How to Model Unitary Oracles

Mark Zhandry
NTT Research

Abstract

We make the case for modeling unitary oracles by allowing for controlled access to the oracle as well as its conjugate transpose (inverse), but also its conjugate and transpose. Controlling and conjugate transposes are common if even standard, but conjugates and transposes appear to be non-standard. In order to justify our modeling, we give several formal examples of what goes wrong or is missed when using a more restrictive modeling. We also argue that our model is the "right" level of granularity, and that other transformations likely do not correspond to efficient computation. We also discuss other modeling choices, such as ancillas and approximation error.

Through our exploration, we uncover interesting phenomena. Examples include an attack on the recent pseudorandom unitary construction of Ma and Huang (STOC'25) if used incorrectly as a *publicly evaluatable* unitary, and a quantum complexity-theoretic separation that follows from a purely classical separation.

1 Introduction

Abstractions are ubiquitous in computer science, as they allow for modularizing various components. In many scenarios, we will treat lower-level components as oracles which can be evaluated via queries.

Recent advances have raised hopes of full-scale quantum computers in the not-too-distant future. The workhorse of quantum computation is a *unitary transformation*, and as such, oracles representing unitary abstractions will be fundamental. In this work, we explore some basic questions about the modeling of unitary oracles for cryptographic and complexity-theoretic purposes:

Question 1. *What does it mean to "efficiently implement" a unitary transformation?*

Question 2. *How should access to (efficiently implemented) oracles be modeled?*

Question 3. *For black-box separations using unitary oracles (either in cryptography or complexity-theory), how should the queries to the oracle be modeled to better-reflect the real world?*

Question 4. *For cryptographic reductions and impossibilities, how should we model the* adversary's *unitaries?*

The unifying theme of the above questions is that the oracles are abstracting underlying efficient computation that is run by individuals locally on their own quantum device. For the first two questions, we consider for example the design of algorithms, where the oracle represents a subroutine that was perhaps developed elsewhere. For the last two questions, the unifying feature is that common techniques for reasoning about complexity classes or cryptographic concepts treat the underlying adversary/cryptosystem as a black box only accessible through queries. In both cases, however, the oracle represents underlying cryptosystems that are ultimately implemented by a quantum circuit known to everyone and run by users themselves.

Our thesis: We propose the following answers to the above questions:

- Efficient implementation should mean, if possible, a small circuit computing the unitary, *including the overall global phase*, to within negligibly-small error.
- An oracle capturing efficient computation should allow, if possible, access to the controlled version of U denoted $\mathsf{C}U$, the (controlled) conjugate transpose (aka inverse) $\mathsf{C}U^\dagger$, *as well as* the (controlled) conjugate $\mathsf{C}U^*$ and transpose $\mathsf{C}U^T$.
- An oracle separation relative to a unitary U – and likewise cryptographic reductions making use of an adversary represented by a unitary – should similarly ideally allow access to $\mathsf{C}U$, $\mathsf{C}U^\dagger$, $\mathsf{C}U^*$, and $\mathsf{C}U^T$.

We provide a number of formal results supporting our proposal, namely showing what may go wrong or what gets missed with other modeling choices. We note that some of this choices are not new, and some are even somewhat standard, though perhaps not universal. However, to the best of our knowledge, ours is the first work to advocate for providing access to U^* or insisting on paying attention to global phase when implementing unitaries. Along the way, we uncover a number of interesting phenomena. For example, we uncover an attack on the recent pseudorandom unitary construction of [MH25] when used incorrectly as a *publicly-accessible* unitary,[1] and also a quantum complexity-theoretic separation that follows from a standard *classical* separation. We now give an overview of our results.

A new notion of unitary complexity (Section 3). Very recently, [BEM+23] give the notion of $\mathsf{UnitaryBQP}$, which is intended to capture efficiently implementable unitaries. However, we observe two weaknesses of their model. First, their model model does not enable controlling, meaning a unitary U may be in $\mathsf{UnitaryBQP}$, but $\mathsf{C}U$ is not. This means that their notion of efficient does not capture a common algorithmic technique of controlling access to a given unitary, including the important Hadamard test. The reason their model does not allow controlling is that it ignores global phases, so U and $e^{i\theta}U$ are equivalent in their model. But $\mathsf{C}U$ and $\mathsf{C}(e^{i\theta}U)$ are different unitaries, *even ignoring global phase*. Second, they only require (arbitrarily-small) inverse-polynomial error. We show that this leads to problems in cryptographic protocols. Namely, a cryptographic algorithm being in $\mathsf{UnitaryBQP}$ according to their definition is not enough for it to be implemented securely and efficiently. This is because in security experiments, the inverse-polynomial error may lead to an inverse-polynomial adversary advantage, breaking security. Even worse, if the experiment runs the cryptographic algorithm many times, the errors could in principle even compound to yield an overwhelming advantage. We lastly also observe that typical algorithms, such as Solovay-Kitaev, actually do give exponentially-small error, meaning that typical techniques offer stronger guarantees than their notion of $\mathsf{UnitaryBQP}$.

We therefore give a new definition of $\mathsf{UnitaryBQP}$ (or unitary complexity more generally) with several desirable features – namely it pays attention to overall phase and insists on negligibly-small error. We actually give both a negligible error and exponentially-small error variants. The exponential variant captures typical algorithms such as Solovay-Kitaev, but is sometimes stronger than necessary. We show that our notion is robust to controlling, as well as conjugating, transposing,

[1] The model of pseudorandom unitary considered in [MH25] only allows queries to a third-party who implements the unitary. Our attack when using it as a publicly-accessible unitary does not contradict their work.

and conjugate transposes. We also show that negligible error *is* sufficient for securely implementing cryptographic protocols.

Remark 5. *[BEM+23] also give an* average-case *notion of* UnitaryBQP*, and most of their formal results utilize this average-case notion. Our focus here is on the worst-case notion.*

Unitary vs Classical Oracles (Section 4). It is widely accepted that cryptographic and complexity-theoretic separations relative classical oracles are "better" than those relative to unitary oracles. This is for several reasons. First, a classical oracle separation hints at a possible standard-model instantiation, as one may be able to use cryptographic tools such as obfuscation [BGI+01, GGH+13] to heuristically obfuscate the classical function. In contrast, we currently do not know any even heuristic method to obfuscate general quantum oracles. Second, and perhaps a bit more fundamentally, one is often interested in quantum computers' ability to solve *classical* problems, and this seems better captured by a classical oracle.

Classical procedures are often combined with quantum computation to yield quantum procedures. This leads to a natural question: given a separation using a unitary oracle U, is there a classical oracle C which can be used to emulate U, so as to achieve a classical oracle separation? The hope is that a generic version of this – which builds such a C from *any* U – would allow for generically lifting unitary oracle separations to classical separations. A version of this question is the famous open Unitary Synthesis Problem [AK07], which was initially posed as a potential strategy for lifting the unitary separation between QMA and QCMA into a classical separation.

We explain that unitary synthesis, even if true, actually does not allow for generically translating unitary oracle separations to classical oracle separations. This is because algorithms would be simulating the unitary oracle for themselves given access to the underlying classical oracle, but access to the classical oracle potentially gives these algorithms more flexibility than just having the quantum oracles. This is not captured by a proof relative to the quantum oracle alone.

For cryptographic purposes, this problem has been well-understood in the classical setting for some time, and the notion of *indifferentiability* [MRH04] captures what happens when algorithms have access to the underlying oracle C being used to implement U. Using the framework of indifferentiability, we show that for any quantum oracle U, for any construction of U from a classical oracle C that is indifferentiable, it is possible to approximately construct from U the unitaries $U^*, U^\dagger, U^T$ (for an appropriate notion of "approximate").

This suggests that modeling a unitary U according to our thesis gives a more believable modeling: it corresponds to our notion of efficient computation, and is necessary if one wants to generically replace U with an indifferentiable construction relative to a classical oracle. Moreover, we make the simple observation that the usual modeling of classical oracles – when viewed as a special case of unitary oracle – automatically allows access to $U^* = U^\dagger = U^T = U$. We note that this particular result does not extend to controlled U gates, since indifferentiability does not guarantee that global phase is preserved.

Remark 6. *For complexity-theoretic separations involving witnesses, even indifferentiability does not seem sufficient. This is because the the witnesses need not be generated efficiently. Nevertheless, indifferentiability seems* necessary *for a fully generic lifting result.*

Attacking publicly-accessible random unitaries (Section 5). As a concrete application of our indifferentiability results, we consider the question of *publicly-accessible* pseudorandom unitaries. These are keyless unitaries implemented by a quantum circuit that is known to everyone,

but nevertheless "behave" as random oracles. We will call this model the "ideal random unitary model," which is an analog of the classical random oracle model [BR93] or ideal cipher model. We note that this model also appears in some theoretical physics literature (see 1.1 for discussion). Ideal random unitaries stand in contrast to *private* pseudorandom unitaries, where the entity evaluating the unitary has a secret key, and pseudorandomness only holds to those who do not know the key.

Recently, [MH25] showed how to construct a (private) pseudorandom unitary from an underlying quantum-secure pseudorandom function (PRF) and pseudorandom permutation (PRP). A natural question is whether the underlying PRF/PRP can be replaced by *public* random functions/permutations (that is, the random oracle model and ideal cipher model[2]) to give an ideal random unitary. This would be a quantum analog of the fundamental classical result that ideal random functions imply ideal random permutations [CPS08, HKT11].We resolve this question negatively: since this construction is built from a classical oracle, it must be possible to query conjugates. We crucially leverage conjugates to give an attack. Note that this does not contradict the security proof of [MH25], as they consider only the standard pseudoarndom unitary case, where the circuit contains a key that is known only to the evaluator.

Quantum Black-Box Reductions (Section 6). We now turn to black-box constructions and reductions in the quantum setting. We show a simple method to generically increase the stretch of pseudorandom state generators (PRS) with 1-time security meeting certain statistical requirements. The idea is to start from the pseudorandom state $|\psi\rangle$, and construct the state

$$|\psi'\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + |1\rangle|\psi^*\rangle\right)$$

This can be easily constructed by conjugating and controlling the circuit which computes $|\psi\rangle$ (assuming the circuit produces no junk). We show that, under a certain (admittedly non-standard) anti-correlation property between $|\psi\rangle$ and $|\psi^*\rangle$, this new PRS is 1-time secure, and it has stretched the output by an additional qubit.

This method conjugates the construction to construct $|\psi^*\rangle$, and the security proof likewise requires conjugating the adversary. We moreover show that conjugating the adversary is *inherent* to the proof.

Thus, any model of black-box constructions and reductions should allow access to the conjugate of the construction and adversary, lest it fails to capture natural techniques. We note that techniques from the literature involving rewinding (e.g. [Unr12, CMSZ22]) already utilize conjugate-transposes and controlling. However, to the best of our knowledge ours is the first to consider conjugates.

Homomorphisms on Unitaries (Section 7). Above we argued for a more fine-grained modeling of unitaries that more-closely approximates actual efficient unitary computation. However, the unitaries abstracted by the oracles still give less access than the full circuit description of the unitary. This is often inherent, especially in the setting of black-box separations. But then a natural question is: why stop at our modeling, and not try to give an even more-refined model?

We argue that our model captures the "right" level of granularity. In particular, we observe that the reason controlling and conjugating can be performed efficiently on efficient circuits is exactly

[2]Here, the model gives superposition-access to these ideal primitives [BDF+11].

because these operations are *homomorphisms* on unitaries, which means they can be applied gate-by-gate while keeping the overall circuit structure. Likewise, conjugate-transposes and transposes are efficient because they are antihomomorphisms.

We show, roughly, that these are the *only* (anti)homomorphisms that can be carried out efficiently. Thus any more fine-grained model that utilizes homomorphisms is attempting to capture inefficient computation.

From Classical To Quantum Hardness (Section 8). One caveat to our claim that these are the only efficient homomorphisms is that it only holds true if the original unitary has ancillas (even a single qubit suffices). What about homomorphisms on unitaries that do not employ ancillas?

In this case, more operations *are* possible, including taking the determinant. Interestingly, we show that this leads to a *quantum* complexity-theoretic separation, under a purely *classical* separation assumption. Namely, we show, under the assumption that $\mathsf{PH} \not\subseteq \mathsf{BPP}$, that there are unitaries that can be implemented efficiently with just 2 qubit ancillas (to within exponentially-small error), but *cannot* be implemented efficiently without ancillas (also with exponentially-small error). Our proof works by taking a supposed ancilla-free implementation, which is a quantum circuit that has a classical description, and computing the determinant of that circuit. The resulting value solves a classical problem, and the algorithm to compute the circuit description and take the determinant is purely classical.

Remark 7. *We caution the reader to not over-interpret such a result. While our separation is about quantum complexity classes, because they are about ancilla complexity, they are of a fundamentally different nature than the "hard" separations like* BPP *vs* QMA *or the classical analogs* P *vs* NP*. In particular, we are not aware of any major barriers to proving our separation unconditionally, though we are also not aware of any such unconditional separation. Nevertheless, the most natural way to prove our separation seems to utilize the classical complexity-theoretic separation.*

Remark 8. *We also note that our separation is sensitive to the error model. If the errors are allowed to be only negligibly small but not exponential, taking the determinant accumulates too many errors and gives a meaningless answer. Our result crucially relies on the errors being exponential.*

1.1 Applications to Physics

While our focus was on crytographic applications, we believe that our modeling may be the right model in some physics settings, and here give some rough criteria for when our modeling should be used.

