N). Since elements of K_{heavy} are separated by at most r qubits (by Assumption 33), we are guaranteed that the region Z will contain at least Δ elements, $K_1, K_2, \ldots, K_\Delta$, of K_{heavy}. For any two slices $K_i, K_j \in K_{heavy}$, let the un-normalized states $\phi_{\mathsf{L},i}, \phi_{i,j}, \phi_{j,\mathsf{R}}$, and corresponding sub-syntheses $S_{\mathsf{L},i}, S_{i,j}, S_{j,\mathsf{R}}$ be as defined in Definition 23. We will use these to precisely define the division step as follows.

  Set the parameters $K = \log(n)$, $T = \log(n)$, $\epsilon_2 = \delta/\mathsf{poly}(n)$.

  return

$$\sum_{i=1}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{i,R}, \eta - 1) \tag{66}$$

$$- \sum_{i=1}^{\Delta} \sum_{j=i+1}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2} \kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{B}(S_{i,j}, \epsilon) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \tag{67}$$

$$+ \sum_{i=1}^{\Delta} \sum_{j=i+2}^{\Delta} \frac{1}{(\kappa^i_{T,\epsilon_2} \kappa^j_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \tag{68}$$

$$\cdot \left[\sum_{\sigma \in \mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \mathcal{B}\left(\left(\otimes_{k\in\sigma} \Pi^K_{F_k} \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k\in\sigma} | 0_{M_k} \rangle \Pi^K_{F_k} \right), \frac{\epsilon}{2^\Delta} \right) \right] \tag{69}$$

  /* Note that for brevity it is implied that $\mathcal{A}(S, \eta) = \mathcal{A}(S, \eta, \Delta, w_0, \epsilon, \mathcal{B}, K_{heavy}, r)$. */

6.1 Global Run-Time and Error Analysis for Algorithms 3 and 4

Theorem (Restatement of Theorem 2)**.** *Let C be any depth-d, 3D geometrically local quantum circuit on n qubits. If C satisfies Assumption 33, then Algorithm 3, $\mathcal{A}_{const}(C, \mathcal{B}, \delta)$ (where $\mathcal{B}$ is the base case algorithm specified to be as in Theorem 5 of [BGM20]), will approximate the scalar quantity $|\langle 0_{ALL}| C |0_{ALL}\rangle|^2$ to within δ additive error in time*

$$T(n) = \mathsf{poly}(n, 2^{(1/\delta)^{1/\log^2(n)}}) \cdot 2^{d^3}. \tag{70}$$

Note that, for any $\delta = \Omega(1/n^{\log(n)})$, we have $(1/\delta)^{1/\log^2(n)} = O(1)$, and this runtime is polynomial. We will not write out a full proof of Theorem 2 as it is very similar to the proof of Theorem 32. Instead, we will simply highlight here the few key differences that lead to the polynomial runtime. Note that the subroutine Algorithm 4 is analogous to the subroutine Algorithm 2. In fact, the only difference between the two is that in Algorithm 4 we are guaranteed that the subregion, Z, that we choose slices from has at most a constant width (and the parameters $K = \log(n)$, $T = \log(n)$, $\epsilon_2 = \delta/\mathsf{poly}(n)$ are set differently). Since the base case algorithm can approximate the output probability of a 2D circuit to an arbitrary error ϵ, our error analysis from Algorithm does not change much in this constant width case. In particular, the following recursive error bound applies for Algorithm 4.

$$f(S, \eta, \Delta, \epsilon) \leq 3\Delta^{3\eta} \left(\Delta \left(\frac{h \log(n)}{n^{\frac{1}{3}}} \right)^{\Delta} + \epsilon \right)$$

Similarly, much of the run-time analysis from Section 5.3 is applicable to Algorithm 4 with a few small modifications. In particular, Equation 59, which gives a run-time bound in terms of the depth, η, and the width of the center region, Z, is modified to the following:

$$T(n) < (2\Delta)^{\eta} \left[R \left(\left(\frac{3}{4} \right)^{\eta} n^{\frac{1}{3}} \right) + 2\Delta^3 R(|Z|) + 2\Delta \mathrm{poly}(n) \right]$$

In Algorithm 4 we have the luxury of assuming (via Assumption 33) that elements of the set K_{heavy} are spaced a constant width apart, h. Further, we assume that the value for $\Delta = \log(\delta)/\log(n)$ is an arbitrarily large constant that does not depend on the number of qubits n (it is the degree of the desired inverse-polynomial precision δ). This allows us to set the width of the center region Z to also be a constant. i.e. $|Z| = 10d(\Delta + r + 2) = O(1)$. Recall that the run-time of the base case algorithm $\mathcal{B}$ on middle subproblems is bounded above by the run-time of a subproblem with width equal to $|Z|$, and depends exponentially on parameters K and T, and polynomially on ϵ. Since $|Z|$ is a constant in Algorithm 4,and $K = \log(n)$, $T = \log(n)$, $\epsilon_2 = \delta/\mathsf{poly}(n)$, this implies that the base case algorithm will be able to compute the output probabilities of middle subproblems in polynomial time. We conclude the following bound on the run-time for Algorithm 4.

Theorem 34. *Suppose $\eta = \frac{-1}{3\log(\frac{3}{4})} \log n$. Given this depth, the run-time for Algorithm 4 will be bounded by*

$$T(n) < \mathsf{poly}(n, 1/\delta) \tag{71}$$

Using Theorem 34 we can then prove Theorem 2 in a manner directly analogous to how Theorem 1 is proven using Theorem 32.

Acknowledgments

MC thanks David Gosset and Sergey Bravyi for helpful discussions. Part of this work was completed while MC was attending the Simons Institute Quantum Wave in Computing workshop.

Appendices

A Proofs of Lemma Statements

A.1 Statements from Section 4

Lemma (Restatement of Lemma 11)**.** *If $|\langle 0^{\otimes n}| C |0^{\otimes n}\rangle| > |1/q(n)|$, then, for any $0 \leq h \leq 1$, $h|K|$ of the slices K_i in K have the property that:*

$$p_{total}(K_i = 0) \geq (|1/q(n)|)^{\frac{1}{(1-h)|K|}} \tag{72}$$

We will let K_{heavy} be the subset of K consisting of those K_i satisfying Equation (72).

Proof. Using 10 we have that:

$$p_{total}(K_i = 0\ \forall i) = \prod_i p_{total}(K_i = 0) \geq |\langle 0^{\otimes n}| C |0^{\otimes n}\rangle| > |1/q(n)| \tag{73}$$

So,

$$\log\left(\prod_i p_{total}(K_i = 0)\right) = \sum_i \log(p_{total}(K_i = 0)) \geq \log(|1/q(n)|) \tag{74}$$

Since every term on both sides of the equation is negative, it follows that at least $h|K|$ of the slices K_i in K must satisfy $\log(p_{total}(K_i = 0)) \geq \frac{1}{(1-h)|K|}\log(|1/q(n)|)$.

So, for at least $h|K|$ of the slices K_i in K must satisfy

$$p_{total}(K_i = 0) \geq \exp\left\{\frac{1}{(1-h)|K|}\log(|1/q(n)|)\right\} = (|1/q(n)|)^{\frac{1}{(1-h)|K|}}$$

□

Lemma (Restatement of Lemma 13)**.** *For any slice $K_i \in K_{heavy}$ satisfying:*

$$p_{total}(K_i = 0) \geq 1 - e(n), \tag{75}$$

the top Schmidt coefficient of $|\psi\rangle_{B_i \cup F_i}$ satisfies $\lambda_1^i \geq 1 - O(e(n))$. (Where the Schmidt decomposition is taken across the partition B_i, F_i.)

Proof. For any $K_i \in K_{heavy}$, recall that, by definition, the constant width of M_i is chosen large enough that B_i and F_i do not have any intersecting light cones (so the two halves of the circuit are lightcone separated). It follows that,

$$\begin{aligned}&\mathrm{tr}_{M_i}(C_{B_i\cup M_i\cup F_i} |0_{B_i\cup M_i\cup F_i}\rangle\langle 0_{B_i\cup M_i\cup F_i}| C^\dagger_{B_i\cup M_i\cup F_i}) = \\ &\mathrm{tr}_{M_i\cup F_i}(C_{B_i\cup M_i\cup F_i} |0_{B_i\cup M_i\cup F_i}\rangle\langle 0_{B_i\cup M_i\cup F_i}| C^\dagger_{B_i\cup M_i\cup F_i}) \otimes \mathrm{tr}_{M_i\cup B_i}(C_{B_i\cup M_i\cup F_i} |0_{B_i\cup M_i\cup F_i}\rangle\langle 0_{B_i\cup M_i\cup F_i}| C^\dagger_{B_i\cup M_i\cup F_i})\end{aligned}$$

But, from Equation 75, which is an assumption of the Lemma, we see that,

$$\mathrm{tr}\left(\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right| C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right|\right) \tag{76}$$

$$= \mathrm{tr}\Big(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i}\Big) = p_{total}(K_i = 0) \geq 1 - e(n), \tag{77}$$

and so,

$$\Big\| C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}$$
$$-\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right| C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right|\Big\|$$
$$\leq 1 - p_{total}(K_i = 0) \leq e(n)$$

So,

$$\begin{aligned}
\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i} &= \mathrm{tr}_{M_i}(\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right| C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right|) \\
&= \mathrm{tr}_{M_i}(C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}) + O(e(n)) \\
&= \mathrm{tr}_{M_i\cup F_i}(C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}) \\
&\otimes \mathrm{tr}_{M_i\cup B_i}(C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}) + O(e(n)) \\
&= \mathrm{tr}_{M_i\cup F_i}(\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right| C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right|) \\
&\otimes \mathrm{tr}_{M_i\cup B_i}(\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right| C_{B_i\cup M_i\cup F_i}\left|0_{B_i\cup M_i\cup F_i}\right\rangle\left\langle 0_{B_i\cup M_i\cup F_i}\right| C^{\dagger}_{B_i\cup M_i\cup F_i}\left|0_{M_i}\right\rangle\left\langle 0_{M_i}\right|) + O(e(n)) \\
&= \mathrm{tr}_{F_i}(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i}) \otimes \mathrm{tr}_{B_i}(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i}) + O(e(n))
\end{aligned} \tag{78}$$

Now, by definition, the largest Schmidt coefficient λ_1^i of $\left|\psi\right\rangle_{B_i\cup F_i}$ is equal to the largest eigenvalue of the mixed state $\mathrm{tr}_{F_i}(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i})$, which is equivalent to the largest eigenvalue of the mixed state $\mathrm{tr}_{B_i}(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i})$ (since $\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i}$ is an unormalized pure state).

For notational brevity we define $\rho_{B_i} \equiv \mathrm{tr}_{F_i}(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i})$ and $\rho_{F_i} \equiv \mathrm{tr}_{B_i}(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i})$. By Holder's Inequality (with Holder parameters set top $p = 1$ and $q = \infty$) we have that:

$$\|\rho_{B_i}^2\|_1 \leq \|\rho_{B_i}\|_1\|\rho_{B_i}\|_\infty = \|\rho_{B_i}\|_\infty = \lambda_1^i,$$
and,
$$\|\rho_{F_i}^2\|_1 \leq \|\rho_{F_i}\|_1\|\rho_{F_i}\|_\infty = \|\rho_{F_i}\|_\infty = \lambda_1^i,$$

where the second to last inequality follows because $\|\rho_{B_i}\|_1 = \mathrm{tr}(\rho_{B_i}) \leq 1$ (resp. $\|\rho_{F_i}\|_1 = \mathrm{tr}(\rho_{F_i}) \leq 1$), and the last equality follows by the definition of λ_1^i. So, we have:

$$\begin{aligned}
(\lambda_1^i)^2 &\geq \|\rho_{B_i}^2\|_1\|\rho_{F_i}^2\|_1 = \mathrm{tr}\left(\rho_{B_i}^2\right)\mathrm{tr}\left(\rho_{F_i}^2\right) = \mathrm{tr}\left(\rho_{B_i}^2\otimes\rho_{F_i}^2\right) = \mathrm{tr}\left(\left(\rho_{B_i}\otimes\rho_{F_i}\right)^2\right) \\
&= \mathrm{tr}\left(\left(\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i}\right)^2\right) + O(e(n)) \geq 1 - O(e(n)),
\end{aligned} \tag{79}$$

Where the first three equalities follow by definition, the fourth equality follows by two uses of Equation 78 (and the fact that $\|\left|\psi\right\rangle\left\langle\psi\right|_{B_i\cup F_i}\|_1 \leq 1$), and the final inequality follows by Equation 77. It follows that:

$$\lambda_1^i \geq 1 - O(e(n)) \tag{80}$$

□

Lemma (Restatement of Lemma 16)**.** *For any $K_i \in K_{heavy}$,*

$$\|P_{F_i}^K - |w_1\rangle\langle w_1|_{F_i}\|_1 \leq f(n) \tag{81}$$

and

$$\|P_{B_i}^K - |v_1\rangle\langle v_1|_{B_i}\|_1 \leq f(n) \tag{82}$$

where $f(n) \equiv \left(\frac{1-\lambda_1^i}{\lambda_1^i}\right)^K$, and $|w_1\rangle\langle w_1|_{F_i}$, $|v_1\rangle\langle v_1|_{B_i}$ are the projectors onto the top Schmidt vectors of $|\psi\rangle_{B_i \cup F_i}$ in F_i and B_i respectively.

Proof. Here we will write the proof for Equation 81, but the proof for Equation 82 is exactly analogous. In particular, by Lemma 8 we have that:

For any constant integer $K > 0$, the following is a 2D-local circuit which gives a block encoding for $\rho_{F_i}^K$:

$$\prod_{j=1}^{K}(C^\dagger_{B_i\cup M_i\cup F_i} \otimes I_{F_i^1,\ldots,F_i^K})(I_{B_i\cup M_i} \otimes \mathrm{SWAP}_{F_iF_i^j})(C_{B_i\cup M_i\cup F_i} \otimes I_{F_i^1,\ldots,F_i^K}).$$

It follows, by the definition of a block encoding, that,

$$\rho_{F_i}^K = \left\langle 0^{B_i,M_i,F_i^1,\ldots F_i^k}\right| \prod_{j=1}^{K}(C^\dagger_{B_i\cup M_i\cup F_i} \otimes I_{F_i^1,\ldots,F_i^K})(I_{B_i\cup M_i} \otimes \mathrm{SWAP}_{F_iF_i^j})(C_{B_i\cup M_i\cup F_i} \otimes I_{F_i^1,\ldots,F_i^K}) \left|0^{B_i,M_i,F_i^1,\ldots F_i^k}\right\rangle$$

Recall the definition of $P_{F_i}^K$:

$$P_{F_i}^K \equiv \frac{1}{(\lambda_1^i)^K} \left\langle 0^{B_i,M_i,F_i^1,\ldots F_i^k}\right| \prod_{j=1}^{K}(C^\dagger_{B_i\cup M_i\cup F_i} \otimes I_{F_i^1,\ldots,F_i^K})(I_{B_i\cup M_i} \otimes \mathrm{SWAP}_{F_iF_i^j})(C_{B_i\cup M_i\cup F_i} \otimes I_{F_i^1,\ldots,F_i^K}) \left|0^{B_i,M_i,F_i^1,\ldots F_i^k}\right\rangle.$$

And thus,

$$P_{F_i}^K = \frac{1}{(\lambda_1^i)^K}\rho_{F_i}^K = \left(\frac{\rho_{F_i}}{\lambda_1^i}\right)^K.$$

By the definition of λ_1^i and the leading Schmidt coefficient we have that:

$$\frac{\rho_{F_i}}{\lambda_1^i} = |w_1\rangle\langle w_1|_{F_i} + E,$$

where $E \equiv (\frac{\rho_{F_i}}{\lambda_1^i} - |w_1\rangle \langle w_1|_{F_i})$ is a PSD operator with trace norm $\|E\| = \frac{1-\lambda_1^i}{\lambda_1^i}$, and which is orthogonal to $\langle w_1|_{F_i}$ (i.e. $|w_1\rangle \langle w_1|_{F_i} \cdot E = 0$). It follows that:

$$\left(\frac{\rho_{F_i}}{\lambda_1^i}\right)^K = \Big(|w_1\rangle \langle w_1|_{F_i} + E\Big)^K = \Big(|w_1\rangle \langle w_1|_{F_i}\Big)^K + E^K. \tag{83}$$

So,

$$\left\|P_{F_i}^K - |w_1\rangle \langle w_1|_{F_i}\right\| = \left\|\left(\frac{\rho_{F_i}}{\lambda_1^i}\right)^K - |w_1\rangle \langle w_1|_{F_i}\right\| = \|E^K\| = \|E\|^K = \left(\frac{1-\lambda_1^i}{\lambda_1^i}\right)^K \tag{84}$$

□

Lemma (Restatement of Lemma 18)**.** *Consider a set K_{heavy} of slices such that, for every $K_i \in K_{heavy}$, $|\psi\rangle_{B_i \cup F_i}$ satisfies $\lambda_1^i \geq 1 - e(n)$, and such that for any $K_i, K_j \in K_{heavy}$, the operators $\Pi_{F_i}^K$ and $\Pi_{F_j}^K$ are light-cone separated whenever $i \neq j$. Then, for any set of Δ slices, $\{K_i\}_{i\in[\Delta]} \subseteq K_{heavy}$, we have that:*

$$\begin{aligned}
&\left\|\sum_{\sigma\in\mathcal{P}([\Delta])} (-1)^{|\sigma|} \otimes_{j\in\sigma} \Pi_{F_j}^K \otimes_{i\in[\Delta]} \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger \otimes_{i\in[\Delta]} |0_{M_i}\rangle \otimes_{j\in\sigma} \Pi_{F_j}^K\right\| \\
&= \Big\| \otimes_{i\in[\Delta]} \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger \otimes_{i\in[\Delta]} |0_{M_i}\rangle \qquad (85) \\
&- \sum_{\sigma\in\mathcal{P}([\Delta])\backslash\{\emptyset\}} (-1)^{|\sigma|+1} \otimes_{j\in\sigma} \Pi_{F_j}^K \otimes_{i\in[\Delta]} \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger \otimes_{i\in[\Delta]} |0_{M_i}\rangle \otimes_{j\in\sigma} \Pi_{F_j}^K \Big\| \\
&= (2e(n) + 2f(n))^\Delta \qquad (86)
\end{aligned}$$

Proof.