Chaotic quantum systems, especially black holes, seem to be good scramblers of information. As information cannot truly be lost, this scrambling must be *complexity-theoretic*, meaning it is only computationally difficult to un-scramble, but information-theoretically decodable. Viewing the scrambling inside black holes as complexity-theoretic has helped physicists understand phenomena such as the black hole information paradox.

One way to justify the complexity-theoretic scrambling is to model the dynamics of these chaotic systems as black-box Haar-random unitaries. This is the approach taken, for example, by [BFV20, PRV24]. Such a model is an "idealization" of the real-world as the dynamics should be computable by polynomial-sized circuits, but Haar-random unitaries are exponential-sized objects. Nevertheless, if the dynamics are complicated enough, it seems reasonable to suppose that the only thing an observer can do with the circuit representing the black hole is evaluate it on

states of its choice – in this case, the idealization seems reasonable. The situation is somewhat analogous to idealized models in cryptography such as the classical random oracle model. In this modeling, our results seem applicable: one should model the unitary by having access to (controlled) $U, U^\dagger, U^*, U^T$. Note that [BFV20, PRV24] only consider (controlled) $U, U^\dagger$. The intuition for modelling $U^\dagger$ is that the dynamics can be simulated with time reversed to invert the unitary.

Other works take different approaches. For example, [YE23, AEH+24] model the dynamics as a *pseudorandom* unitary. The key difference from the modeling above is that pseudorandom unitaries have a secret key. This key usually considered the internal state of the black hole, which is inaccessible to the outside observer but determines the input/output behavior of the black hole. In this case, the observer does not actually have access to the circuit representing the dynamics due to part of the system being hidden. Therefore, there is no ability to simulate the dynamics - all that is possible is to actually send information through the system in the real world. As such, access to U seems sufficient, without the need to have access to $U^\dagger, U^*$ or U^T.

Based on the two examples above, the following therefore seems like a reasonable criteria for when our modeling should be used: if it makes sense to allow queries to $U^\dagger$, then one should likely allow access to (controlled) $U, U^\dagger, U^*, U^T$. On the other hand, if the modeling does not need to allow queries to $U^\dagger$, then it may also be unnecessary to allow queries to U^*, U^T.

2 Preliminaries

2.1 Conventions for quantum gates and circuits

A quantum gate is simply a unitary, usually on a small number of qubits, representing a subsystem of the overall system. Given a unitary/gate U, we will denote by $\mathsf{C}U$ the controlled gate, defined on basis states $|b\rangle|\psi\rangle$ for $b \in \{0,1\}$ as $\mathsf{C}U(|b\rangle|\psi\rangle) = |b\rangle U^b|\psi\rangle$.

A quantum circuit is a circuit made of quantum gates, typically from a finite gate set. A *generalized* quantum circuit is one that also allows for initializing new qubits, measuring qubits, and discarding them. To make it clear, we will sometimes use "unitary quantum circuit" or simply "unitary circuit" to denote a quantum circuit that is not generalized. The Principle of Deferred Measurements indicates that all measurements can be deferred to the end of the computation, at the cost of using ancillas. We can also always initialize any needed ancillas at the beginning of a computation. Thus, a generalized quantum circuit can always be thought of as a unitary circuit acting on a larger system, where the extra qubits (called ancillas) are initialized to $|0\rangle$.

For any quantum state $|\psi\rangle = \sum_x \alpha_x |x\rangle$, let $|\psi^*\rangle = \sum_x \alpha_x^* |x\rangle$, where $\{|x\rangle\}$ is the computational basis. Likewise for any unitary U, let U^* denote the unitary obtained by conjugating each amplitude in the transition matrix when written in the computational basis.

Paying attention to global phases. The literature usually treats two unitaries/ unitary circuits U, V as equivalent if they differ by an overall phase: $U = e^{i\theta}V$. This is because two quantum states that differ in an overall phase are considered identical.

However, one of the key points of this work is that there are certain operations that can be performed on quantum circuits: in particular given any unitary circuit U, one can also build a controlled circuit $\mathsf{C}U$ simply by controlling each gate individually, and then building each controlled gate from the underlying gate set. However, we point out that the controlled unitaries $\mathsf{C}U$ and $\mathsf{C}(e^{i\theta}U)$ are *not* equivalent, even if one ignores global phase for the controlled unitaries. For

example, if $U = \mathbf{I}$, then $\mathsf{C}U = \mathbf{I}$, but $\mathsf{C}(e^{i\theta}U) = \mathsf{P}(\theta) \otimes \mathbf{I}$. This is because the overall phase in $e^{i\theta}U$ becomes a relative phase once controlled.

The takeaway is that, if we have a unitary circuit V that realizes a unitary U, and if we want to be able to realize the unitary $\mathsf{C}U$, it is important that we actually pay attention to overall phase, either insisting that V, U are identical unitaries, or at least that we know the overall global phase difference between them.

Note that the usual notion of a "universal" gate set only requires that the gate set can arbitrarily-closely approximate any unitary up to global phase. In general, we would not expect a universal gate set to actually approximate all unitaries when global phase is taken into account. Indeed, the Clifford+T gate set is universal, but all gates in this set have determinants that are powers of i. Hence any circuit built from these gates has determinants in the discrete set of powers of i[3]. Thus, such circuits cannot get arbitrarily close to all unitaries once global phase is considered, since the determinant of a general unitary can be any complex number of unit norm.

Thus, in this work we will always consider a quantum circuit to additionally come with a phase term θ, and the unitary computed by the circuit is the product of all the gates and $e^{i\theta}$.

Remark 9. *An alternative would be to update the notion of universal gate set to enable approximating any unitary including global phase. However, in order to be as consistent as possible with existing notions, we do not re-define universality but instead modify the circuit model to also specify global phase.*

Paying attention to ancillas. The use of ancillas will also be important for this work. When defining decisional complexity classes such as BQP, it is fine to imagine the ancillas being initialized to $|0\rangle$, computed on, and then simply discarded at the end. However, for general quantum processes mapping quantum inputs to quantum outputs, we need to ensure that the ancillas do not become entangled with the output. This is, for example, important for controlling unitaries, for similar reasons as above. For this reason, we will typically insist that the ancillas are returned to contain $|0\rangle$. This gives rise to the following notion of *extensions* of unitaries:

Definition 10. *Let U be a unitary acting on n qubits. Let $\overline{U}$ be a unitary acting on $n + a$ qubits. Then we say that $\overline{U}$ is an* a-qubit extension *of U if $\overline{U}(\ |\phi\rangle|0^a\rangle\ ) = (\ U|\phi\rangle\ )|0^a\rangle$ for all states $|\phi\rangle$.*

We will often ignore the parameter a, and simply call $\overline{U}$ an *extension* of U. Recall our convention that we keep track of overall phases, so the equality in Definition 10 includes the overall global phase. Note that Definition 10 places no restriction on how $\overline{U}$ operates on states $|\phi\rangle|x\rangle$ for $x \neq 0^a$, except that unitarity implies the support of $\overline{U}(|\phi\rangle|x\rangle)$ cannot contain any terms of the form $|\psi\rangle|0^a\rangle$.

2.2 Quantum Complexity

To measure the complexity of a quantum computational problem, we first fix a universal gate set. The choice of gate set is usually arbitrary up to polynomial losses, but for concreteness we can take the Clifford+T gate set.

We follow the typical model of computation where a classical Turing machine M generates a classical description of a (potentially generalized) quantum circuit C, which is then run on an input. For decisional complexity classes like BQP, M takes as input only the instance length (written in unary), and then C is run on the instance x encoded as a quantum state $|x\rangle$.

[3] In fact, as is discussed in Section 7, the determinants of the overall unitary will typically be exactly 1.

2.3 Quantum Distance Notions

Distances between quantum states. For two pure states $|\psi\rangle = \sum_x \alpha_x |x\rangle$ and $|\phi\rangle = \sum_x \beta_x |x\rangle$, their distance is defined using the L2 norm:

$$\||\psi\rangle - |\phi\rangle\|_2 = \sqrt{\sum_x |\alpha_x - \beta_x|^2} \tag{1}$$

Consider two mixed states represented by density matrices $\rho = \sum_{x,x'} \rho_{x,x'} |x\rangle\langle x'|$ and $\tau = \sum_{x,x'} \tau_{x,x'} |x\rangle\langle x'|$. Their distance is known as the trace distances, is defined as

$$\|\rho - \tau\|_{\mathsf{Tr}} = \sum_\lambda |\lambda| \tag{2}$$

where λ ranges over the eigenvalues of $\rho - \tau$. Note that we can extend the trace distance to non-hermitian matrices, where we use the singular values instead of eigenvalues. We can relate the distance between pure states to their trace distances by $\||\psi\rangle\langle\psi| - |\tau\rangle\langle\tau|\|_{\mathsf{Tr}} \leq 2\||\psi\rangle - |\phi\rangle\|_2$.

Distances between quantum operations. A quantum channel is a general map between mixed states. The diamond distance between channels $\mathcal{E}$ and $\mathcal{F}$ is defined as

$$\|\mathcal{E} - \mathcal{F}\|_\diamond = \max_\rho \|(\mathbb{E} \otimes \mathbf{I}_N)(\rho) - (\mathcal{F} \otimes \mathbf{I}_N)(\rho)\|_{\mathsf{Tr}} \tag{3}$$

That is, it is the maximum distance between the images under the two maps of some mixed state. Note that the $\otimes \mathbf{I}_N$ ensures that the diamond norm considers also what happens when the maps are applied to a sub-system of a larger system. N can be taken to be the dimension of ρ.

For unitary transformations in particular, we can consider multiple notions of distance. The simplest is to simply view the unitary matrix as a quantum channel. This give the diamond distance for two unitary matrices U, V, denoted $\|U - V\|_\diamond$.

One observation is that the diamond distance between unitaries ignores global phase. A stronger notion of distance is therefore the operator distance

$$\|U - V\|_{op} = \max_\lambda |\lambda| \tag{4}$$

where λ ranges over the singular values of $U - V$.

We finally define an average-case distance notion.

$$\|U - V\|_{avg} = 1 - |\mathsf{Tr}[U^\dagger V]|/N \tag{5}$$

where N is the dimension of U, V.

Our average-case notion is convenient since it can actually be efficiently tested, as shown in the following few lemmas.

Lemma 11. *There exists a quantum algorithm $\mathcal{A}$ making a single query to each of U and V, such that if $\|U - V\|_{avg} = \epsilon$, then $\Pr[\mathcal{A}^{U,V}() = 1] = 1 - \Theta(\epsilon)$.*

Proof. $\mathcal{A}$ constructs two copies of a Haar-random state $|\psi\rangle$, computes $U|\psi\rangle$ and $V|\psi\rangle$, and applies the swap test to the results. For a given $|\psi\rangle$, the probability of acceptance is

$$\begin{aligned}\frac{1}{2}+\frac{|\langle\psi|V^\dagger U|\psi\rangle|^2}{2} &= \frac{1}{2}+\frac{\langle\psi|V^\dagger U|\psi\rangle\langle\psi|U^\dagger V|\psi\rangle}{2}\\ &= \frac{1}{2}+\frac{\mathsf{Tr}[(V^\dagger U)\otimes(U^\dagger V)|\psi\rangle^{\otimes 2}\langle\psi|^{\otimes 2}]}{2}\end{aligned}$$

We now average over $|\psi\rangle$. For a Haar random $|\psi\rangle$ over a system $\mathcal{H}$, $|\psi\rangle^{\otimes 2}\langle\psi|^{\otimes 2}$ is the totally-mixed state over the symmetric subspace of $\mathcal{H}^{\otimes 2}$. We choose a basis B which diagonalizes $U^\dagger V$ (and hence $V^\dagger U$) with eigenvalues λ_i. In this basis, we can write

$$\begin{aligned}\mathbb{E}_{\psi\rangle}[|\psi\rangle^{\otimes 2}\langle\psi|^{\otimes 2}] &= \frac{2}{N(N+1)}\left(\sum_i |i,i\rangle\langle i,i| + \frac{1}{2}\sum_{i<j}(|i,j\rangle+|j,i\rangle)(\langle i,j|+\langle j,i|)\right)\\ (V^\dagger U)\otimes(U^\dagger V) &= \sum_{i,j}\lambda_i\lambda_j^*|i,j\rangle\langle i,j|\end{aligned}$$

Then we have that, over the choice of $|\psi\rangle$, the probability of acceptance is:

$$\begin{aligned}&\frac{1}{2}+\frac{\sum_i \lambda_i\lambda_i^* + \frac{1}{2}\sum_{i<j}\lambda_i\lambda_j^* + \lambda_j\lambda_i^*}{N(N+1)}\\ &= \frac{1}{2}+\frac{2\sum_i \lambda_i\lambda_i^* + \sum_{i\neq j}\lambda_i\lambda_j^*}{2}\\ &= \frac{1}{2}+\frac{\sum_i \lambda_i\lambda_i^* + \sum_{i,j}\lambda_i\lambda_j^*}{2N(N+1)}\\ &= \frac{1}{2}+\frac{\sum_i |\lambda_i|^2 + |\sum_i \lambda_i|^2}{2N(N+1)}\\ &= \frac{1}{2}+\frac{N+|\mathsf{Tr}[U^\dagger V]|^2}{2N(N+1)}\\ &= \frac{1}{2}+\frac{1}{2(N+1)}+\frac{N(1-\epsilon)^2}{2(N+1)} = 1-\frac{N}{N+1}\epsilon(1-\epsilon/2) = 1-\Theta(\epsilon)\end{aligned}$$

Note that in general, choosing a Haar random state is computationally inefficient. But we can replace the generation of $|\psi\rangle$ with any 2-design instead to make the algorithm efficient. □

Lemma 12. *There exists a quantum algorithm $\mathcal{A}$ making a single query to each of U and V, such that if $\|U-V^{-1}\|_{avg}=\epsilon$, then $\Pr[A^{U,V}()=1]=1-\Theta(\epsilon)$.*

Proof. This runs the algorithm from Lemma 11 on the unitaries UV and $\mathbf{I}$. Lemma 12 follows, since $\|U-V^{-1}\|_{avg}=\|UV-\mathbf{I}\|_{avg}$. □

Lemma 13. *There exists a quantum algorithm A making a single query to each of U and V, such that if $\|U-V^*\|_{avg}=\epsilon$, then $\Pr[A^{U,V}()=1]=1-\Theta(\epsilon)$.*

Proof. $\mathcal{A}$ prepares the maximally-entangled state $|\psi\rangle = \sum_x |x,x\rangle$, applies U to one half and V to the other half, and then projects onto $|\psi\rangle$. The acceptance probability is then $|\langle\psi|U \otimes V|\psi\rangle|^2$.