$$\left\| \sum_{\sigma \in \mathcal{P}([\Delta])} (-1)^{|\sigma|} \otimes_{j\in\sigma} \Pi^K_{F_j} C \left|0_{ALL}\right\rangle \left\langle 0_{ALL}\right| C^\dagger \otimes_{j\in\sigma} \Pi^K_{F_j} \right\| =$$

$$\left\| \sum_{\sigma \in \mathcal{P}([\Delta])} (-1)^{|\sigma|} \otimes_{j\in\sigma} \Pi^K_{F_j} \left(C_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_{|\sigma|},R} \circ \otimes_{j\in[|\sigma|]} C_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \left|0_{ALL}\right\rangle \left\langle 0_{ALL}\right| C^\dagger \right) \otimes_{j\in\sigma} \Pi^K_{F_j} \right\|$$

$$= \left\| \sum_{\sigma \in \mathcal{P}([\Delta])} (-1)^{|\sigma|} \otimes_{j\in\sigma} \Pi^K_{F_j} \left(C_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_{|\sigma|},R} \circ \otimes_{j\in[|\sigma|]} C_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \right) \left|0_{ALL}\right\rangle \left\langle 0_{ALL}\right| \right.$$
$$\left. \left(C_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_{|\sigma|},R} \circ \otimes_{j\in[|\sigma|]} C_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \right)^\dagger \otimes_{j\in\sigma} \Pi^K_{F_j} \right\|$$

$$= \left\| \sum_{\sigma \in \mathcal{P}([\Delta])} (-1)^{|\sigma|} \otimes_{j\in\sigma} \Pi^K_{F_j} \left(C_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_{|\sigma|},R} \circ \otimes_{j\in[|\sigma|]} C_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \right) \left|0_{ALL}\right\rangle \left\langle 0_{ALL}\right| \right.$$
$$\left. \left(\otimes_{j\in[|\sigma|]} C^\dagger_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \circ C^\dagger_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C^\dagger_{\sigma_j,\sigma_{j+1}} \circ C^\dagger_{\sigma_{|\sigma|},R} \right) \otimes_{j\in\sigma} \Pi^K_{F_j} \right\|$$

$$= \left\| \sum_{\sigma \in \mathcal{P}([\Delta])} (-1)^{|\sigma|} \left(C_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_{|\sigma|},R} \circ \otimes_{j\in[|\sigma|]} P^K_{F_j} C_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \right) \left|0_{ALL}\right\rangle \left\langle 0_{ALL}\right| \right.$$
$$\left. \left(\otimes_{j\in[|\sigma|]} P^K_{F_j} C^\dagger_{B_{\sigma_j}\cup M_{\sigma_j}\cup F_{\sigma_j}} \circ C^\dagger_{L,\sigma_1} \circ \otimes_{j\in[|\sigma|-1]} C^\dagger_{\sigma_j,\sigma_{j+1}} \circ C^\dagger_{\sigma_{|\sigma|},R} \right) \right\|$$

$$= \left\| C_{L,1} \circ \otimes_{j\in[\Delta-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_\Delta,R} \circ \right.$$
$$\otimes_{j\in[\Delta]} \left(C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} - P^K_{F_j} C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} P^K_{F_j} \right)$$
$$\otimes \left|0_{ALL\setminus \cup_{j\in\Delta} B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{ALL\setminus \cup_{j\in\Delta} B_j\cup M_j\cup F_j}\right|$$
$$\left. \circ C^\dagger_{L,1} \circ \otimes_{j\in[\Delta-1]} C^\dagger_{\sigma_j,\sigma_{j+1}} \circ C^\dagger_{\Delta,R} \right\|$$
$$= \left\| \otimes_{j\in[\Delta]} \left(C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} - P^K_{F_j} C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} P^K_{F_j} \right) \right.$$
$$= \prod_{j\in[\Delta]} \left\| \left(C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} - P^K_{F_j} C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} P^K_{F_j} \right) \right\|$$

Here the first three equalities follow from definitions in a straightforward manner. The fourth equality follows by the definition of $\Pi^K_{F_j}$, Definition , and the definition of C_{wrap_i}, Definition . The fifth inequality can be shown by expanding the middle tensor product on the right hand side of the equality, which ultimately results in the sum of terms on the left hand side of the equality. The sixth equality follows because $C_{L,1} \circ \otimes_{j\in[\Delta-1]} C_{\sigma_j,\sigma_{j+1}} \circ C_{\sigma_\Delta,R}$ is a unitary operator by definition, and conjugating by this unitary operator does not change the trace norm of the middle operator in that equation:

$$\otimes_{j\in[\Delta]} \left(C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} - P^K_{F_j} C_{B_j\cup M_j\cup F_j} \left|0_{B_j\cup M_j\cup F_j}\right\rangle \left\langle 0_{B_j\cup M_j\cup F_j}\right| C^\dagger_{B_j\cup M_j\cup F_j} P^K_{F_j} \right).$$

The final equality in the above sequence of equalities follows by a standard property of the trace norm.

Now, since the terms $\otimes_{i\in[\Delta]}\left|0_{M_i}\right\rangle$, $\otimes_{i\in[\Delta]}\left\langle 0_{M_i}\right|$ commute with the terms $\otimes_{j\in\sigma}\Pi^K_{F_j}$, and the terms $P^K_{F_j}$ and all i, j, and σ, we have that:

$$\left\|\sum_{\sigma\in\mathcal{P}([\Delta])}(-1)^{|\sigma|}\otimes_{j\in\sigma}\Pi^K_{F_j}\left\langle 0_{M_j}\right|C\left|0_{ALL}\right\rangle\left\langle 0_{ALL}\right|C^\dagger\otimes_{j\in\sigma}\left|0_{M_j}\right\rangle\Pi^K_{F_j}\right\|$$
$$=\prod_{j\in[\Delta]}\left\|\left(\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\otimes_{i\in[\Delta]}\left|0_{M_j}\right\rangle\right.\right. \tag{87}$$
$$\left.\left.-P^K_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle P^K_{F_j}\right)\right\| \tag{88}$$

On the other hand, for every $i\in[\Delta]$ we have that:

$$\left\|\left(\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\right.\right. \tag{89}$$
$$\left.\left.-P^K_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle P^K_{F_j}\right)\right\| \tag{90}$$
$$\leq\left\|\left(\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\right.\right. \tag{91}$$
$$\left.\left.-\left|w_1\right\rangle\left\langle w_1\right|_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\left|w_1\right\rangle\left\langle w_1\right|_{F_j}\right)\right\| \tag{92}$$
$$+\left\|\left(\left|w_1\right\rangle\left\langle w_1\right|_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\left|w_1\right\rangle\left\langle w_1\right|_{F_j}\right.\right. \tag{93}$$
$$\left.\left.-P^K_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle P^K_{F_j}\right)\right\| \tag{94}$$
$$\leq\left\|\left(\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\right.\right. \tag{95}$$
$$\left.\left.-\left|w_1\right\rangle\left\langle w_1\right|_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\left|w_1\right\rangle\left\langle w_1\right|_{F_j}\right)\right\|+2f(n) \tag{96}$$
$$\leq 2e(n)+2f(n) \tag{97}$$

Where the first inequality follows by triangle inequality and the second inequality follows by Lemma 16, where $f(n)\equiv$, and $\left|w_1\right\rangle\left\langle w_1\right|_{F_j}$, is the projector onto the top Schmidt vector of $\left|\psi\right\rangle_{B_j\cup F_j}$ in F_j. The third inequality follows using the assumption taken in the statement of Lemma 18 that the top schmidt coefficient of $\left|\psi\right\rangle_{B_j\cup F_j}$ satisfies $\lambda^j_1\geq 1-e(n)$ (for every j).

If follows that:

$$\left\|\sum_{\sigma\in\mathcal{P}([\Delta])}(-1)^{|\sigma|}\otimes_{j\in\sigma}\Pi^K_{F_j}\otimes_{i\in[\Delta]}\left\langle 0_{M_i}\right|C\left|0_{ALL}\right\rangle\left\langle 0_{ALL}\right|C^\dagger\otimes_{i\in[\Delta]}\left|0_{M_i}\right\rangle\otimes_{j\in\sigma}\Pi^K_{F_j}\right\|$$
$$=\prod_{j\in[\Delta]}\left\|\left(\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle\right.\right. \tag{98}$$
$$\left.\left.-P^K_{F_j}\left\langle 0_{M_j}\right|C_{B_j\cup M_j\cup F_j}\left|0_{B_j\cup M_j\cup F_j}\right\rangle\left\langle 0_{B_j\cup M_j\cup F_j}\right|C^\dagger_{B_j\cup M_j\cup F_j}\left|0_{M_j}\right\rangle P^K_{F_j}\right)\right\|\leq(2e(n)+2f(n))^\Delta \tag{99}$$