We can instead think of $|\psi\rangle$ as a matrix ψ whose columns are indexed by the first subsystem and rows are indexed by the second subsystem. Then $\psi = \mathbf{I}/\sqrt{N}$. Moreover, acting on the first subsystem by V is just a left-multiplication, and acting on the right subsystem by U is just right-multiplication by U^T. Then we can write the acceptance probability as

$$\begin{aligned}|\mathsf{Tr}[\psi^\dagger V \psi U^T]|^2 &= |\mathsf{Tr}[U^T V]|^2/N^2 = |\mathsf{Tr}[U^\dagger V^*]|^2/N^2 \\ &= (1 - \|U - V^*\|_{avg})^2 = (1-\epsilon)^2 = 1 - \Theta(\epsilon)\end{aligned}$$

□

2.4 Other Useful Quantum Lemmas

Consider the state $|\phi_t\rangle$ of a quantum query algorithm when it makes its t-th quantum query. Define $q_x(|\phi_t\rangle)$ to be the magnitude squared of x in the superposition of query t, that is $q_x(|\phi_t\rangle) = \sum_y |\alpha_{x,y}|^2$. Call this the query magnitude of x. Let $q_x = \sum_t q_x(|\phi_t\rangle)$ be the total query magnitude of x. For a set S, let $q_S = \sum_{x\in S} q_x$ be the total query magnitude of S.

Lemma 14 ([BBBV97] Theorem 3.1)**.** *Suppose $\||\psi\rangle - |\phi\rangle\|_2 \leq \epsilon$. Then, performing the same measurement on $|\psi\rangle$ and $|\phi\rangle$ yields distributions with statistical distance at most 4ϵ.*

Lemma 15 ([BBBV97] Theorem 3.3 paraphrased)**.** *Let $\mathcal{A}$ be a quantum query algorithm making T queries to an oracle O. Let $\epsilon > 0$ and let S be a set such that $q_S \leq \epsilon$. Let O' be another oracle that is identical to O on all points not in S. Let $p = \Pr[A^O() = 1]$ and $p' = \Pr[A^{O'}() = 1]$. Then $|p - p'| \leq O(\sqrt{T\epsilon})$.*

Let $O_{|\psi\rangle}$ be the oracle which reflects around the state $|\psi\rangle$, which is equivalent to having an oracle which projects onto $|\psi\rangle$.

Lemma 16 ([JLS18] Theorem 4)**.** *There exists a stateful simulator $\mathcal{S}$ that approximately simulates $O_{|\psi\rangle}$ using several copies of $|\psi\rangle$. In particular, for any state $|\psi\rangle$ and for any algorithm $\mathcal{A}$ making q queries,*

$$\left|\Pr[\mathcal{A}^{O_{|\psi\rangle}}() = 1] - \Pr[\mathcal{A}^{\mathcal{S}(|\psi\rangle^\ell)}() = 1]\right| \leq O(q/\sqrt{\ell})$$

Lemma 17. *Let $|\psi\rangle$ and $|\phi\rangle$ denote independent Haar random states over n qubits. Then*

$$\left\|\mathbb{E}\left[(|\psi\rangle\langle\psi|)^{\otimes t} \otimes (|\psi^*\rangle\langle\psi^*|)^{\otimes u}\right] - \mathbb{E}\left[(|\psi\rangle\langle\psi|)^{\otimes t} \otimes (|\phi\rangle\langle\phi|)^{\otimes u}\right]\right\| \leq \frac{3(t+u)^2}{2^n}$$

.

Proof. Let ρ_0 be the density matrix obtained by averaging over $|\psi\rangle^{\otimes t}|\psi^*\rangle^{\otimes u}$. Let P be the projection onto computational basis states $x_1, \cdots, x_{t+u}$ that are all distinct. Then let ρ_1 be the re-normalized density matrix $P\rho_0 P/\mathsf{Tr}[P\rho_0 P]$.

We first compute $\mathsf{Tr}[P\rho_0 P]$. This is the same as the probability the following accepts: measure ρ_0 in the computational basis, and accept if all entries in the result are distinct. If we write $|\psi\rangle = \sum_x \alpha_x |x\rangle$, then the probability of any particular vector of outcomes $(x_1, \cdots, x_{t+u})$ is exactly $\prod_{i=1}^{t+u} |\alpha_{x_i}|^2$. This is identical to the case of receiving $t+u$ copies of $|\psi\rangle$. Since $|\psi\rangle$ is Haar random,

$t+u$ copies are identical to the totally-mixed state over the symmetric subspace of $t+u$ copies. The symmetric subspace is spanned by vectors labeled by the un-ordered multi-sets of $t+u$ terms. The quantity of such vectors is is $\binom{2^n+t+u}{t+u}$. On the other hand, the number of such multi-sets containing only distinct elements is $\binom{2^n}{t+u}$. Thus, the probability of obtaining all distinct elements is:

$$\begin{aligned}
\frac{\binom{2^n}{t+u}}{\binom{2^n+t+u}{t+u}} &= \frac{[(2^n)!]^2}{(2^n-t-u)!(2^n+t+u)!} \\
&= \left(\frac{2^n}{2^n+t+u}\cdot\frac{2^n-1}{2^n+t+u-1}\cdots\frac{2^n-t-u+1}{2^n+1}\right) && \text{if } t-u<2^n \\
&= \left(1+\frac{t+u}{2^n}\right)^{-1}\cdots\left(1+\frac{t+u}{2^n-t-u}\right)^{-1} \\
&\geq \left(1+\frac{t+u}{2^n-t-u}\right)^{-t-u} \\
&\geq \left(1-\frac{t+u}{2^n-t-u}\right)^{t+u} && \text{if } t-u\leq 2^n/3 \\
&\geq 1-(t+u)^2/(2^n-t-u) \\
&\geq 1-(2/3)(t+u)^2/2^n && \text{if } t-u\leq 2^n/3
\end{aligned}$$

Since $\mathsf{Tr}[P\rho_0 P]\geq 1-(2/3)(t+u)^2/2^n$, we therefore have that $\|\rho_0-\rho_1\|_{\mathsf{Tr}}\leq(2/3)(t+u)^2/2^n$.

Now we define ρ_2 and ρ_3. ρ_3 is the density matrix obtained by averaging over $|\psi\rangle^{\otimes t}|\phi\rangle^{\otimes u}$, and $\rho_2 = P\rho_3 P/\mathsf{Tr}[P\rho_3 P]$. By a similar argument as above, we have that $\|\rho_2-\rho_3\|_{\mathsf{Tr}}\leq(3/2)(t^2+u^2)/2^n\leq(3/2)(t+u)^2/2^n$.

Now we look at $\|\rho_2-\rho_3\|$. Both matrices have support only on the product of the symmetric subspace for the first t sets of n qubits, an the symmetric subspace for the remaining u sets of n qubits. Let $|\{x_i\}_i\rangle$ and $\rangle|\{y_i\}_i\rangle$ for un-ordered multi-sets $\{x_i\}_i$ and $\{y_i\}_i$ denote the uniform superpositions over all possible orderings of those sets. Then $|\{x_i\}_i\rangle\rangle|\{y_i\}_i\rangle$ form a basis for the symmetric subspace, and we will write ρ_2 and ρ_3 in this basis.

Since we have projected onto P, we can also keep only the subspaces where all entries are the same. We can therefore restrict our attention to $\{x_i\}_i,\{y_i\}_i$ that contain only distinct elements and have the two multi-sets themselves being disjoint.

Since ρ_3 is just the projection of the product of two totally-mixed states on the symmetric subspace, we see that ρ_3 is just the (appropriately scaled) identity matrix.

For ρ_2, let us compute some off-diagonal entry. The off-diagonal entry corresponding to basis state$|\{x_i\}_i\rangle|\{y_i\}\rangle$ and $|\{x'_i\}_i\rangle|\{y'_i\}_i\rangle$ will be

$$\prod_{i=1}^{t}\alpha_{x_i}\prod_{i=1}^{u}\alpha^*_{y_i}\prod_{i=1}^{t}\alpha^*_{x'_i}\prod_{i=1}^{u}\alpha_{y'_i}$$

Due to the phase invariance of Haar random states, averaging over α sampled from a Haar random state gives 0 unless the un-conjugated α's are exactly matched by corresponding conjugated α's. In other words, these entries average to 0 unless the multi-sets $\{x_i\}\cup\{y'_i\}$ and $\{x'_i\}\cup\{y_i\}$ are equal. But since the x_i are distinct and disjoint from the y_i (and likewise for the x'_i, y'_i), the only way for this to happen is for the multisets $\{x_i\}$ and $\{x'_i\}$ to be equal, and likewise for the multisets $\{y_i\}$ and $\{y'_i\}$ to be equal. In other worrds, ρ_2 is also the (scaled) identity matrix. Thus, $\rho_2=\rho_3$.

Using the triangle inequality on the trace differences between $\rho_0, \rho_1, \rho_2, \rho_3$ gives the lemma. □

Concentration of measure. We recall the concentration of measure lemma:

Lemma 18 (Simplified version of [Mec19])**.** *Let μ be the Haar measure on dimension N. Let f be an L-Lipshitz function in the Frobenius norm, mapping N-dimensional unitaries to real numbers. Then the following holds for every $\epsilon > 0$:*

$$\Pr_{U \leftarrow \mu}[|f(U) - \mathbb{E}_{V \leftarrow \mu}[f(V)]| \geq \epsilon] \leq 2e^{-\frac{(N-2)\epsilon^2}{24L^2}}$$

Lemma 19. *Let A^U be a quantum algorithm making q queries to a Haar random unitary U of dimension N, and producing a classical string x of length ℓ. Suppose $\ell \leq N\epsilon^2/q^2$. Let D be the distribution over x for a Haar random U, and let D_U be the distribution for a given U. Then*

$$\Pr[\|D_U - D\| \geq \epsilon] \leq e^{-O(N\epsilon^2/q^2)}$$

In particular, in the usual setting of $N = 2^n$ for polynomial n and q polynomial, we can take ϵ to be exponentially small while keeping the term on the right double-exponentially small. Thus, D_U will almost certainly be close to D.

Proof. We recall that two distributions D_1, D_2 over $\{0,1\}^\ell$ being ϵ-close means there is a subsets $S \subseteq \{0,1\}^\ell$ such that $\|\Pr_{x \leftarrow D_1}[x \in S] - \Pr[x \leftarrow D_2][x \in S]\| = \epsilon$. Thus, we fix a set $S \subseteq \{0,1\}^\ell$. We let $f(U)$ be the probability that A^U outputs an $x \in S$. Invoke Lemma 18, to conclude that

$$\Pr_{U \leftarrow \mu}[|\Pr[x \in S : x \leftarrow D_U] - \Pr[x \in S : x \leftarrow D]| \geq \epsilon] \leq 2e^{-\frac{(N-2)\epsilon^2}{96q^2}}$$

We then union-bound over all S to get that

$$\Pr[\|D_U - D\| \geq \epsilon] \leq 2^\ell \times 2e^{-\frac{(N-2)\epsilon^2}{96q^2}}$$

Then we use that $\ell \leq N\epsilon^2/q^2$ to absorb the exponent ℓ into the a Big-Oh in the exponent, giving the lemma. □

3 Unitary Complexity

In this section, we explore some modeling questions surrounding unitary complexity. In particular, we argue for some modifications to what were proposed in [BEM+23].

3.1 New Definitions of Unitary Complexity

A unitary synthesis problem is the task of implementing a family of unitaries. In [BEM+23], definitions for such complexity classes were given. Here, we give our new definitions and argue that they have certain advantages over [BEM+23].

Definition 20. *For a function $T : \mathbb{Z} \to \mathbb{Z}$ and function $\delta : \mathbb{Z} \to [0,1]$, let $\mathsf{UnitaryTime}(T, \delta)$ be the collection of families of unitaries $\{U_s\}_{s \in \{0,1\}^*}$ for which the following hold. There exists a family of unitaries $\{\overline{U}_s\}_s$ such that each $\overline{U}_s$ is an extension of U_s, together with a Turing machine $M(s)$ that runs in deterministic time $T(|s|)$ and outputs a unitary circuit C_s such that $\|C_s - \overline{U}_s\|_{op} \leq \delta(|s|)$.*

Note that since M runs in time at most T, the circuit C_s must have size at most T.

Unitary BQP, E. We define three variants of Unitary BQP, depending on how close of an approximation is desired. $\mathsf{UnitaryBQP}_{\mathsf{poly}}$ (resp. $\mathsf{UnitaryBQP}_{\mathsf{exp}}$) is the set $\cap_\delta (\cup_p \mathsf{UnitaryTime}(p,\delta))$ where p ranges over all polynomials $n^{O(1)}$ and δ ranges over all inverse-polynomials (resp. inverse exponentials $2^{-n^{O(1)}}$). That is, $\mathsf{UnitaryBQP}_{\mathsf{poly}}$ (resp. $\mathsf{UnitaryBQP}_{\mathsf{exp}}$) is the set of sequences of unitaries where, for any inverse-polynomial (resp. inverse-exponential) error δ, the unitary can be implemented to within error δ in polynomial-time. We then define $\mathsf{UnitaryBQP}_{\mathsf{negl}}$ to be the set $\cup_{p,\delta}\mathsf{UnitaryTime}(p,\delta)$ where p ranges over all polynomials $n^{O(1)}$ and δ ranges over all negligible functions $n^{-\omega(1)}$.

Define $\mathsf{UnitaryE}_{\mathsf{poly}} = \cap_\delta (\cup_c \mathsf{UnitaryTime}(2^{cn},\delta))$ where c ranges over all integers c and δ ranges over all inverse-polynomials. Likewise define $\mathsf{UnitaryE}_{\mathsf{exp}}$, $\mathsf{UnitaryE}_{\mathsf{negl}}$. We can also extend our notion to oracle-aided unitaries, by considering circuits that contain oracle gates.

There are two key differences between our definition of $\mathsf{UnitaryBQP}$ and the definition given in [BEM+23]:

- We insist that the unitary C_s is close to U_s, when considered as operators over $\mathbb{C}^{2^n}$. This means that the overall phase matters in our definition, and C_s cannot produce any extra garbage state. In contrast, [BEM+23] only ask that the circuit C_s is close to U_s as quantum channels, where the ancilla qubits of the output are traced out. This means C_s can have an arbitrary global phase, and can even produce garbage states that get traced out.
- The dependence on error in [BEM+23] matches our poly definition. However, our negl version insists on negligibly-small error, while exp insists on exponentially-small error.

In the next subsections, we discuss our new modeling choice for these complexity classes.

Ancilla Complexity. We now consider a version of unitary complexity where the circuit is restricted to having few or no ancilla qubits. Note that ancilla complexity is an important metric for quantum computation, since it captures how much extra quantum storage is needed to perform a computation.

Definition 21. *For functions $T, a : \mathbb{Z} \to \mathbb{Z}$ and function $\delta : \mathbb{Z} \to [0,1]$, let $\mathsf{UnitaryTime}^a(T,\delta)$ be the collection of families of unitaries $\{U_s\}_{s\in\{0,1\}^*}$ for which the following hold. There exists a family of unitaries $\{\overline{U}_s\}_s$ such that each $\overline{U}_s$ is an $a(|s|)$-extension of U_s, together with a Turing machine $M(s)$ that runs in deterministic time $T(|s|)$ and outputs a unitary circuit C_s such that $\|C_s - \overline{U}_s\|_{op} \leq \delta(|s|)$.*

Similarly define $\mathsf{UnitaryBQP}^a_{\mathsf{exp}}$, $\mathsf{UnitaryBQP}^a_{\mathsf{negl}}$, $\mathsf{UnitaryBQP}^a_{\mathsf{poly}}$, $\mathsf{UnitaryE}^a_{\mathsf{exp}}$, $\mathsf{UnitaryE}^a_{\mathsf{negl}}$, and $\mathsf{UnitaryE}^a_{\mathsf{poly}}$.

3.2 Features of our Definition

Proposition 22. *Let a be a function, $\mathcal{C} \in \{\mathsf{UnitaryBQP}, \mathsf{UnitaryE}\}$ and $\delta \in \{\mathsf{exp}, \mathsf{negl}, \mathsf{poly}\}$. Let $\{U_s\}_s \in \mathcal{C}^a_\delta$. Then $\{\mathsf{C}U_s\}_s$, $\{U_s^*\}_s$, $\{U_s^\dagger\}_s$, $\{U_s^T\}_s$, $\{\mathsf{C}U_s^*\}_s$, $\{\mathsf{C}U_s^\dagger\}_s$, $\{\mathsf{C}U_s^T\}_s$ are all also in $\mathcal{C}^a_\delta$.*

Proof. The proofs for all choices of $\mathcal{C}$ and δ are essentially identical, so we focus on the case of $\mathsf{UnitaryBQP}^a_{\mathsf{exp}}$. For any $\{U_s\}_s \in \mathsf{UnitaryBQP}^a_{\mathsf{exp}}$, we will prove that $\{U_s^*\}_s, \{U_s^\dagger\}_s, \{\mathsf{C}U_s\}_s$ are each in $\mathsf{UnitaryBQP}^a_{\mathsf{exp}}$. The proposition follows by composing these operations.

Fix any inverse-exponential δ. For each operation $^*, ^\dagger, \mathsf{C}$, we will devise a circuit family C_s such that $\|C_s - \overline{V}_s\|_{op} \leq \delta(|s|)$, where $\overline{V}_s$ is an extension of $U_s^*, U_s^\dagger, \mathsf{C}U_s$, respectively.

- Conjugation: since $\{U_s\}_s \in \mathsf{UnitaryBQP}^a_{\mathsf{exp}}$, there is a Turing machine M such that $N(s)$ outputs a quantum circuit D_s such that $\|D_s - \overline{U}_s\|_{op} \leq \delta(|s|)/2$, where $\overline{U}_s$ is an a-qubit extension of U_s. Then observe that $\overline{U}_s^*$ is also an a-qubit extension of U_s^*, and $\|D_s^* - \overline{U}_s^*\|_{op} = \|D_s - \overline{U}_s\|_{op} \leq \delta(|s|)/2$. Thus, let $M(s)$ be the Turing machine that runs $N(s)$ to get the circuit D_s, and outputs D_s^*. Here, D_s^* is obtained by replacing each gate G in D_s with its conjugate G^*. Note that a universal gate set containing G does not necessarily contain G^*, but would instead need to be approximated. Fortunately, thanks to the Solovay-Kitaev theorem, we can approximate G^* to error $\delta(|s|)/2\ell$ using $\mathsf{polylog}(2\ell/\delta(|s|))$ gates (and known global phase), where ℓ is the size of D_s. This can also be done without any additional ancillas. Thus, we have that the overall error is at most $\delta(|s|)$, and since δ is at most exponential, the running time/circuit complexity only blows up by a polynomial factor.

- Inversion: This is basically the same as conjugation, except that we replace each gate G with its inverse $\mathsf{G}^\dagger$, and moreover reverse the order of the gates, since $(\mathsf{G}_0\mathsf{G}_1)^\dagger = \mathsf{G}_1^\dagger\mathsf{G}_0^\dagger$. Note that typically universal gate sets are considered to be closed under inverses, so if G is in the universal gate set, so is G^{-1}. However, [BGT21] shows that this assumption is not necessary.

- Controlling: this is essentially identical to conjugation. □

Remark 23. *Note that controlling has appeared informally many times in the literature, and a formal version of controlling was already considered in [KTP20]. However, their proof implicitly assumes a unitary U_s implemented exactly by a quantum circuit. Their proof also implicitly assumes that the circuit preserves global phase, though this is never stated.*

3.3 On our "closeness" metric

Here, we make the case for our closeness metric that pays attention to global phase, as opposed to that of [BEM+23], which ignores global phase. Concretely, we will see that no analog of Proposition 22 holds for the class as defined in [BEM+23]. Let $\mathsf{UnitaryBQP}'_{\mathsf{poly}}$ (resp. $\mathsf{UnitaryBQP}'_{\mathsf{exp}}$) denote the definition in [BEM+23], which is the same as $\mathsf{UnitaryBQP}_{\mathsf{poly}}$ (resp. $\mathsf{UnitaryBQP}_{\mathsf{exp}}$) except that the circuit C_s only has to be δ-close to U_s as quantum channels, which ignores global phase. Likewise define $\mathsf{UnitaryE}'_{\mathsf{poly}}$ and $\mathsf{UnitaryE}'_{\mathsf{exp}}$.

Proposition 24. *There exists a family $\{U_s\}_s \in \mathsf{UnitaryBQP}'_{\mathsf{exp}}$ where $\{\mathsf{C}U_s\}_s$ is not even in $\mathsf{UnitaryE}'_{\mathsf{poly}}$.*

Proof. Let U_s be the unitary $(-1)^{H(s)}\mathbf{I}$, where $H(s)$ interprets s as a Turing machine and outputs 1 if the machine halts on the empty input, 0 otherwise.

Ignoring global phase, U_s is just $\mathbf{I}$, which can be computed exactly trivially. Thus $\{U_s\}_s \in \mathsf{UnitaryBQP}'_{\mathsf{exp}}$. However, we now argue that $\{\mathsf{C}U_s\}_s \notin \mathsf{UnitaryE}'_{\mathsf{poly}}$. In fact, $\{\mathsf{C}U_s\}_s$ cannot be approximated by *any* computable algorithm, even to within an error as large as $1-\epsilon$ for any computable function ϵ.

Suppose to the contrary that $\{\mathsf{C}U_s\}_s \in \mathsf{UnitaryE}'_{\mathsf{poly}}$. Let δ be some desireable error, and let $M(s)$ be the Turing machine which outputs a circuit C_s that is δ-close to an extension of U_s. Consider the quantum algorithm $Q(s)$ which solves the Halting problem. It runs $M(s)$ to get C_s. Then it repeats the following a $\mathsf{poly}(1/\delta)$ number of times: it initializes the state $|+\rangle|0^n\rangle$, applies C_s, and then measures the first qubit in the $|+\rangle, |-\rangle$ basis.

Finally, $Q(s)$ outputs the majority of all the measurements.

If C_s were actually $\mathsf{C}U_s$, then $\mathsf{C}U_s|+\rangle|0^n\rangle = (|0\rangle + (-1)^{H(s)}|1\rangle)|0^n\rangle$, Measuring the first qubit in the $|+\rangle, |-\rangle$ basis reveals $H(s)$. Since C_s actually has error, we instead get a noisy version of $H(s)$. But taking the majority of a large number of samples gives $H(s)$ the result with overwhelming probability.

Thus we have a quantum algorithm which solves the Halting problem. As classical and quantum decidability are equivalent, this gives a classical algorithm for the Halting problem as well, which is impossible. Thus we have that $\{\mathsf{C}U_s\}_s \notin \mathsf{UnitaryE}'_{\mathsf{poly}}$. □

3.4 On Negligibly-Small Error

Here, we make the case for negligibly-small error, and maybe even exponentially-small error as opposed to polynomial error as in [BEM+23].

First, we observe that algorithmic results such as the Solovay-Kitaev theorem promise an exponentially-small error. We also point out that classical algorithms for sampling problems – such as sampling random numbers in an interval, random primes, (discrete) Gaussians, etc – are typically expected to have exponentially-small error.

Cryptographic applications. Next, we argue that negligibly-small error is crucial for cryptographic applications. While our discussion applies generally, for concreteness here we will focus on the case of pseudorandom unitaries [JLS18]. In [JLS18] and many of the works following it (e.g. [MPSY24, BM24, MH25]), a pseudorandom unitary is roughly defined as follows: it consists of a collection of unitaries $\{U_s\}_s$ such that

1. $\{U_s\}_s$ is efficiently computable by a quantum algorithm
2. No polynomial-time quantum query algorithm can distinguish using quantum queries U_x for a random x of length n from a Haar random unitary V of the same dimension

What it means to be "computable" by a quantum algorithm is not specified. One interpretation is that there is an algorithm which computes $\{U_s\}_s$ *exactly*. But this is likely too strong: for example, even implementing relatively simple quantum operations like the quantum Fourier transform (QFT) from a finite universal gate set seems to require approximations.[4] Moreover, there are even universal gate sets that cannot even simulate *classical* computation exactly.

Intuitively, however, it makes sense to ask "efficiently computable" $\{U_s\}_s$ are captured by $\{U_s\}_s \in \mathsf{UnitaryBQP}$, in whatever way the class $\mathsf{UnitaryBQP}$ is defined. We now see how the choice of error in defining $\mathsf{UnitaryBQP}$ affects pseudorandom unitaries by considering the cases where U_s is implemented to inverse-polynomial error vs negligibly-small error.

First consider inverse-polynomial error, say $1/p(|x|)$. Unfortunately, such error actually invalidates the security proof of the PRU construction. This is because what is shown is that, for any polynomial-time adversary A, A^{U_x} is indistinguishable from A^V (which we will denote $A^{U_x} \approx A^V$) where x is chosen randomly and V is Haar random. However, in the "real world", A does not necessary see the exact unitary U_x, but rather sees the unitary U'_x provided by the implementation.

[4] [MZ03] claims to give an "exact" QFT but the computational model is not specified precisely. In particular, as a subroutine they apply "exact" amplitude amplification [BHMT02] to an inexact algorithm. In general, exact amplitude amplification requires preparing states whose amplitudes depend in complex ways on the success probability of the underlying algorithm, and in general it appears unlikely that these states can be generated exactly using a finite universal gate set.