□

Lemma (Restatement of Lemma 20)**.** *For any $K_i \in K_{heavy}$ such that $|\psi\rangle_{B_i \cup F_i}$ satisfies $\lambda_1^i \geq 1 - e(n)$, the state $\Pi_{F_i}^K \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C |0_{M_i}\rangle \Pi_{F_i}^K$ is within $(2e(n))^K$ of an unnormalized product state about M_i, described as follows:*

$$\Big\| \Pi_{F_i}^K \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger |0_{M_i}\rangle \Pi_{F_i}^K \tag{100}$$

$$-1/\lambda_1^i \operatorname{tr}_{F_i}\left(P_{F_i}^K \langle 0_{M_i}| C_{L_i} C_{B_i \cup M_i \cup F_i} |0_{L_i \cup B_i \cup M_i \cup F_i}\rangle \langle 0_{L_i \cup B_i \cup M_i \cup F_i}| C_{B_i \cup M_i \cup F_i}^\dagger C_{L_i}^\dagger |0_{M_i}\rangle P_{F_i}^K\right)$$

$$\otimes \operatorname{tr}_{B_i}\left(P_{B_i}^K \langle 0_{M_i}| C_{R_i} C_{B_i \cup M_i \cup F_i} |0_{R_i \cup B_i \cup M_i \cup F_i}\rangle \langle 0_{R_i \cup B_i \cup M_i \cup F_i}| C_{B_i \cup M_i \cup F_i}^\dagger C_{R_i}^\dagger |0_{M_i}\rangle P_{B_i}^K\right)\Big\| \tag{101}$$

$$= O((2e(n))^K) \tag{102}$$

Proof. Recall that, by Definition 15, $\Pi_{F_i}^K \equiv C_{Wrap_i} P_{F_i}^K C_{Wrap_i}^\dagger$. So,

$$\Pi_{F_i}^K \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger |0_{M_i}\rangle \Pi_{F_i}^K = \langle 0_{M_i}| \Pi_{F_i}^K C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger \Pi_{F_i}^K |0_{M_i}\rangle$$
$$= \langle 0_{M_i}| C_{Wrap_i} P_{F_i}^K C_{Wrap_i}^\dagger C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger C_{Wrap_i} P_{F_i}^K C_{Wrap_i}^\dagger |0_{M_i}\rangle$$
$$= \langle 0_{M_i}| C_{Wrap_i} \circ P_{F_i}^K \circ C'_{L_i} \circ C_{B_i \cup M_i \cup F_i} \circ C'_{R_i} |0_{ALL}\rangle \langle 0_{ALL}| (C')_{L_i}^\dagger \circ C_{B_i \cup M_i \cup F_i}^\dagger \circ (C')_{R_i}^\dagger \circ P_{F_i}^K \circ C_{Wrap_i}^\dagger |0_{M_i}\rangle$$
$$= C_{Wrap_i} P_{F_i}^K \left(\langle 0_{M_i}| \circ C'_{L_i} \circ C_{B_i \cup M_i \cup F_i} \circ C'_{R_i} |0_{ALL}\rangle \langle 0_{ALL}| (C')_{L_i}^\dagger \circ C_{B_i \cup M_i \cup F_i}^\dagger \circ (C')_{R_i}^\dagger |0_{M_i}\rangle\right) P_{F_i}^K C_{Wrap_i}^\dagger$$
$$= C_{Wrap_i} P_{F_i}^K \Big(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')_{L_i}^\dagger \otimes \langle 0_{M_i}| C_{B_i \cup M_i \cup F_i} |0_{B_i \cup M_i \cup F_i}\rangle \langle 0_{B_i \cup M_i \cup F_i}| C_{B_i \cup M_i \cup F_i}^\dagger |0_{M_i}\rangle$$
$$\otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')_{R_i}^\dagger\Big) P_{F_i}^K C_{Wrap_i}^\dagger$$
$$= C_{Wrap_i} P_{F_i}^K \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')_{L_i}^\dagger \otimes |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')_{R_i}^\dagger\right) P_{F_i}^K C_{Wrap_i}^\dagger$$
$$= C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')_{L_i}^\dagger \otimes P_{F_i}^K |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P_{F_i}^K \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')_{R_i}^\dagger\right) C_{Wrap_i}^\dagger \tag{103}$$

Here the first equality holds because the $\Pi_{F_i}^K$ operators only act on register F_i, which is disjoint from register M_i. The second equality holds by the the definition of $\Pi_{F_i}^K$, see Definition 15. The third equality holds by Equation 3, repeated below for the convenience of the reader.

$$C_{Wrap_i}^\dagger \circ C = C'_{L_i} \circ C_{B_i \cup M_i \cup F_i} \circ C'_{R_i}$$

The fourth equality holds because neither the operator C_{Wrap_i}, nor the operator $P_{F_i}^K$ act (non-trivially) on the register M_i. The fifth equality holds because the operators C'_{L_i}, C'_{R_i}, and $C_{B_i \cup M_i \cup F_i}$ all act on disjoint registers and are therefore in tensor product by definition. The sixth equality holds by the definition of $|\psi\rangle_{B_i \cup M_i \cup F_i}$. The seventh equality follows because $P_{F_i}^K$ only acts (non-trivially) on the register F_i, by definition.

Now, define:

$$E \equiv C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')_{L_i}^\dagger \otimes |w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')_{R_i}^\dagger\right) C_{Wrap_i}^\dagger$$
$$- C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')_{L_i}^\dagger \otimes P_{F_i}^K |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P_{F_i}^K \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')_{R_i}^\dagger\right) C_{Wrap_i}^\dagger$$
$$= C_{Wrap_i} \Big(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')_{L_i}^\dagger \otimes \left(|w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} |w_1\rangle \langle w_1|_{F_i} - P_{F_i}^K |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P_{F_i}^K\right)$$
$$\otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')_{R_i}^\dagger\Big) C_{Wrap_i}^\dagger \tag{104}$$

(We will show below, using Lemma 16, that the trace norm of E is small, so it can be regarded as an error term.) We have, from Equation 103, that:

$$
\begin{aligned}
&\Pi^K_{F_i} \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger |0_{M_i}\rangle \Pi^K_{F_i} \\
&= C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes P^K_{F_i} |\psi\rangle \langle\psi|_{B_i\cup M_i\cup F_i} P^K_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{Wrap_i} \\
&= C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes |w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle\psi|_{B_i\cup M_i\cup F_i} |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{Wrap_i} + E \\
&= C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes \left(\lambda^i_1 |w_1\rangle_{B_i} |w_1\rangle_{F_i} \langle w_1|_{B_i} \langle w_1|_{F_i} \right) \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{Wrap_i} + E \\
&= \lambda^i_1 C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \right) \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{Wrap_i} + E \\
&= \lambda^i_1 C_{L-Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes |w_1\rangle \langle w_1|_{B_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \left(|w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} + E \\
&= \lambda^i_1 C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes |w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} + E
\end{aligned} \tag{105}
$$

Now defining G_1, and G_2 as,

$$
\begin{aligned}
G_1 &\equiv \lambda^i_1 C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes |w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} \\
&- C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes P^K_{F_i} |\psi\rangle \langle\psi|_{B_i\cup M_i\cup F_i} P^K_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i}
\end{aligned} \tag{106}
$$

and

$$
\begin{aligned}
G_2 &= C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes P^K_{F_i} |\psi\rangle \langle\psi|_{B_i\cup M_i\cup F_i} P^K_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} \\
&- 1/\lambda^i_1 C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes P^K_{F_i} |\psi\rangle \langle\psi|_{B_i\cup M_i\cup F_i} P^K_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(P^K_{B_i} |\psi\rangle \langle\psi|_{B_i\cup M_i\cup F_i} P^K_{B_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i}
\end{aligned} \tag{107}
$$

(we will later show that G_1 and G_2 are small in the trace norm, so they can be regarded as error terms) it follows from Equation 105 that:

$$
\begin{aligned}
&\Pi_{F_i}^K \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger |0_{M_i}\rangle \Pi_{F_i}^K \\
&= \lambda_1^i C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes |w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} + E \\
&= C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes P^K_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(|w_1\rangle \langle w_1|_{B_i} \otimes |w_1\rangle \langle w_1|_{F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} + E + G_1 \\
&= 1/\lambda_1^i C_{L-Wrap_i} \operatorname{tr}_{F_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes P^K_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{F_i} \right) C^\dagger_{L-Wrap_i} \\
&\otimes C_{R-Wrap_i} \operatorname{tr}_{B_i} \left(P^K_{B_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{B_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} + E + G_1 + G_2 \\
&= 1/\lambda_1^i \operatorname{tr}_{F_i} \left(P^K_{F_i} C_{L-Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} \right) C^\dagger_{L-Wrap_i} P^K_{F_i} \right) \\
&\otimes \operatorname{tr}_{B_i} \left(P^K_{B_i} C_{R-Wrap_i} \left(|\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{R-Wrap_i} P^K_{B_i} \right) + E + G_1 + G_2 \\
&= 1/\lambda_1^i \operatorname{tr}_{F_i} \left(P^K_{F_i} \langle 0_{M_i}| C_{L_i} C_{B_i \cup M_i \cup F_i} |0_{L_i \cup B_i \cup M_i \cup F_i}\rangle \langle 0_{L_i \cup B_i \cup M_i \cup F_i}| C^\dagger_{B_i \cup M_i \cup F_i} C^\dagger_{L_i} |0_{M_i}\rangle P^K_{F_i} \right) \\
&\otimes \operatorname{tr}_{B_i} \left(P^K_{B_i} \langle 0_{M_i}| C_{R_i} C_{B_i \cup M_i \cup F_i} |0_{R_i \cup B_i \cup M_i \cup F_i}\rangle \langle 0_{R_i \cup B_i \cup M_i \cup F_i}| C^\dagger_{B_i \cup M_i \cup F_i} C^\dagger_{R_i} |0_{M_i}\rangle P^K_{B_i} \right) + E + G_1 + G_2
\end{aligned}
$$

It follows, by triangle inequality, that:

$$
\begin{aligned}
&\Big\| \Pi_{F_i}^K \langle 0_{M_i}| C |0_{ALL}\rangle \langle 0_{ALL}| C^\dagger |0_{M_i}\rangle \Pi_{F_i}^K && (108) \\
&-1/\lambda_1^i \operatorname{tr}_{F_i} \left(P^K_{F_i} \langle 0_{M_i}| C_{L_i} C_{B_i \cup M_i \cup F_i} |0_{L_i \cup B_i \cup M_i \cup F_i}\rangle \langle 0_{L_i \cup B_i \cup M_i \cup F_i}| C^\dagger_{B_i \cup M_i \cup F_i} C^\dagger_{L_i} |0_{M_i}\rangle P^K_{F_i} \right) \\
&\otimes \operatorname{tr}_{B_i} \left(P^K_{B_i} \langle 0_{M_i}| C_{R_i} C_{B_i \cup M_i \cup F_i} |0_{R_i \cup B_i \cup M_i \cup F_i}\rangle \langle 0_{R_i \cup B_i \cup M_i \cup F_i}| C^\dagger_{B_i \cup M_i \cup F_i} C^\dagger_{R_i} |0_{M_i}\rangle P^K_{B_i} \right) \Big\| && (109) \\
&\leq \|E\| + \|G_1\| + \|G_2\| && (110)
\end{aligned}
$$

It remains to bound the norms (in this case the trace norm) of E, G_1, G_2. We will start with E: From the definition of E (Equation 104) we see that:

$$
\begin{aligned}
\|E\| &= \Big\| C_{Wrap_i} \left(C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i} \otimes \left(|w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} |w_1\rangle \langle w_1|_{F_i} - P^K_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{F_i} \right) \right. \\
&\quad \left. \otimes C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i} \right) C^\dagger_{Wrap_i} \Big\| \\
&= \|C'_{L_i} |0_{L_i}\rangle \langle 0_{L_i}| (C')^\dagger_{L_i}\| \cdot \| |w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} |w_1\rangle \langle w_1|_{F_i} - P^K_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{F_i}\| \\
&\quad \cdot \|C'_{R_i} |0_{R_i}\rangle \langle 0_{R_i}| (C')^\dagger_{R_i}\| \\
&= \| |0_{L_i}\rangle \langle 0_{L_i}| \| \cdot \| |w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} |w_1\rangle \langle w_1|_{F_i} - P^K_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{F_i}\| \cdot \| |0_{R_i}\rangle \langle 0_{R_i}| \| \\
&= \| |w_1\rangle \langle w_1|_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} |w_1\rangle \langle w_1|_{F_i} - P^K_{F_i} |\psi\rangle \langle \psi|_{B_i \cup M_i \cup F_i} P^K_{F_i}\| \leq 2f(n) = 2 \left(\frac{1-\lambda_1^i}{\lambda_1^i} \right)^K
\end{aligned}
$$

Here the first equality follows by definition of E (Equation 104), the second equality follows because C_{Wrap_i} is unitary and by using the tensor product structure after C_{Wrap_i} is removed, the third equality follows because C'_{L_i} and C'_{R_i} are unitary, the fourth equality follows because $\| |0_{L_i}\rangle\langle 0_{L_i}| \| = \| |0_{R_i}\rangle\langle 0_{R_i}| \| = 1$, and the inequality follows by two sequential applications of Lemma 16.

Next we will bound $\|G_1\|$. From the definition of G_1 in Equation 106 we have that:

$$
\begin{aligned}
&\|G_1\| \equiv \Big\|\lambda_1^i C_{L-Wrap_i}\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\right)C^\dagger_{L-Wrap_i}\\
&\otimes C_{R-Wrap_i}\operatorname{tr}_{B_i}\left(|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\otimes C'_{R_i}|0_{R_i}\rangle\langle 0_{R_i}|(C')^\dagger_{R_i}\right)C^\dagger_{R-Wrap_i}\\
&-C_{L-Wrap_i}\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes P^K_{F_i}|\psi\rangle\langle\psi|_{B_i\cup M_i\cup F_i}P^K_{F_i}\right)C^\dagger_{L-Wrap_i}\\
&\otimes C_{R-Wrap_i}\operatorname{tr}_{B_i}\left(|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\otimes C'_{R_i}|0_{R_i}\rangle\langle 0_{R_i}|(C')^\dagger_{R_i}\right)C^\dagger_{R-Wrap_i}\Big\|\\
&=\Big\|\Big(\lambda_1^i C_{L-Wrap_i}\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\right)C^\dagger_{L-Wrap_i}\\
&-C_{L-Wrap_i}\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes P^K_{F_i}|\psi\rangle\langle\psi|_{B_i\cup M_i\cup F_i}P^K_{F_i}\right)C^\dagger_{L-Wrap_i}\Big)\\
&\otimes C_{R-Wrap_i}\operatorname{tr}_{B_i}\left(|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\otimes C'_{R_i}|0_{R_i}\rangle\langle 0_{R_i}|(C')^\dagger_{R_i}\right)C^\dagger_{R-Wrap_i}\Big\|\\
&=\Big\|\Big(\lambda_1^i C_{L-Wrap_i}\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\right)C^\dagger_{L-Wrap_i}\\
&-C_{L-Wrap_i}\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes P^K_{F_i}|\psi\rangle\langle\psi|_{B_i\cup M_i\cup F_i}P^K_{F_i}\right)C^\dagger_{L-Wrap_i}\Big)\Big\|\\
&\cdot\Big\|C_{R-Wrap_i}\operatorname{tr}_{B_i}\left(|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\otimes C'_{R_i}|0_{R_i}\rangle\langle 0_{R_i}|(C')^\dagger_{R_i}\right)C^\dagger_{R-Wrap_i}\Big\|\\
&=\Big\|\Big(\lambda_1^i\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}\right)\\
&-\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes P^K_{F_i}|\psi\rangle\langle\psi|_{B_i\cup M_i\cup F_i}P^K_{F_i}\right)\Big)\Big\|\\
&\cdot\Big\||w_1\rangle\langle w_1|_{F_i}\otimes C'_{R_i}|0_{R_i}\rangle\langle 0_{R_i}|(C')^\dagger_{R_i}\Big\|\\
&=\Big\|\operatorname{tr}_{F_i}\left(C'_{L_i}|0_{L_i}\rangle\langle 0_{L_i}|(C')^\dagger_{L_i}\otimes\left(\lambda_1^i|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}-P^K_{F_i}|\psi\rangle\langle\psi|_{B_i\cup M_i\cup F_i}P^K_{F_i}\right)\right)\Big\|\\
&\cdot\Big\||w_1\rangle\langle w_1|_{F_i}\otimes|0_{R_i}\rangle\langle 0_{R_i}|\Big\|\\
&\leq\Big\|\lambda_1^i|w_1\rangle\langle w_1|_{B_i}\otimes|w_1\rangle\langle w_1|_{F_i}-P^K_{F_i}|\psi\rangle\langle\psi|_{B_i\cup M_i\cup F_i}P^K_{F_i}\Big\|\\
&\leq 2\lambda_1^i f(n)=2\lambda_1^i\left(\frac{1-\lambda_1^i}{\lambda_1^i}\right)^K\leq 2\left(\frac{1-\lambda_1^i}{\lambda_1^i}\right)^K
\end{aligned}
$$

Here the first equality follows by definition (Equation 106), the second equality follows by regrouping terms, and the third equality follows by the tensor product structure.

The proof for the bound on G_2 is extremely similar to the bound on G_1, and so we will not repeat the argument.

□

A.2 Proofs for Statements in Section 5

Lemma (Restatement of Lemma 30)**.**

$$\begin{aligned}
&\left\| \sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \left(\frac{1}{(\kappa^{i}_{T,\epsilon_2}\kappa^{j}_{T,\epsilon_2})^{4K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \right.\right. \\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma} \Pi^{K}_{F_k} \left\langle 0_{M_k} \right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left| 0_{M_k} \right\rangle \Pi^{K}_{F_k} \right), \frac{\epsilon}{2^{\Delta}} \right) \\
&\left.\left. - \left\langle 0_{ALL} \middle| \Psi_{\{i,j\}\cup\sigma} \right\rangle \left\langle \Psi_{\{i,j\}\cup\sigma} \middle| 0_{ALL} \right\rangle \right) \right\| \\
&\leq E_3(n,K,T,\epsilon_2,\epsilon) + 16 f(S,\eta-1,\Delta,\epsilon),
\end{aligned} \tag{111}$$

where

$$E_3(n,K,T,\epsilon_2,\epsilon) \equiv O\left(2^{\Delta}(2e(n))^{K} + 2^{\Delta}K\left(e(n)^{2T} + \epsilon_2 \right) + \epsilon \right)$$

Proof.