The hope would be to use that $A^{U'_x} \approx A^{U_x}$ to conclude that $A^{U'_x} \approx A^V$ by the triangle inequality. However, in the inverse-polynomial error regime, $A^{U'_x}$ is only inverse-poly close to A^{U_x} and hence A^V, which is not enough to conclude security as security requires negligible distinguishing advantage. Worse, we could have A make $q \gg p$ queries, in which case the error over all queries is actually ≈ 1. One could try moving to a better error p, but no matter what p is used there may exist an adversary with slightly-smaller distinguishing advantage, or that makes more than $1/p$ queries to achieve a large distinguishing advantage.

On the other hand, consider the case of negligibly-small error. Then we would indeed have that $A^{U'_x}$ is negligibly-close to A^{U_x}, for any polynomial number of queries, allowing us to conclude that $A^{U'_x} \approx A^{U_x} \approx A^V$, showing that the actual construction as implemented is secure.

While we focused on PRUs, the above can be applied to any crypographic primitive that utilizes unitaries, showing that it is sufficient to have the unitaries be in $\mathsf{UnitaryBQP}_{\mathsf{negl}}$.

Remark 25. *Note that [BEM+23] draw connections between unitary complexity and certain cryptographic notions. However, their connections consider whether the* adversary *for the cryptographic notion is in a complexity class, in contrast to our example above considering the* algorithms *of the cryptosystem being in the class. From their perspective, an inverse-poly error makes sense, as an adversary is typically considered successful even if it incurs such error. More generally, when we want an upper-bound (i.e., and algorithm), we would typically ask for at least negligibly-small error, and possibly exponentially-small error. However, when we ask for a lower-bound, we would typically want the lower-bound to apply even in the polynomial-error case. So whether a complexity class is more naturally defined using exponentially-small error or polynomial error would depend on whether that class is supposed to capture upper- or lower-bounds.*

4 Oracle Separations and Indifferentiability

Here, we consider a quantum version of indifferentiability, originally defined in the classical setting by [MRH04].

Definition 26. *Let $\{\mathcal{D}_\lambda\}_\lambda, \{\mathcal{E}_\lambda\}_\lambda$ be two (families of) distributions over unitary transformations, and $\{C_\lambda\}_\lambda$ a family of oracle-aided unitaries in $\mathsf{UnitaryBQP}_{\mathsf{negl}}$. Let $C_\lambda^{\mathcal{D}_\lambda}$ denote the distribution over C_λ^D where $D \leftarrow \mathcal{D}_\lambda$. We say $\{C_\lambda^{\mathcal{D}_\lambda}\}_\lambda$ is quantum indifferentiable from $\{\mathcal{E}_\lambda\}_\lambda$ if, for every QPT algorithm $\mathcal{A}$, there exists a quantum polynomial-time stateful simulator $\mathcal{S}$ and a negligible negl such that*

$$\|\Pr[\mathcal{A}^{C_\lambda^D, D}(1^\lambda) = 1 : D \leftarrow \mathcal{D}_\lambda] - \Pr[\mathcal{A}^{E, \mathcal{S}^E(1^\lambda)}(1^\lambda) = 1 : E \leftarrow \mathcal{E}_\lambda]\| \leq \mathsf{negl}(\lambda)$$

We will often drop the sub-script λ and also the curly braces and just write $C, \mathcal{D}, \mathcal{E}$. The case C^D, D is called the "real" world, while the case E, S^E is called the "ideal" world.

A single-stage game is an interactive game between a single adversary and challenger. The adversary is allows arbitrary local computation and storage between its messages, except for being polynomial-time. The game itself is also polynomial-time. The following is a trivial adaptation of the analogous classical result of [MRH04].

Theorem 27. *Suppose a cryptographic primitive P has security specified by a single-stage game. Suppose that $R^{\mathcal{E}}$ is a secure realization of P relative to the oracle distribution $\mathcal{E}$, and $C^{\mathcal{D}}$ is quantum indifferentiable from $\mathcal{E}$. Then $R^{C^{\mathcal{D}}}$ is a secure realization of P relative to the oracle $\mathcal{D}$.*

Proof. We need to prove that $R^{C^{\mathcal{D}}}$ is secure. Toward that end, we imagine a hypothetical adversary $\mathcal{A}$ breaking $R^{C^{\mathcal{D}}}$ given oracle access to $\mathcal{D}$. By viewing the game and adversary as a single entity $\mathcal{B}$ making queries to $C^{\mathcal{D}}$ (coming from R) and $\mathcal{D}$, indifferentiability tells us that $\mathcal{B}$ and hence $\mathcal{A}$ still breaks $R^{\mathcal{E}}$ when given access to $\mathcal{S}^{\mathcal{E}}$. But then we combine $\mathcal{A}$ and $\mathcal{S}$ into a third adversary $\mathcal{A}'$ which breaks $R^{\mathcal{E}}$ by making queries to $\mathcal{E}$. This breaks the assumption that $R^{\mathcal{E}}$ was secure relative to $\mathcal{E}$. □

A simple converse of this theorem is that, in order for a construction $C^{\mathcal{D}}$ to securely realize $\mathcal{E}$ in *all* single-stage games, it must be indifferentiable.

Theorem 27 allows for translating separations between different oracles as follows. Suppose P exists and Q does not relative to $\mathcal{D}$. Moreover, suppose that $\mathcal{D}$ and $\mathcal{E}$ are *equivalent*, in the sense that there are indifferentiable constructions of $\mathcal{E}$ from $\mathcal{D}$ and vice versa. Then we conclude that P exists relative to $\mathcal{E}$ as well using Theorem 27. Moreover, Q cannot exist relative to $\mathcal{E}$, lest Theorem 27 implies it also exists relative to $\mathcal{D}$, a contradiction.

Witness Classes and Perfect Indifferentiability. Unfortunately, the above is actually insufficient for lifting complexity classes that involve witnesses. The reason is that complexity classes involve witnesses that are inefficiently computed given the entire description of the oracle. As such, the "game" modeling a complexity class is not efficient. Nevertheless, indifferentiability is a *necessary* requirement for generically translating a complexity separation from one-oracle to another.

4.1 On Building Quantum Oracles from Classical Oracles

Definition 28. *Let $\mathcal{U}$ be a distribution on unitaries U, and M a mapping between unitary transformations. We say that $\mathcal{U}$ is* M-composable *if there exists negligible functions ϵ, δ and algorithm A such that $\Pr_{U \leftarrow \mathcal{U}}[\|A^U - M(U)\|_{\mathsf{avg}} > \epsilon] < \delta$.*

Theorem 29. *Let $\mathcal{O}$ be a distribution over classical oracles, $\mathcal{U}$ be a distribution over unitaries, and C an oracle-aided quantum circuit making queries to $\mathcal{O}$. Suppose that $C^{\mathcal{O}}$ is indifferentiable from $\mathcal{U}$. Then $\mathcal{U}$ must be M-composable for $M \in \{U \to U^\dagger, U \to U^*, U \to U^T\}$.*

Note that Theorem 29 does *not* apply to controlling.

Proof. Suppose that $C^{\mathcal{O}}$ is indifferentiable from $\mathcal{U}$. This means for any potential distinguisher D, there is a simulator S such that $\Pr[D^{C^{\mathcal{O}},\mathcal{O}}() = 1]$ is negligibly-close to $\Pr[D^{\mathcal{U},S^{\mathcal{U}}}() = 1]$.

We first handle the U^* case. Consider the circuit $(C^*)^O$, which conjugates every gate except query gates. In the real world, O is a classical oracle, which means that the gate representing O is its own conjugate. Hence $(C^*)^O = (C^O)^*$.

By Lemma 13, there is an algorithm $\mathcal{A}^{U,V}$ making queries to U, V and outputs 1 with probability equal to $1 - \Theta(\|V - U^*\|_{avg})$. Consider running the algorithm $\mathcal{A}^{U,(C^*)^O}$. In the real world where $U = C^O$, this is $\mathcal{A}^{C^O,(C^O)^*}$, which outputs 1 with probability 1. Therefore, there is a simulator S such that in the ideal world, $\mathcal{A}^{U,(C^*)^{S^U}}$ outputs 1 with probability $1-\epsilon$ for some negligible function ϵ. This means with probability at least $1-\sqrt{\epsilon}$ over the choice of U, $\Pr[\mathcal{A}^{U,(C^*)^{S^U}}() = 1] \geq 1-\sqrt{\epsilon}$ where the probability now is just over the randomness of $\mathcal{A}$. Thus, in this event $\|U^* - (C^*)^{S^U}\|_{avg} \leq O(\sqrt{\epsilon})$.

The construction for $U^\dagger$ is analogous, but uses the algorithm from Lemma 12. U^T is simply a composition of the two. □

Remark 30. *The above does not work for controlling. This is because the simulator may introduce a global phase, and there is no way to account for this in the analysis.*

5 Breaking the Indifferentiability of [MH25]

In this section, we show that the very recent construction of a pseudorandom unitary by [MH25] cannot be indifferentiable from a random unitary. Their construction has the form $\mathsf{U} = \mathsf{C}_0\mathsf{P}\mathsf{F}\mathsf{C}_1$, where:

- $\mathsf{C}_0, \mathsf{C}_1$ are random Cliffords in $\mathbf{C}_n$ (or $\mathbf{C}'_n$).
- P is a random permutation matrix
- F is a diagonal matrix where each diagonal entry is chosen randomly from the q-th roots of unity, for some $q \geq 2$.

We will call this the CPFC construction, which is a generalization of the PFC construction from [MPSY24]. [MH25] show that this construction looks like a random unitary give only query access to U (and U^{-1}). This allows it to give a pseudorandom unitary, by replacing P, F with appropriate pseudorandom objects. This captures settings where the unitary is being computed by a third party, rather than users themselves.

We will now consider whether this can be used to give *publicly accessible* random-looking unitary. In other words, we want CPFC to look like a random unitary, even if it is publicly evaluatable, meaning users compute it for themselves using the underlying $\mathsf{P}, \mathsf{F}, \mathsf{C}_0, \mathsf{C}_1$. In this setting, the correct notion of security indifferentiability, since the attacker will have access to $\mathsf{P}, \mathsf{F}, \mathsf{C}_0, \mathsf{C}_1$ themselves in order to compute U.

Note that while F is not a classical unitary, it can be implemented by a random function from $\{0,1\}^n \to \mathbb{Z}_q$ by putting the output of the function in the phase. The Cliffords $\mathsf{C}_0, \mathsf{C}_1$ are also given by an oracle which simply outputs their description.

Our main result, unfortunately, shows that this construction cannot possibly be indifferentiable from a random unitary.

Theorem 31. *Regardless of the distribution over $\mathsf{P}, \mathsf{F}, \mathsf{C}_0, \mathsf{C}_1$, the CPFC construction cannot be indifferentiable from a random unitary.*

Proof. Suppose that the CPFC construction is indifferentiable. Then in the "ideal" world, there is a simulator for the oracles $\mathsf{P}, \mathsf{F}, \mathsf{C}_0, \mathsf{C}_1$ which itself makes queries to a Haar random U. Since the adversary may as well query to obtain $\mathsf{C}_0, \mathsf{C}_1$, we can assume these are given as classical strings. Then Lemma 19 says that we can consider $\mathsf{C}_0, \mathsf{C}_1$ to be sampled independently of U. Let $\mathsf{V} = \mathsf{C}_0^\dagger \mathsf{U} \mathsf{C}_1^\dagger$. Then since U is Haar random and $\mathsf{C}_0, \mathsf{C}_1$ are independent of U, this means V is Haar random.

Also, per Theorem 29, we can assume that the adversary and simulator have access to both U and U^*, which implies that the adversary has access to V, V^*. Technically they only have access to a circuit that is close to U^* in an average-case sense, but this is sufficient for our proof. We therefore ignore this distinction and assume for simplicity that access to U^* itself is provided.

Our main observation is that, in the "real" world, $\mathsf{P}^* = \mathsf{P}$ and $\mathsf{F}^* = \mathsf{F}^\dagger$. Thus, if U is sampled from the CPFC construction and we define $\mathsf{V} = \mathsf{C}_0^\dagger \mathsf{U} \mathsf{C}_1^\dagger = \mathsf{P}\mathsf{F}$, then $\mathsf{V}^* = \mathsf{P}\mathsf{F}^\dagger$. In particular, this means that if we apply $\mathsf{V} \otimes \mathsf{V}^*$ to a basis state $|x\rangle^{\otimes 2}$, the phases cancel out and we are left with

$|y\rangle^{\otimes 2}$ for $|y\rangle = \mathsf{P}|x\rangle$. We can then test for this property by comparing the two output registers, and this property will almost certainly *not* hold for a general random unitary V. We now give the attack in more detail.

Prepare phased EPR state. Choose a random pairwise independen function $f : \{0,1\}^n \to \{0,1\}$, and prepare $|\psi_f^0\rangle = \frac{1}{2^{n/2}} \sum_x (-1)^f(x)|x,x\rangle$.

Apply $\mathsf{V} \otimes \mathsf{V}^*$. Now we compute $|\psi_f^1\rangle = \mathsf{V} \otimes \mathsf{V}^*|\psi_f^0\rangle$. This equals:

$$|\psi_f^1\rangle = \frac{1}{2^{n/2}} \sum_{x,y,y'} V_{y,x} V^*_{y',x} (-1)^{f(x)} |y,y'\rangle$$

where $V_{y,x}$ are the entries of V.