$$
\begin{aligned}
&\Bigg\| \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} (-1)^{|\sigma|+1} \Bigg(\frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{2K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right), \frac{\epsilon}{2^\Delta}\right) - \left\langle 0_{ALL}\middle|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\middle|0_{ALL}\right\rangle \Bigg)\Bigg\| \\
&\le \Bigg\| \frac{1}{(\lambda^i_1\lambda^j_1)^{4K+1}} \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} (-1)^{|\sigma|+1} \Bigg(\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right), \frac{\epsilon}{2^\Delta}\right) \\
&- \langle 0_{ALL}|\,\phi_{L,i}\,|0_{ALL}\rangle\cdot\langle 0_{ALL}|\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right)|0_{ALL}\rangle\cdot\langle 0_{ALL}|\,\phi_{j,R}\,|0_{ALL}\rangle \Bigg)\Bigg\| \\
&+ \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} \Bigg\| \left\langle 0_{ALL}\middle|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\middle|0_{ALL}\right\rangle \\
&- \frac{1}{(\lambda^i_1\lambda^j_1)^{4K+1}} \langle 0_{ALL}|\,\phi_{L,i}\,|0_{ALL}\rangle\cdot\langle 0_{ALL}|\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right)|0_{ALL}\rangle\cdot\langle 0_{ALL}|\,\phi_{j,R}\,|0_{ALL}\rangle \Bigg\| \\
&+ \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} \Bigg\| \left(\frac{1}{(\lambda^i_1\lambda^j_1)^{4K+1}} - \frac{1}{(\kappa^i_{T,\epsilon_2}\kappa^j_{T,\epsilon_2})^{4K+1}} \right) \\
&\cdot \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1)\cdot\mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right), \frac{\epsilon}{2^\Delta}\right) \Bigg\| \\
&\le \Bigg\| \frac{1}{(\lambda^i_1\lambda^j_1)^{4K+1}} \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} (-1)^{|\sigma|+1} \Big(\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right), \frac{\epsilon}{2^\Delta}\right) \\
&- \langle 0_{ALL}|\,\phi_{L,i}\,|0_{ALL}\rangle\cdot\langle 0_{ALL}|\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right)|0_{ALL}\rangle\cdot\langle 0_{ALL}|\,\phi_{j,R}\,|0_{ALL}\rangle \Bigg)\Bigg\| \\
&+ G_1 + G_2
\end{aligned} \tag{112}
$$

Where,

$$
\begin{aligned}
G_1 \equiv &\sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})} \Bigg\| \left\langle 0_{ALL}\middle|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\middle|0_{ALL}\right\rangle \\
&- \frac{1}{(\lambda^i_1\lambda^j_1)^{4K+1}} \langle 0_{ALL}|\,\phi_{L,i}\,|0_{ALL}\rangle\cdot\langle 0_{ALL}|\left(\otimes_{k\in\sigma}\Pi^K_{F_k}\left\langle 0_{M_k}\right|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}\left|0_{M_k}\right\rangle\Pi^K_{F_k}\right)|0_{ALL}\rangle\cdot\langle 0_{ALL}|\,\phi_{j,R}\,|0_{ALL}\rangle \Bigg\|
\end{aligned}
$$

and

$$G_2 \equiv \sum_{\sigma \in \mathcal{P}(\{j-1,\ldots,i+1\})} \Bigg\| \left(\frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} - \frac{1}{(\kappa_{T,\epsilon_2}^i \kappa_{T,\epsilon_2}^j)^{4K+1}} \right)$$

$$\cdot \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \cdot \mathcal{B}\left(\left(\otimes_{k \in \sigma} \Pi_{F_k}^K \left\langle 0_{M_k} \right| \right) \phi_{i,j} \left(\otimes_{k \in \sigma} \left| 0_{M_k} \right\rangle \Pi_{F_k}^K \right), \frac{\epsilon}{2^{\Delta}} \right) \Bigg\|$$

Later we will bound the size of G_1 and G_2 using Lemmas 20, and 26 respectively. For now we carry them along in our calculation. So, continuing where we left off in Equation 112:

$$
\begin{aligned}
&\Bigg\| \sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \Bigg(\frac{1}{(\kappa^{i}_{T,\epsilon_2}\kappa^{j}_{T,\epsilon_2})^{2K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right), \frac{\epsilon}{2^{\Delta}} \right) - \left\langle 0_{ALL} \middle| \Psi_{\{i,j\}\cup\sigma} \right\rangle \left\langle \Psi_{\{i,j\}\cup\sigma} \middle| 0_{ALL} \right\rangle \Bigg) \Bigg\| \\
&\leq \Bigg\| \frac{1}{(\lambda^{i}_{1}\lambda^{j}_{1})^{4K+1}} \sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \big(\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right), \frac{\epsilon}{2^{\Delta}} \right) \\
&- \left\langle 0_{ALL}\right| \phi_{L,i} \left|0_{ALL}\right\rangle \cdot \left\langle 0_{ALL}\right| \left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right) \left|0_{ALL}\right\rangle \cdot \left\langle 0_{ALL}\right| \phi_{j,R} \left|0_{ALL}\right\rangle \Big) \Bigg\| \\
&+ G_1 + G_2 \\
&\leq \Bigg\| \frac{1}{(\lambda^{i}_{1}\lambda^{j}_{1})^{4K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \big(\\
&\cdot \mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right), \frac{\epsilon}{2^{\Delta}} \right) \\
&- \left\langle 0_{ALL}\right| \left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right) \left|0_{ALL}\right\rangle \Big) \Bigg\| \\
&+ \Bigg\| \frac{1}{(\lambda^{i}_{1}\lambda^{j}_{1})^{4K+1}} \Big(\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) - \left\langle 0_{ALL}\right| \phi_{L,i} \left|0_{ALL}\right\rangle \cdot \left\langle 0_{ALL}\right| \phi_{j,R} \left|0_{ALL}\right\rangle \Big) \\
&\cdot \sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \left\langle 0_{ALL}\right| \left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right) \left|0_{ALL}\right\rangle \Bigg\| \\
&+ G_1 + G_2 \\
&\leq 2^{\Delta}\cdot\frac{\epsilon}{2^{\Delta}}\cdot \left\| \frac{1}{(\lambda^{i}_{1}\lambda^{j}_{1})^{4K+1}} \mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) \right\| \\
&+ \Bigg\| \frac{1}{(\lambda^{i}_{1}\lambda^{j}_{1})^{4K+1}} \Big(\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1) - \left\langle 0_{ALL}\right| \phi_{L,i} \left|0_{ALL}\right\rangle \cdot \left\langle 0_{ALL}\right| \phi_{j,R} \left|0_{ALL}\right\rangle \Big) \\
&\cdot \sum_{\sigma\in\mathcal{P}(\{j-1,\ldots,i+1\})} (-1)^{|\sigma|+1} \left\langle 0_{ALL}\right| \left(\otimes_{k\in\sigma}\Pi^{K}_{F_k} \left\langle 0_{M_k}\right| \right) \phi_{i,j} \left(\otimes_{k\in\sigma} \left|0_{M_k}\right\rangle \Pi^{K}_{F_k} \right) \left|0_{ALL}\right\rangle \Bigg\| \\
&+ G_1 + G_2
\end{aligned}
$$

$$\leq \epsilon \cdot \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \right\|$$
$$+ \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \left(\mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right) \cdot \langle 0_{ALL} | \phi_{i,j} | 0_{ALL} \rangle \right\|$$
$$+ \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \left(\mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right) \right.$$
$$\left. \cdot \left(\langle 0_{ALL} | \phi_{i,j} | 0_{ALL} \rangle - \sum_{\sigma \in \mathcal{P}(\{j-1,...,i+1\})} (-1)^{|\sigma|+1} \langle 0_{ALL} | \left(\otimes_{k \in \sigma} \Pi_{F_k}^K \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k \in \sigma} | 0_{M_k} \rangle \Pi_{F_k}^K \right) | 0_{ALL} \rangle \right) \right\|$$
$$+ G_1 + G_2$$
$$\leq G_1 + G_2 + G_3 + \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \left(\mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right) \right\| \cdot$$
$$\left(\left\| \langle 0_{ALL} | \phi_{i,j} | 0_{ALL} \rangle - \sum_{\sigma \in \mathcal{P}(\{j-1,...,i+1\})} (-1)^{|\sigma|+1} \langle 0_{ALL} | \left(\otimes_{k \in \sigma} \Pi_{F_k}^K \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k \in \sigma} | 0_{M_k} \rangle \Pi_{F_k}^K \right) | 0_{ALL} \rangle \right\| + 1 \right)$$
$$\leq G_1 + G_2 + G_3 + G_4$$

Where

$$G_3 \equiv \epsilon \cdot \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \right\|$$

and

$$G_4 \equiv \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \left(\mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right) \right\| \cdot$$
$$\left(\left\| \langle 0_{ALL} | \phi_{i,j} | 0_{ALL} \rangle - \sum_{\sigma \in \mathcal{P}(\{j-1,...,i+1\})} (-1)^{|\sigma|+1} \langle 0_{ALL} | \left(\otimes_{k \in \sigma} \Pi_{F_k}^K \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k \in \sigma} | 0_{M_k} \rangle \Pi_{F_k}^K \right) | 0_{ALL} \rangle \right\| + 1 \right)$$

We will now prove the bounds: $G_1 \leq O(2^{\Delta}(2e(n))^K) =$, $G_2 \leq O\left(2^{\Delta} K \left(e(n)^{2T} + \epsilon_2\right)\right)$, $G_3 \leq O(\epsilon)$, $G_4 \leq 8(1 + (2e(n) + 2f(n))^{\Delta-2}) f(S, \eta - 1, \Delta, \epsilon) = 16 \cdot f(S, \eta - 1, \Delta, \epsilon)$. The desired result follows from these bounds, so all that remains is to prove them, which we do below.

We begin by bounding G_1. For any fixed subset $\sigma \in \mathcal{P}(\{j-1, ..., i+1\})$ we know, by using two applications of Lemma 20 that:

$$\left\| \left\langle 0_{ALL} \middle| \Psi_{\{i,j\} \cup \sigma} \right\rangle \left\langle \Psi_{\{i,j\} \cup \sigma} \middle| 0_{ALL} \right\rangle \right.$$
$$\left. - \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \left(\otimes_{k \in \sigma} \Pi_{F_k}^K \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k \in \sigma} | 0_{M_k} \rangle \Pi_{F_k}^K \right) | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right\|$$
$$\leq 2 \cdot O((2e(n))^K) = O((2e(n))^K)$$

This follows because we can use Lemma 20 to "cut" the state $\left|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\right|$ twice, once at cut i and once at cut j, which produces the above product state, incurring error $2\cdot O((2e(n))^K)$. It follows that:

$$
\begin{aligned}
&G_1 \equiv \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})}\Bigg\| \left\langle 0_{ALL}\middle|\Psi_{\{i,j\}\cup\sigma}\right\rangle\left\langle\Psi_{\{i,j\}\cup\sigma}\middle|0_{ALL}\right\rangle \\
&-\frac{1}{(\lambda_1^i\lambda_1^j)^{4K+1}}\langle 0_{ALL}|\,\phi_{L,i}\,|0_{ALL}\rangle\cdot\langle 0_{ALL}|\left(\otimes_{k\in\sigma}\Pi_{F_k}^K\langle 0_{M_k}|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}|0_{M_k}\rangle\,\Pi_{F_k}^K\right)|0_{ALL}\rangle\cdot\langle 0_{ALL}|\,\phi_{j,R}\,|0_{ALL}\rangle\Bigg\| \\
&\le 2^{\Delta+1}O((2e(n))^K),
\end{aligned}
$$

as desired.

For the next three bounds we will repeatedly use the fact that $(\lambda_1^i)^{4K+1}=\Theta(1)=(\lambda_1^j)^{4K+1}$, and thus, $\frac{1}{(\lambda_1^i\lambda_1^j)^{4K+1}}=\Theta(1)$. The reason for this is that, we know, from the use of Lemma 13 in the error analysis of Algorithm 2, that $(\lambda_1^i)^{4K+1},(\lambda_1^j)^{4K+1}\ge 1-O(e(n))$, where $e(n)\ge(1-2^{\frac{\log(\delta)}{\log^7(n)}})=O(1/\log^4(n))$ (since $\delta\ge n^{-\log^2(n)}$ as verified in the check in the driver Algorithm 1). Since $K=O(\log^3(n))$, as specified in Algorithm 2, it follows that $(\lambda_1^i)^{4K+1}=\Theta(1)=(\lambda_1^j)^{4K+1}$.

We now bound G_2.

$$
\begin{aligned}
&G_2 \equiv \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})}\Bigg\|\left(\frac{1}{(\lambda_1^i\lambda_1^j)^{4K+1}}-\frac{1}{(\kappa_{T,\epsilon_2}^i\kappa_{T,\epsilon_2}^j)^{4K+1}}\right) \\
&\cdot\mathcal{A}(S_{L,i},\eta-1)\cdot\mathcal{A}(S_{j,R},\eta-1)\cdot\mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi_{F_k}^K\langle 0_{M_k}|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}|0_{M_k}\rangle\,\Pi_{F_k}^K\right),\frac{\epsilon}{2^\Delta}\right)\Bigg\| \\
&\le \sum_{\sigma\in\mathcal{P}(\{j-1,\dots,i+1\})}\left\|\left(\frac{1}{(\lambda_1^i\lambda_1^j)^{4K+1}}-\frac{1}{(\kappa_{T,\epsilon_2}^i\kappa_{T,\epsilon_2}^j)^{4K+1}}\right)\right\| \\
&=2^{\Delta-2}\left\|\left(\frac{1}{(\lambda_1^i\lambda_1^j)^{4K+1}}-\frac{1}{(\kappa_{T,\epsilon_2}^i\kappa_{T,\epsilon_2}^j)^{4K+1}}\right)\right\|=2^{\Delta-2}\left\|\left(\frac{(\lambda_1^i\lambda_1^j)^{4K+1}-(\kappa_{T,\epsilon_2}^i\kappa_{T,\epsilon_2}^j)^{4K+1}}{(\lambda_1^i\lambda_1^j)^{4K+1}(\kappa_{T,\epsilon_2}^i\kappa_{T,\epsilon_2}^j)^{4K+1}}\right)\right\| \\
&=O(2^\Delta)\left\|(\lambda_1^i\lambda_1^j)^{4K+1}-(\kappa_{T,\epsilon_2}^i\kappa_{T,\epsilon_2}^j)^{4K+1}\right\|=O\left(2^\Delta(4K+1)\left(|\lambda_1^i-\kappa_{T,\epsilon_2}^i|+|\lambda_1^j-\kappa_{T,\epsilon_2}^j|\right)\right) \\
&\le O\left(2^\Delta K\left(\frac{e(n)^{2T}+\epsilon_2}{(\lambda_1^i)^{2T+1}}\right)\right)=O\left(2^\Delta K\left(e(n)^{2T}+\epsilon_2\right)\right)
\end{aligned}
$$

Where the first inequality follows because, by definition, $\mathcal{A}(S_{L,i},\eta-1),\mathcal{A}(S_{j,R},\eta-1),\mathcal{B}\left(\left(\otimes_{k\in\sigma}\Pi_{F_k}^K\langle 0_{M_k}|\right)\phi_{i,j}\left(\otimes_{k\in\sigma}|0_{M_k}\rangle\,\Pi_{F_k}^K\right),\frac{\epsilon}{2^\Delta}\right)=O(1)$ (since each is a close approximation of a quantum state amplitude squared, which is at most 1 by definition). The remaining steps follow by using the fact that $(\lambda_1^i)^{4K+1}=\Theta(1)=(\lambda_1^j)^{4K+1}$ as discussed above (note that $(\lambda_1^i)^{2T}=\Theta(1)$ for the same reason, since $T=O(\log^3(n))$), and by using Lemma 26 which gives the error bound for how well the κ terms approximate the λ terms.

We now bound G_3:

$$G_3 \equiv \epsilon \cdot \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) \right\| \leq O(\epsilon)$$

Where we have used that $(\lambda_1^i)^{4K+1} = \Theta(1) = (\lambda_1^j)^{4K+1}$, and $\mathcal{A}(S_{L,i}, \eta - 1), \mathcal{A}(S_{j,R}, \eta - 1) = O(1)$, for the same reasons as in the bound of G_2.

We now bound G_4:

$$\begin{aligned}
&G_4 \equiv \left\| \frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \left(\mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right) \right\| \cdot \\
&\left(\left\| \langle 0_{ALL} | \phi_{i,j} | 0_{ALL} \rangle - \sum_{\sigma \in \mathcal{P}(\{j-1,\dots,i+1\})} (-1)^{|\sigma|+1} \langle 0_{ALL} | \left(\otimes_{k \in \sigma} \Pi_{F_k}^K \langle 0_{M_k} | \right) \phi_{i,j} \left(\otimes_{k \in \sigma} | 0_{M_k} \rangle \Pi_{F_k}^K \right) | 0_{ALL} \rangle \right\| + 1 \right) \\
&\leq 4 \left\| \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right\| \cdot \\
&\left((2e(n) + 2f(n))^{\Delta - 2} + 1 \right) \\
&\leq 8 \left\| \mathcal{A}(S_{L,i}, \eta - 1) \cdot \mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle \cdot \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle \right\| \\
&\leq 8 \cdot 2 \cdot f(S, \eta - 1, \Delta, \epsilon) = 16 f(S, \eta - 1, \Delta, \epsilon)
\end{aligned}$$

Here the first inequality follows by our previous argument that $\frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} = \Theta(1)$, as well as Lemma 20. (In fact, since we find it desirable to have an explicit constant for this particular error term, we are using $\frac{1}{(\lambda_1^i \lambda_1^j)^{4K+1}} \leq 4$, which the reader may verify, although we emphasize that the value of this constant does not matter for the asymptotic scaling and is only used for simplicity of presentation elsewhere in this paper.) Note that our use of Lemma 20, will simple, was key here in order to avoid a factor of 2^Δ appearing in the bound of G_4. The second inequality follows because the bound $(2e(n) + 2f(n))^{\Delta - 2} = o(1)$ is immediate (in fact, since $e(n), f(n) = o(1)$, and $\Delta = \Theta(\log(n))$, this quantity actually quite small, but here we only need that it is $o(1)$). The final inequality follows by two uses of the definition of $f(S, \eta - 1, \Delta, \epsilon)$, which, we recall, is defined, recursively, to be the error bound on $\mathcal{A}(\cdot, \eta - 1)$, so that $f(S, \eta - 1, \Delta, \epsilon) \geq |\mathcal{A}(S_{L,i}, \eta - 1) - \langle 0_{ALL} | \phi_{L,i} | 0_{ALL} \rangle|$, and $f(S, \eta - 1, \Delta, \epsilon) \geq |\mathcal{A}(S_{j,R}, \eta - 1) - \langle 0_{ALL} | \phi_{j,R} | 0_{ALL} \rangle|$ by definition. (This final step also uses the triangle inequality, and the facts that $\mathcal{A}(S_{L,i}, \eta - 1), \mathcal{A}(S_{j,R}, \eta - 1) = O(1)$, etc).

Now that we have bounded G_1, G_2, G_3, and G_4, the proof is complete.