Project onto $y = y'$. Now we apply the projection $S = \sum_y |y,y\rangle\langle y,y|$, obtaining the unnormalized state

$$|\psi_f^2\rangle = S|\psi_f^1\rangle = \frac{1}{2^{n/2}} \sum_{x,y} |V_{y,x}|^2 (-1)^{f(x)} |y,y\rangle$$

The amplitude squared of $\langle\psi_f^2|\psi_f^2\rangle$ is exactly the probability the projection accepts. Let this probability be denoted by p_f. Then we have that:

$$\begin{aligned} p_f &= \frac{1}{2^n} \sum_y \left| \sum_x |V_{y,x}|^2 (-1)^{f(x)} \right|^2 \\ &= \frac{1}{2^n} \sum_{y,x,x'} |V_{y,x}|^2 \cdot |V_{y,x'}|^2 (-1)^{f(x)+f(x')} \end{aligned}$$

Let $p = \mathbb{E}_f[p_f]$ over the choice of f. Observe that since f is pairwise independent, then $\mathbb{E}_f[(-1)^{f(x)+f(x')}]$ is 1 if and only if $x = x'$, and otherwise it is 0. Therefore we have that

$$p = \frac{1}{2^n} \sum_{x,y} |V_{y,x}|^4$$

We see for real world matrices $\mathsf{V} = \mathsf{PF}$ that $p = 1$ since the sum is over the 2^n non-zero entries, each of which have norm 1. On the other hand, for Haar random V, p will in general be very small. In particular, for a Haar random matrix V, it is not hard to show that $\mathbb{E}_{\mathsf{V}}[p] = \frac{2}{2^n - 1}$, which is negligible. Thus, our attack distinguishes the two cases. □

6 On Quantum Black Box Reductions

Definition 32. *Let $n(\lambda) > \lambda$ be a function on $\mathbb{Z}$. A one-time pseudorandom state (1-PRS) is a family of states $\{|\psi_k\rangle\}_k$ such that:*

- *There exists a family of unitaries $\{U_k\}_k \in \mathsf{UnitaryBQP}'_{\mathsf{negl}}$ such that for all $k \in \{0,1\}^\lambda$, $U_k|0^{n(\lambda)}\rangle = |\psi_k\rangle$. In particular, if $k \in \{0,1\}^\lambda$, then $|\psi_k\rangle$ is a state over $n(\lambda)$ qubits.*

- *For any QPT adversary $\mathcal{A}$, there exists a negligible* negl *such that for all* $\lambda \in \mathbb{Z}$, $\|\Pr[\mathcal{A}(|\psi_k\rangle) = 1 : k \leftarrow \{0,1\}^\lambda] - \Pr[\mathcal{A}(|x\rangle) = 1 : x \leftarrow \{0,1\}^{n(\lambda)}]\| \leq \mathsf{negl}(\lambda)$

Definition 33. *A PRS is called* junk-free *if* $U_k \in \mathsf{UnitaryBQP}_{\mathsf{negl}}$.

We are not aware of any PRS in the literature that is not junk-free. In known constructions, it seems any side-information can always be uncomputed.

Definition 34. *A PRS is called* $*$*-anti-correlated if* $\|\mathbb{E}_k |\psi_k\rangle\langle\psi_k^*|\|$ *is negligible in* $|k|$.

$*$-anti-correlation, intuitively, means that $|\psi^*\rangle$ is fairly un-related to $|\psi\rangle$. For example, if $|\psi^*\rangle = |\psi\rangle$, then the expectation is actually just the totally mixed state, which has trace norm 1.

Note that $*$-anti-correlation is a very strong statistical property, and we do not expect it to hold for many or even most PRS constructions. However, it is straightforward to devise PRSs that have this property. For example, by slightly increasing the key length, we can apply to any PRS a global phase $e^{i\theta}$ where θ is determined from bits of the key that are independent of the original PRS. then $|\psi_k\rangle\langle\psi_k^*|$ picks up a phase $e^{i2\theta}$. As long as this phase averages to 0 – for example, if θ is uniform in $\{0, \pi/2\}$ – then $\|\mathbb{E}_k |\psi_k\rangle\langle\psi_k^*|\| = 0$.

Construction 35. *Let* $\{|\psi_k\rangle\}_k$ *be a 1-PRS. Then define the new family* $\{|\psi'_k\rangle\}_k$ *where*

$$|\psi'_k\rangle = \frac{1}{\sqrt{2}}|0\rangle|\psi_k\rangle + \frac{1}{\sqrt{2}}|1\rangle|\psi_k^*\rangle$$

Theorem 36. *If* $\{|\psi_k\rangle\}_k$ *is a 1-PRS that is junk-free and* $*$*-anti-correlated, then* $\{|\psi'_k\rangle\}_k$ *in Construction 35 is also a 1-PRS.*

Proof. First, we need to show that $|\psi'_k\rangle$ can be implemented efficiently. This is straightforward. Given a circuit for a unitary $\{U_k\}_k \in \mathsf{UnitaryBQP}_{\mathsf{negl}}$ on n qubits which computes $|\psi_k\rangle$, we define the unitary $U' = (\mathsf{C}U_k^*)\mathsf{X}_1(\mathsf{C}U_k)\mathsf{X}_1\mathsf{H}_1$, where the subscripts indicate applying the gate to the first wire, which is also the control for $\mathsf{C}U_k, \mathsf{C}U_k^*$. By Proposition 22, the unitary U' is in $\mathsf{UnitaryBQP}_{\mathsf{negl}}$.

Next, we need to argue security. Consider an adversary $\mathcal{A}'$ for $\{|\psi'_k\rangle\}_k$ and define $\epsilon(\lambda) = \|\Pr[\mathcal{A}'(|\psi'_k\rangle) = 1 : k \leftarrow \{0,1\}^\lambda] - \Pr[\mathcal{A}'(|x\rangle) = 1 : x \leftarrow \{0,1\}^{n(\lambda)+1}]\|$. Our goal is to show that ϵ is negligible.

Toward that end, we define

$$\begin{aligned}
p_0 &= \Pr[\mathcal{A}'(|\psi'_k\rangle) = 1 : k \leftarrow \{0,1\}^\lambda] \\
p_1 &= \Pr\left[\mathcal{A}'(|b\rangle, |\phi_b\rangle) = 1 : \begin{smallmatrix} b\leftarrow\{0,1\}, k\leftarrow\{0,1\}^\lambda \\ |\phi_0\rangle\leftarrow|\psi_k\rangle \\ |\phi_1\rangle\leftarrow|\psi_k^*\rangle \end{smallmatrix}\right] \\
p_2 &= \Pr\left[\mathcal{A}'(|b\rangle, |\phi_b\rangle) = 1 : \begin{smallmatrix} b\leftarrow\{0,1\}, k\leftarrow\{0,1\}^\lambda, x\leftarrow\{0,1\}^{n(\lambda)} \\ |\phi_0\rangle\leftarrow|x\rangle \\ |\phi_1\rangle\leftarrow|\psi_k^*\rangle \end{smallmatrix}\right] \\
p_3 &= \Pr\left[\mathcal{A}'(|b\rangle, |x\rangle) = 1 : b \leftarrow \{0,1\}, x \leftarrow \{0,1\}^{n(\lambda)}\right]
\end{aligned}$$

We now argue that $\epsilon = |p_3 - p_0|$ is small by showing that $|p_i - p_{i-1}|$ is small for each i and using the triangle inequality.

Claim 1. *If* $\{|\psi_k\rangle\}_k$ $*$*-anti-correlated, then* $|p_0 - p_1|$ *is negligible in* λ.

Proof. The mixed state $\mathcal{A}'$ sees in p_0 is $\rho = \mathbb{E}_k[|\psi'\rangle\langle\psi'|]$, whereas the state seen in p_1 is the result of measuring the first qubit of ρ. Call this ρ'. Then we have that

$$\rho - \rho' = |0\rangle\rangle 1| \otimes |\psi_k\rangle\rangle\psi_k^*| + |1\rangle\rangle 0| \otimes |\psi_k^*\rangle\rangle\psi_k|$$

Then we have that $|p_0 - p_1| \leq \|\rho - \rho'\| = \|\mathbb{E}_k|0\rangle\rangle 1| \otimes |\psi_k\rangle\rangle\psi_k^*|\| + |\mathbb{E}_k 1\rangle\rangle 0| \otimes |\psi_k^*\rangle\rangle\psi_k|\| = 2\|\mathbb{E}_k|0\rangle\rangle 1| \otimes |\psi_k\rangle\rangle\psi_k^*|\| = 2\|\mathbb{E}_k|\psi_k\rangle\rangle\psi_k^*|\| \leq 2\mathsf{negl}(\lambda)$, where the inequality is exactly the definition of $*$-anti-correlation. □

Claim 2. *If $\{|\psi_k\rangle\}_k$ is a secure 1-PRS, then $|p_1 - p_2|$ is negligible in λ.*

Proof. Let $\mathcal{A}(|\phi\rangle)$ be the following adversary for $\{|\psi_k\rangle\}_k$:

- Choose a random b.
- If $b = 1$, choose a random k' and run $\mathcal{A}'(|1\rangle|\psi_{k'}^*\rangle)$
- If $b = 0$, run $\mathcal{A}'(|0\rangle|\phi\rangle)$
- Output whatever $\mathcal{A}$ outputs

Observe that if $|\phi\rangle$ is a random $|x\rangle$, then the probability $\mathcal{A}$ outputs 1 is exactly p_2. Likewise, if $|\phi\rangle$ is $|\psi_k\rangle$ for a random k, then the probability $\mathcal{A}$ outputs 1 is exactly p_1.[5] Thus, by the assumed 1-PRS security of $\{|\psi_k\rangle\}_k$, we must have that $|p_2 - p_1|$ is negligible. □

Claim 3. *If $\{|\psi_k\rangle\}_k$ is a secure 1-PRS, then $|p_2 - p_3|$ is negligible in λ.*

Proof. Let $\mathcal{A}(|\phi\rangle)$ be the following adversary for $\{|\psi_k\rangle\}_k$:

- Choose a random b.
- If $b = 0$, choose a random x and run $(\mathcal{A}')^*(|1\rangle|x\rangle)$
- If $b = 1$, run $(\mathcal{A}')^*(|0\rangle|\phi\rangle)$
- Output whatever $\mathcal{A}$ outputs

Observe that by conjugating everything in the definitions of p_2, p_3, an equivalent expression for p_2, p_3 is the following:

$$p_2 = \Pr\left[(\mathcal{A}')^*(|b\rangle, |\phi_b\rangle) = 1 : \begin{matrix} b\leftarrow\{0,1\}, k\leftarrow\{0,1\}^\lambda, x\leftarrow\{0,1\}^{n(\lambda)} \\ |\phi_0\rangle\leftarrow|x\rangle \\ |\phi_1\rangle\leftarrow|\psi_k\rangle \end{matrix}\right]$$
$$p_3 = \Pr\left[(\mathcal{A}')^*(|b\rangle, |x\rangle) = 1 : b \leftarrow \{0,1\}, x \leftarrow \{0,1\}^{n(\lambda)}\right]$$

Then we see that if $|\phi\rangle$ is given $|x\rangle$ for a random x, the probability $\mathcal{A}$ outputs 1 is exactly p_3. Likewise if $|\phi\rangle = |\psi_k\rangle$ for a random k, the probability $\mathcal{A}$ outputs 1 is exactly p_2. Thus, by the assumed 1-PRS security of $\{|\psi_k\rangle\}_k$, we must have that $|p_3 - p_2|$ is negligible. □

Thus, p_3 is negligibly close to p_0, showing that $\{|\psi_k'\rangle\}_k$ is a 1-PRS and proving Theorem 36. □

[5] Note that the definition of p_1 has $|\phi_0\rangle$ and $|\phi_1\rangle$ both use the same k, whereas this simulation sets $|\phi_1\rangle = |\psi_{k'}^*\rangle$ for an independent k'. However, since the view of $\mathcal{A}'$ only depends on at most one of $|\phi_0\rangle, |\phi_1\rangle$, the cases where the index k is the same or independent are identical.

6.1 Construction 35 requires conjugating the adversary

Here, we show that, relative to an oracle, there is no black box reduction proving the security of Construction 35 that does not conjugate the adversary.

Definition 37. *Let O be an oracle. A black-box reduction proving the security of Construction 35 relative to O consists of the following. For every purported 1-PRS family $\{|\psi_k^O\rangle\}_k$ relative to O, and for any purported adversary $\mathcal{A}$ for the 1-PRS $\{|(\psi'_k)^O\rangle\}_k$ from construction 1, there exists a polynomial-time reduction R which makes queries to $\mathcal{A}$ such that the following is true: if $\{|\psi_k^O\rangle\}_k$ is a secure PRS and $\mathcal{A}$ is a possibly inefficient algorithm which breaks $\{|(\psi'_k)^O\rangle\}_k$, then $R^{\mathcal{A}}$ breaks $\{|\psi_k^O\rangle\}_k$.*

Definition 38. *An ϵ-net is a family of states $\{|\psi_k\}_k$ such that (1) for every k, there exists a distribution D_k with support on states $|\phi\rangle$ such that $||\phi\rangle - |\psi\rangle|_2 \leq \epsilon$, and (2) the mixture of D_k for random k is Haar random.*

Theorem 39. *Relative to a random oracle O, there is no conjugate-free black box reduction proving the security of Construction 35.*

Proof. Let D^f be a unitary in $\mathsf{UnitaryBQP}_{\mathsf{negl}}$ which maps $|0^n\rangle$ into a quantum state $|\phi^f\rangle$ on n-qubits, such that if f is a random function then $|\phi^f\rangle$ is an ϵ-net, for an exponentially-small ϵ. Such states were constructed e.g. in [LQS+24].