□

Lemma (Restatement of Lemma 31)**.** *The error function $f(S, \eta, \Delta, \epsilon)$ obeys the following bound:*

$$\begin{aligned}
f(S, \eta, \Delta, \epsilon) \leq &\frac{\Delta^{3\eta} - 1}{\Delta^3 - 1} \left(\Delta^3 e(n)^T + O\left(e(n)^\Delta + \Delta \cdot e(n)^T \right) \right) \\
&+ \left(\Delta^3 \frac{\Delta^{3\eta} - 1}{\Delta^3 - 1} + \Delta^{3\eta} \right) \epsilon
\end{aligned} \tag{113}$$

Proof. We will now use recursive calls of Equation 47 to derive an upper bound for $f_4(S, \eta, \Delta, \epsilon)$. For simplicity we will use the substitution

$$x = e(n) \tag{114}$$

so that Equation 47 can be written as

$$f(S, \eta, \Delta, \epsilon) \leq O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left(x^T + f(S, \eta - 1, \Delta, \epsilon) + \epsilon\right) \tag{115}$$

Also note that when $\eta = 0$ it follows from the definition of Algorithm 2 that the algorithm uses subroutine $\mathcal{B}(S, \epsilon)$ to compute the desired quantity to within error ϵ. Therefore,

$$f(S, 0, \Delta, \epsilon) = \epsilon. \tag{116}$$

Repeated applications of Equation 115 yield

$$\begin{aligned}
f(S, \eta, \Delta, \epsilon) &\leq O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left(x^T + f(S, \eta - 1, \Delta, \epsilon) + \epsilon\right) && (117)\\
&\leq O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left(x^T + O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3\left(x^T + f(S, \eta - 2, \Delta, \epsilon) + \epsilon\right) + \epsilon\right) && (118)\\
&\leq (1 + \Delta^3) O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left((1 + \Delta^3) x^T + \Delta^3 f(S, \eta - 2, \Delta, \epsilon) + (1 + \Delta^3)\epsilon\right) && (119)\\
&\leq (1 + \Delta^3 + \Delta^6) O\left(x^{\Delta} + \Delta x^{T}\right) \\
&\quad + \Delta^3 \left((1 + \Delta^3 + \Delta^6) x^T + \Delta^6 f(S, \eta - 3, \Delta, \epsilon) + (1 + \Delta^3 + \Delta^6)\epsilon\right) && (120)\\
&= (\sum_{k=0}^{2} \Delta^{3k}) O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left((\sum_{k=0}^{2} \Delta^{3k}) x^T + \Delta^6 f(S, \eta - 3, \Delta, \epsilon) + (\sum_{k=0}^{2} \Delta^{3k})\epsilon\right) && (121)\\
&\leq (\sum_{k=0}^{3} \Delta^{3k}) O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left((\sum_{k=0}^{3} \Delta^{3k}) x^T + \Delta^9 f(S, \eta - 4, \Delta, \epsilon) + (\sum_{k=0}^{3} \Delta^{3k})\epsilon\right) && (122)\\
&\vdots && (123)
\end{aligned}$$

We can continue this recursion up to η times until we reach

$$\begin{aligned}
f(S, \eta, \Delta, \epsilon) &\leq (\sum_{k=0}^{\eta-1} \Delta^{3k}) O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left((\sum_{k=0}^{\eta-1} \Delta^{3k}) x^T + \Delta^{3(\eta-1)} f(S, \eta - \eta, \Delta, \epsilon) + (\sum_{k=0}^{\eta-1} \Delta^{3k})\epsilon\right) && (124)\\
&= (\sum_{k=0}^{\eta-1} \Delta^{3k}) O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left((\sum_{k=0}^{\eta-1} \Delta^{3k}) x^T + \Delta^{3(\eta-1)} f(S, 0, \Delta, \epsilon) + (\sum_{k=0}^{\eta-1} \Delta^{3k})\epsilon\right) && (125)\\
&\leq (\sum_{k=0}^{\eta-1} \Delta^{3k}) O\left(x^{\Delta} + \Delta x^{T}\right) + \Delta^3 \left((\sum_{k=0}^{\eta-1} \Delta^{3k}) x^T + \Delta^{3(\eta-1)} \epsilon + (\sum_{k=0}^{\eta-1} \Delta^{3k})\epsilon\right) && (126)
\end{aligned}$$

Note that the summation term is a geometric sequence with ratio $r = \Delta^3$. The solution to a geometric sum in r is given by

$$\sum_{j=1}^{p-1} r^j = \frac{1-r^p}{1-r} = \frac{r^p-1}{r-1} \tag{127}$$

From this, our bound can be written as

$$f(S,\eta,\Delta,\epsilon) \leq \frac{\Delta^{3\eta}-1}{\Delta^3-1} O\left(x^\Delta + \Delta x^T\right) + \Delta^3 \left(\frac{\Delta^{3\eta}-1}{\Delta^3-1} x^T + \Delta^{3(\eta-1)}\epsilon + \frac{\Delta^{3\eta}-1}{\Delta^3-1}\epsilon\right) \tag{128}$$

$$= \frac{\Delta^{3\eta}-1}{\Delta^3-1}\left(\Delta^3 x^T + O\left(x^\Delta + \Delta x^T\right)\right) + \left(\Delta^3 \frac{\Delta^{3\eta}-1}{\Delta^3-1} + \Delta^{3\eta}\right)\epsilon \tag{129}$$

which is the desired result. □

A.3 Statements from Section 6

Theorem (Restatement of Theorem 34)**.** *Suppose* $\eta = \frac{-1}{3\log\left(\frac{3}{4}\right)} \log n$*. Given this depth, the run-time for Algorithm 4 will be bounded by*

$$T(n) < \Delta^3 \cdot \epsilon^{-2} \cdot poly(n) \tag{130}$$

Proof. Note that the choice for η is still positive since $\log\left(\frac{3}{4}\right)$ is negative. Using this value in Equation 59 yields:

$$T(n) < (2\Delta)^{\frac{-1}{3\log\left(\frac{3}{4}\right)}\log n}\left[R\left(\left(\frac{3}{4}\right)^{\frac{-1}{3\log\left(\frac{3}{4}\right)}\log n} n^{\frac{1}{3}}\right) + 2\Delta^3 \cdot R(|Z|) + 2\Delta \cdot \mathrm{poly}(n)\right] \tag{131}$$

$$= n^{-\frac{\log 2\Delta}{3\log\frac{3}{4}}}\left[R\left(n^{-\frac{1}{3}} n^{\frac{1}{3}}\right) + 2\Delta^3 \cdot R(|Z|) + 2\Delta \cdot \mathrm{poly}(n)\right] \tag{132}$$

$$T(n) < \Delta^3 \cdot \mathrm{poly}(n)\left[R(1) + R(|Z|) + 1\right] \tag{133}$$

From Equation 55, $R(1) = \mathrm{poly}(n) \cdot \epsilon^{-2}$. The same equality holds for $R(|Z|)$ since $|Z| = O(1)$. Thus we have the desired run-time.

□

References

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 333–342, New York, NY, USA, 2011. Association for Computing Machinery. URL: https://doi.org/10.1145/1993636.1993682, doi:10.1145/1993636.1993682.

[BFNV19] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, Feb 2019. doi:10.1038/s41567-018-0318-2.

[BGM20] Sergy Bravyi, David Gosset, and Ramis Movassagh. Classical algorithms for quantum mean values. QIP, 2020. URL: https://arxiv.org/abs/1909.11485.

[BJS11] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, Feb 2011. doi:10.1098/rspa.2010.0301.

[BMS17] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *Quantum*, 1:8, Apr 2017. URL: http://dx.doi.org/10.22331/q-2017-04-25-8, doi:10.22331/q-2017-04-25-8.

[BVHS+18] Juan Bermejo-Vega, Dominik Hangleiter, Martin Schwarz, Robert Raussendorf, and Jens Eisert. Architectures for quantum simulation showing a quantum speedup. *Phys. Rev. X*, 8:021010, Apr 2018. URL: https://link.aps.org/doi/10.1103/PhysRevX.8.021010, doi:10.1103/PhysRevX.8.021010.

[DHKLP20] Alexander M. Dalzell, Aram W. Harrow, Dax Enshan Koh, and Rolando L. La Placa. How many qubits are needed for quantum computational supremacy? *Quantum*, 4:264, May 2020. URL: http://dx.doi.org/10.22331/q-2020-05-11-264, doi:10.22331/q-2020-05-11-264.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. STOC, 2019. URL: https://arxiv.org/pdf/1806.01838.pdf.

[HZN+20] Cupjin Huang, Fang Zhang, Michael Newman, Junjie Cai, Xun Gao, Zhengxiong Tian, Junyin Wu, Haihong Xu, Huanjun Yu, Bo Yuan, Mario Szegedy, Yaoyun Shi, and Jianxin Chen. Classical simulation of quantum supremacy circuits, 2020. arXiv:2005.06787.

[Mov20] Ramis Movassagh. Quantum supremacy and random circuits. QIP, 2020. URL: https://arxiv.org/pdf/1909.06210.pdf.

[NPD+20] John Napp, Rolando L. La Placa, Alexander M. Dalzell, Fernando G. S. L. Brandao, and Aram W. Harrow. Efficient classical simulation of random shallow 2d quantum circuits, 2020. arXiv:2001.00021.

[NSC+17] Alex Neville, Chris Sparrow, Raphaël Clifford, Eric Johnston, Patrick M. Birchall, Ashley Montanaro, and Anthony Laing. Classical boson sampling algorithms with superior performance to near-term experiments. *Nature Physics*, 13(12):1153–1157, Dec 2017. doi:10.1038/nphys4270.