Given an oracle O, consider the PRS with seed $(s,t) \in \{0,1\}^m \times \{0,1\}$ defined as $|\psi_{s,t}\rangle = i^t|\phi^{O(s,\cdot)}\rangle$

Secure PRS. This follows a standard argument. We first replace $|\psi_{s,t}\rangle$ with $i^t|\phi^f\rangle$ for a random function f independent of the oracle O. This change is indistinguishable via a straightforward application of Lemma 15. Then we replace $|\phi^f\rangle$ with a Haar-random state, which is indistinguishable by the ϵ-net property.

***-anti-correlation.** Observe that

$$\mathbb{E}_{s,t}[|\psi_{s,t}\rangle\langle\psi^*_{s,t}|] = \mathbb{E}_{s,t}[i^t|\phi^{O(s,\cdot)}\rangle\langle(\phi^{O(s,\cdot)})^*|i^t] = \mathbb{E}_{s,t}[(-1)^t|\phi^{O(s,\cdot)}\rangle\langle(\phi^{O(s,\cdot)})^*|]$$

Factor out the expectation $\mathbb{E}_t[(-1)^t]$, which is 0.

No conjugate-free black box reduction. Apply Construction 35 to our PRS. Thus, we obtain the new PRS $|\psi'_{s,t}\rangle = \frac{1}{\sqrt{2}}i^t|0\rangle|\phi^{O(s,\cdot)}\rangle + \frac{1}{\sqrt{2}}(-i)^t|0\rangle|(\phi^{O(s,\cdot)})^*\rangle$.

Now consider the following exponential-time adversary $\mathcal{A}$: let P be the projection onto the span of $\{|1\rangle|(\phi^{O(s,\cdot)})^*\rangle\}_s$ for all s. Then $\mathcal{A}$ projects its input state $|\tau\rangle$ onto P; if this projection accepts is outputs 1, and otherwise it rejects.

Observe that for our PRS, $\mathcal{A}$ outputs 1 with probability at least $1/2$, whereas for Haar random states it outputs 1 with probability equal to the ratio of the dimension of the projection to the overall space, which is at most $2^{-(n-s)} \ll 1/2$. Thus $\mathcal{A}$ is a valid adversary.

Now consider a supposed conjugate-free reduction $\mathcal{R}$ breaking $|\psi_{s,t}\rangle$ given queries to $\mathcal{A}$. Let q be the number of queries $\mathcal{R}$ makes to O (not including queries made by $\mathcal{A}$), which is a polynomial. We prove that $\mathcal{R}$ must fail to distinguish $|\psi_{s,t}\rangle$ from random. We do so through a sequence of hybrids.

Hybrid 0: Run $\mathcal{R}^{O,\mathcal{A}}(|\phi\rangle)$ for a Haar random state $|\phi\rangle$. Let p_0 be the probability $\mathcal{R}$ outputs 1.

Hybrid 1: Here, we choose a random $s^* \in \{0,1\}^m$, and we "puncture" the oracle O at all prefixes s^*, setting $O(s^*, x) = 0$ for all x. Call this oracle O'. Let p_1 be the probability $\mathcal{R}^{\mathcal{A},O'}$ outputs 1.

In Hybrid 0, let w_0 be the expected total query weight of all queries R makes to O (not including queries made by $\mathcal{A}$) on points with prefix s^*. Since Hybrid 0 is independent of s^*, s^* is random in $\{0,1\}^m$, and the total query weight is q, w_0 is exactly $q \times 2^{-m}$. Let w_1 be the expected total query weight in Hybrid 1.

We therefore invoke Lemma 15 to conclude that $|p_0 - p_1| \leq O(\sqrt{q^2 2^{-m}}) = O(q2^{-m/2})$. Let $\mathcal{B}$ be the algorithm which runs $\mathcal{R}$ to a random query to O or O', measures the query, and outputs 1 if and only if the measured prefix s is identical to s^*. The probability $\mathcal{B}$ outputs 1 is exactly the query weight of s^*, divided by the number of queries. Thus, by invoking Lemma 15 on $\mathcal{B}$, we see that $|w_0/q - w_1/q| \leq O(q2^{-m/2})$. In particular, we have that $w_1 \leq O(q \times 2^{-m} + q^2 2^{-m/2}) \leq O(q^2 2^{-m/2})$.

Hybrid 2: Now we give $\mathcal{R}$ a different adversary $\mathcal{A}'$ that does not make any queries of the form $O(s^*, x)$. Instead, $\mathcal{A}'$ will be initialized with ℓ copies of the state $|\tau^*\rangle := |\phi^{O(s^*,\cdot)})^*\rangle$. If first projects onto the span of $\{|1\rangle|(\phi^{O(s,\cdot)})^*\rangle\}_{s \neq s^*}$, is the same as $\mathcal{A}$ except that it does not include the projection onto $|1\rangle|(\phi^{O(s^*,\cdot)})^*\rangle$. Therefore, if the first projection fails, $\mathcal{A}'$ simulates the projection onto $|\phi^{O(s^*,\cdot)})^*\rangle$ using the simulator $\mathcal{S}$ from Lemma 16.

We will let $\ell = \min((q/\epsilon)^{2/3}, (q2^n)^{2/5})$, which will minimize certain terms in the proceeding derivations.

Let p_2 be the probability $\mathcal{R}^{\mathcal{A}',O'}(|\phi\rangle)$ outputs 1. Let w_2 be the expected query magnitude of s^*. Invoking Lemma 16 on $\mathcal{A}$, we have that $|p_2 - p_1| \leq O(q/\sqrt{\ell})$, and invoking it on $\mathcal{B}$ we have $|w_2/q - w_1/q| \leq O(q/\sqrt{\ell})$. In particular, this means that $|p_2 - p_0| \leq O(q2^{-m/2} + q/\sqrt{\ell})$ and $w_2 \leq O(q^2 2^{-m/2} + q^2/\sqrt{\ell})$.

Hybrid 3. Next, we replace the state $|\tau^*\rangle$ given to $\mathcal{A}'$ with a Haar random state. Let p_3 be the probability $\mathcal{R}$ outputs 1, and w_3 be the query weight on s^*. Observe that in Hybrid 2, no queries are made to $O(s^*, x)$, and thus these values are independent of the view of $\mathcal{R}$ except through $|\tau^*\rangle$. We can therefore invoke the ϵ-net property to conclude that each copy of $|\tau^*\rangle$ in Hybrid 2 is ϵ-close to a Haar random state. Thus, by applying Lemma 14 to both $\mathcal{A}$ and $\mathcal{B}$, we have that $|p_3 - p_2|, |w_3/q - w_2/q| \leq O(\ell\epsilon)$. In particular, this means $|p_3 - p_0| \leq O(q2^{-m/2} + q/\sqrt{\ell} + \ell\epsilon)$ and $w_3 \leq O(q^2 2^{-m/2} + q^2/\sqrt{\ell} + q\ell\epsilon)$.

Hybrid 4. Now we switch to $|\tau^*\rangle = |\phi^*\rangle$. Define p_4, w_4 analogously to the previous hybrids. We invoke Lemma 17 to conclude that $|p_4 - p_3|, |w_4/q - w_3/q| \leq O(\ell^3 \times 2^{-2n})$. In particular, this means $|p_4 - p_0| \leq O(q2^{-m/2} + q/\sqrt{\ell} + \ell\epsilon + \ell^2 2^{-n})$, which using our choice of ℓ give $|p_4 - p_0| \leq O(q2^{-m/2} + q^{4/5}2^{-n/5} + q^{2/3}\epsilon^{1/3})$. Likewise, $w_4 \leq O(q^2 2^{-m/2} + q^{9/5}2^{-n/5} + q^{5/3}\epsilon^{1/3})$.

Hybrid 5. Now we switch to $|\phi\rangle = |\tau\rangle = |\phi^{O(s^*,\cdot)}\rangle$ and $|\phi^*\rangle = |\tau^*\rangle = |(\phi^{O(s^*,\cdot)})^*\rangle$. This move is essentially identical to the move between Hybrid 2 and Hybrid 3, but in reverse. We have $|p_5 - p_0| \leq O(q2^{-m/2} + q^{4/5}2^{-n/5} + q^{2/3}\epsilon^{1/3})$, $w_5 \leq O(q^2 2^{-m/2} + q^{9/5}2^{-n/5} + q^{5/3}\epsilon^{1/3})$.

Hybrid 6. Now we switch back to giving $\mathcal{R}$ access to the adversary $\mathcal{A}$ instead of $\mathcal{A}'$. This is essentially identical to the move betwen Hybrid 1 and Hybrid 2 but in reverse. We have $|p_6 - p_0| \leq O(q2^{-m/2} + q^{4/5}2^{-n/5} + q^{2/3}\epsilon^{1/3})$, $w_6 \leq O(q^2 2^{-m/2} + q^{9/5}2^{-n/5} + q^{5/3}\epsilon^{1/3})$.

Hybrid 7. Finally, we switch back to giving $\mathcal{R}$ the unpunctured oracle $\mathcal{O}$. This is analogous to the transition between between Hybrid 0 and Hybrid 1, except that we have to use the bound on w_6 rather than the trivial bound on w_0. The result is that $|p_7 - p_6| \leq O(\sqrt{qw_6})$, which we can bound as $O(q^{3/2}2^{-m/4} + q^{14/10}2^{-n/10} + q^{4/3}\epsilon^{1/6})$. This bound dominates all the other terms, and so by slightly increasing the constant hidden by Big-Oh, this is also a bound on $|p_7 - p_0|$.

Hybrid 7 is the case where $\mathcal{R}^{O,\mathcal{A}}(|\phi^{O(s^*,\cdot)}\rangle)$, which is the PRS. Thus, we have proved that no polynomial-query $\mathcal{R}$ can distinguish the PRS from Haar random. □

7 Homomorphisms on Unitaries

Here, we we discuss possible refinements of our model for public quantum oracles. More precisely, we ask: what makes computing $U^\dagger$, U^*, or controlled-U possible? Could these be part of a larger class of operations on U that need to be modeled in a quantum oracle?

Our key observation is that the maps $U \mapsto \mathsf{C}U$, $U \mapsto U^*$ are *homomorphisms* between unitary groups. Moreover they are "nicely behaved" – for example their entries are continuous functions of the entries of U (we will call these "continuous" homomorphisms). Given a circuit description of U, we can evaluate these homomorphisms on U by evaluating the homomorphism on each gate. In turn, the homomorphism on each gate simply gives another finite gate set, and any efficient circuit over any finite gate set can be efficiently simulated by any universal gate set using the Solovay-Kitaev theorem.

$U \mapsto U^\dagger$ is not a homomorphism, but an *anti*-homomorphism (like a homomorphism but reversing the order of multiplication). Similarly $U \mapsto U^T$ is an anti-homomorphism. We can evaluate such an anti-homomorphism by by evaluating the anti-homomorphism on each gate, and then reversing the order of gates. Note that every anti-homomorphism is just a homomorphism followed by conjugate transpose. Hence, we will just focus on understanding the case of homomorphisms.

One may expect from the preceding discussion that *any* homomorphism could be computed, as long as the homomorphism applied to each gate in the starting gate set gave rise to an efficiently computable unitary. This turns out to be true in a certain restricted sense, but as we will see, this does not give rise to any additional "power", and it suffices to consider only access to $\mathsf{C}U, \mathsf{C}U^*$.

Determinant. As an illustrative example, we consider the case of the determinant, which is not a member of $U, U^\dagger, U^*, U^T$, and in general cannot be computed efficiently given black box access to these oracles. If we could compute the determinant of a general unitary U given a quantum circuit implementing it, this would have profound implications. In particular, for a binary-output classical function f, consider the unitary $U_f|x, b\rangle = |x, b \oplus f(x)\rangle$. Then the determinant of U_f is just $(-1)^p$ where p is the number of accepting inputs of f, or equivalent p is the parity of f since $(-1)^p$ loses all information about the higher-order bits. If we could compute the parity of f for any efficiently computable f, then a result of [Tod91] tells us that $\mathsf{PH} \subseteq \mathsf{BPP}$, which is considered highly unlikely.

But det is a homomorphism between unitary groups, since we have the multiplicative identity $\det(UV) = \det(U)\det(V)$. So why can't we compute $\det(U)$ by computing the determinant of each gate and multiplying?

First, we must observe that if U has a gate G acting on m qubits, but U acts on n qubits, then $\mathbb{G}$ is really acting on all n qubits as $\mathsf{G} \otimes \mathbf{I}_{n-m}$ (for an appropriate permutation of the n qubits), where $\mathbf{I}_{n-m}$ acts on $n-m$ qubits. When we evaluate a homomorphism gate-by-gate, we must apply the homomorphism to this overall unitary operation on n qubits. This gives rise to the following property of a homomorphism:

Definition 40. *A homomorphism Φ is called* extendable *if, for any fixed unitary U acting on a constant m qubits, there is a polynomial-time (in n) algorithm which evaluates the unitary $\Phi(U \otimes \mathbf{I}_{n-m})$ for any ordering of the n qubits.*

Taking conjugates and controlling are both easily seen to be extendable, as $(\mathsf{G} \otimes \mathbf{I}_{n-m})^* = \mathsf{G}^* \otimes \mathbf{I}$ and $\mathsf{C}(\mathsf{G} \otimes \mathbf{I}_{n-m}) = (\mathsf{CG}) \otimes \mathbf{I}$. This means that we can evaluate both homomorphisms by simply evaluating the homomorphism applied to the finite gate G. Since G is finite, computing the homomorphism is constant-time.

It turns out that computing the determinant is also easily extendable: $\det(\mathsf{G} \otimes \mathbf{I}_{n-m}) = \det(\mathsf{G})^{2^{n-m}}$,[6] which can be computed by repeated squaring. So extendability alone is not an issue with determinants. A more problematic issue is the fact that the determinant of G gets raised to a power. Indeed, observe that any Clifford+T gate has determinant in $\{\pm 1, \pm i\}$ and $m \leq 2$. As such, for $n \geq 4$, the determinant of any Clifford+T circuit acting on n qubits is identically 1. Thus, the determinant gives us no useful information, at least for the Clifford+T gate set.

A more general view. But wait, can't Clifford+T simulate *any* quantum circuit since they are universal? Thus, shouldn't we be able simulate any unitary, even ones that don't have determinant 1? Yes, but there are two caveats. First, recall that universal gate sets only simulate arbitrary unitaries up to global phase, and a global phase would naturally alter the determinant. This can be remedied by our convention that a circuit also carry a global phase term; then to compute the determinant of the unitary, we can compute the determinant of each gate (when considered as a unitary over all n qubits), multiply the determinants together, and also multpily by the result by the global phase (raised to the appropriate power). Using the Clifford+T universal gate set, all the information about the determinant is actually in the global phase, so the gates themselves can be ignored.

The second caveat is that efficient quantum circuits often involve ancilla qubits. If we had an ancilla-free circuit C computing a unitary U (including global phase), then we can in fact compute $\det(U)$ by computing the determinant of each gate of C (when considering the gates as a unitary over all n qubits), and then multiplying. However, it turns out that the presence of ancilla qubits complicates our goal of applying homomorphisms to the circuits.

Consider a circuit C implementing an n-qubit unitary U, but where C makes use of m ancilla qubits. That is, C is actually approximating an m-qubit extension $\overline{U}$ of U (see Definition 10). Define $N = 2^n$, $W = 2^{m+n}$, and $M = W - N$. This means that for any state $|\psi\rangle$ on n qubits, $\overline{U}|\psi\rangle|0^m\rangle = (U|\psi\rangle)|0^m\rangle$. Writing $\overline{U}$ out in matrix form, it will look like:

$$\overline{U} = \begin{pmatrix} U & \\ & V \end{pmatrix}$$

Above, $V \in \mathbb{C}^{M \times M}$, and it is allowed to be truly arbitrary, as it corresponds to the action of $\overline{U}$ when the ancillas are something other than $|0^m\rangle$.

[6] Recall that $\mathbf{I}_{n-m}$ acts on $n-m$ qubits, so it is actually a matrix of size 2^{n-m}.

Now observe that $\det(\overline{U}) = \det(U)\det(V)$. Moreover, if C approximates $\overline{U}$ to sufficiently-small exponential error, then $\det(C) \approx \det(\overline{U}) = \det(U)\det(V)$. But since V has no necessary relationship to U, $\det(\overline{U})$ and hence $\det(C)$ is completely untied to $\det(U)$. Hence, computing the determinant of a circuit with ancillas yields no useful information about the unitary that the circuit evaluates.

Ancilla-respecting homomorphisms. What we need then is not an arbitrary homomorphism, but one that in some sense preserves the ancilla structure. To simplify the discussion, we will assume that C exactly implements the unitary $\overline{U}$; this assumption can be relaxed to sufficiently small exponential error, but would not apply to the more relaxed polynomial-error case.

Specifically, we want to compute some homomorphism $\Phi(U)$ taking an n-qubit unitary U and outputting an n'-qubit unitary, but we only have a circuit C implementing an m-qubit extension of U. We want to implement the homomorphism Φ on U by implementing a homomorphism Φ' on C. This gives rise to the following definition:

Definition 41. *Let Φ, Ψ be two homomorphisms on unitary groups, where Φ acts on n qubit unitaries and Ψ acts on $n+m$-qubit unitaries. We say that Ψ is an $(m \to m')$-Ancilla-respecting Implementation of Φ if the following holds. For unitaries U on n qubits and all unitaries C that are m-qubit extensions of U, then $\Psi(C)$ is an m'-qubit extension of $\Phi(U)$.*

That is, if C implements U using m ancilla qubits, then $\Psi(C)$ implements $\Phi(U)$ using m' ancilla qubits. This is because $\Psi(C)(|\psi\rangle \otimes |0^{m'}\rangle) = (\Phi(U)|\psi\rangle) \otimes |0^{m'}\rangle$.

$U^*, \mathsf{C}U$, and by composition, $\mathsf{C}U^*$, each have efficient ancilla-respecting implementations. For example, $U' \mapsto \mathsf{C}U'$ for U' acting on $n+m$ qubits is an ancilla-respecting implementation of $U \mapsto \mathsf{C}U$ for U acting on n qubits.

The question is then: *what "nice" unitary homomorphisms admit ancilla-respecting implementations?* Are there any other nice unitary homomorphisms besides $U^*, \mathsf{C}U$, and compositions?

7.1 Efficient Ancilla-respecting Implementations

Now we explain that the only homomorphisms that admit *efficient* ancilla-respecting implementations are $U^*, \mathsf{C}U$, and compositions thereof. Formally:

Theorem 42. *Let Φ be a group homomorphism from unitaries on n qubits to unitaries on n' qubits. Suppose that Φ is continuous. Let $m \geq 1$. Then Φ admits an efficient $(m \to m')$-ancilla-respecting implementation for some m' if and only if $\Phi(U)$ can be computed by a polynomial number of queries to oracles for $\mathsf{C}U$ and $\mathsf{C}U^*$.*

Proof. We prove Theorem 42 by showing that, for any continuous homomorphism Φ that cannot be obtained by a polynomial number of queries to $\mathsf{C}U$ and $\mathsf{C}U^*$, m' must be super-polynomial. In other words, in order to implement Φ, one must use a super-polynomial number of ancillas. Such an implementation cannot be efficient.

Representation Theory of Unitary Groups. A group homomorphism between unitary groups is just a unitary representation of the unitary group. It is therefore useful to recall the representation theory of finite-dimensional unitary groups.

Let $V = \mathbb{C}^{2^n}$ and $W = V^q$. The symmetric group S_q acts on W by permuting the copies of V. There are certain subspaces of W which are preserved under action by S_q. We will denote these by W_D, where the indices D are Young diagrams of q boxes.

Given any unitary U acting on V, $U^{\otimes q}$ acting on W preserves the subspace W_D for every D. Its action on W_D is described by the Schur functor $\mathbb{S}_D(U)$. If W_D has dimension M, we can get a homomorphism from unitaries on n qubits to unitaries on $\log M$ qubits by mapping $\mathbb{C}^{2^m}$ into W_D, and then performing the unitary $U^{\otimes q}$.

The representation theory of finite unitary groups tells us that any continuous representation of U is a direct sum of unitaries that are isomorphic $\mathbb{S}_D(U) \otimes \det(U)^k$ for integers k. For example, $\mathsf{C}U$ is just the direct sum $\mathbf{I} \oplus U$. For another example, U^* is isomorphic to U under the non-trivial $\mathbb{C}$ homomorphism $z \mapsto z^*$.

What representations admit efficient ancilla-respecting implementations? Direct sums can be implemented by controlling on other qubits, and direct products simply correspond to independently acting on two sets of qubits. Thus, it suffices to explore which $\mathbb{S}_D(U)$ have efficient ancilla-respecting implementations.

Note that for polynomial q, the representation $\mathbb{S}_D(U)$ is query efficient, since it can be implemented using q queries to U. However, we now argue that, for super-polynomial q, this is no longer the case.

We focus on the case of determinants for simplicity. Consider unitaries acting on n qubits, and suppose there is even a single ancilla qubit. This means the circuit computing the unitary acts on $n+1$ qubits. Thus we have $N = 2^n$ and $M = 2^{n+1} - N = N$.

Now suppose there is an ancilla-respecting implementation Φ' of the determinant. We will consider applying Φ' to diagonal unitaries C, where we will treat the diagonal entries of C as formal variables. Since C is diagonal, it automatically has the form

$$C = \begin{pmatrix} U & \\ & V \end{pmatrix},$$

Hence by the fact that Φ' is an ancilla-respecting implementation, we have that

$$\Phi'(C) = \begin{pmatrix} \det(U) & \\ & V' \end{pmatrix}.$$

Let S be a subset of $\{0,1\}^{2n}$ of size N. Let P_S be some permutation of $\{0,1\}^{2n}$ which maps S into the first N positions. We will abuse notation and let P_S also denote the associated permutation matrix. Let $Q_S = \Phi'(P_S)$.

Consider PCP^{-1}, which is also a diagonal matrix obtained by permuting the diagonal entries of C according to P. Let U_S denote the upper-left $N \times N$ diagonal matrix of C. Then U_S is a diagonal matrix whose entries are exactly the entries of C belonging to the set S. Moreover, observe that $\Phi'(P_S C P_S^{-1}) = Q_S \Phi'(C) Q_S^{-1}$. The upper-left entries of $Q_S \Phi'(C) Q_S^{-1}$ therefore contains the determinant $\det(U_S)$. Thus, we can construct $\det(U_S)$ as a linear combination of the elements in $\Phi'(C)$. On the other hand, $\det(U_S)$ is just the product of the diagonal entries of C belonging to the set S.

By varying over all possible sets S, we can therefore obtain all possible products of N diagonal elements of C as linear combinations of the elements of $\Phi'(C)$. Each of these products are linearly independent formal functions of the underlying elements, and therefore span a space of $\binom{2N}{N} \approx$

$2^{2N}/\sqrt{\pi N}$. Therefore, the elements of $\Phi'(C)$ must span a space of at least this dimension. But since there are only $(N'+M')^2$ such elements of $\Phi'(C)$, we must have that $N'+M' \gtrsim 2^N/(\pi N)^{1/4}$. The number of qubits is therefore at least $n'+m' = \log(N'+M') \gtrsim N - \log(\pi N)/4 \geq 2^n - n/4 - 1$. Thus, any ancilla-respecting implementation of the determinant must have an exponential number of qubits, and therefore cannot be efficient.

A similar analysis extends to any representation, showing that an ancilla-respecting representation of $\mathbb{S}_D(U)$ must operate on a number of qubits that grows polynomially with q. Hence, only the case of polynomial q can be efficient. □

7.2 Is This The End?

Above, we argued that "natural" homomorphisms aside from $U, U^*, \mathsf{C}U, \mathsf{C}U^*$ and those computable via polynomially-many queries to these cannot be implemented efficiently. Specifically, we assumed that the homomorphisms are computable by continuous functions, and moreover that they *exactly* preserved ancillas.

There are therefore two possible ways to overcome this barrier to give more general homomorphisms:

- Develop a homomorphism that uses non-continuous functions. This seems very unlikely to yield an efficient homomorphism
- Develop a homomorphism with an efficient implementation that only *approximately* preserves ancilla structure. This means that if

$$C = \begin{pmatrix} U & \\ & V \end{pmatrix},$$

then $\Phi'(C)$ satisfies

$$\Phi'(C) \approx \begin{pmatrix} \Phi(U) & \\ & V' \end{pmatrix}.$$

for an appropriate notion of "$\approx$." We leave it as an interesting direction for future work to develop such homomorphisms or show barriers to constructing them.

8 Ancilla-Free Complexity, and from Classical to Quantum Hardness

For a function $C : \{0,1\}^n \to \{0,1\}$, let U_C be the unitary $U_C|x,b\rangle = |x, b \oplus C(x)\rangle$. We recall a result of [Cle91].

Theorem 43 ([Cle91])**.** *If C is computable by log-depth classical circuits, then there is a polynomial-sized classical reversible circuit U'_C mapping (x,b,c,d) to $(x, b \oplus C(x), c, d)$, where b, c, d are bits.*

As any classical reversible circuit is also a quantum circuit, then we can compute the unitary U_C using two ancilla qubits for c, d, with a polynomial-sized circuit (assuming C is log-depth). As a result, we have:

Corollary 44. *For C being log-depth circuits, $\{U_C\}_C \in \mathsf{UnitaryBQP}^2_{\mathsf{exp}}$.*

We next explore the interesting question eliminating the need for ancillas.

8.1 Exponential-sized ancilla-free computation of U_C

Theorem 45. *For $\{U_s\}_s$ being any sequence of unitaries where the matrix for U_s can be computed to polynomially-many bits of precision in exponential time in $|s|$. Then $\{U_s\}_s \in \mathsf{UnitaryE}^0_{\mathsf{exp}}$.*

The proof is given in Appendix B following the standard proof of universality of single qubit gates and CNOT. The main difference is that the standard proof uses ancilas, and we need to carefully modify the proof to eliminate ancillas.

Note that for circuits C, the entries of U_C can be computed in polynomial-time. Namely, $\langle x, b|U_C|x', b'\rangle$ is 1 if $x = x'$ and $b \oplus b' = C(x)$, and otherwise the value is 0. Thus, we have that:

Corollary 46. *For C being interpreted as circuits, $\{U_C\}_C \in \mathsf{UnitaryE}^0_{\mathsf{exp}}$.*

8.2 No polynomial-sized ancilla-free circuits unless PH $\subseteq$ BPP

Theorem 47. *Suppose $\mathsf{PH} \not\subseteq \mathsf{BPP}$. Then for C being log-depth circuits, we have $\{U_C\}_C \notin \mathsf{UnitaryBQP}^0_{\mathsf{exp}}$. In particular, $\mathsf{PH} \not\subseteq \mathsf{BPP}$ implies $\mathsf{UnitaryBQP}^0_{\mathsf{exp}} \subsetneq \mathsf{UnitaryBQP}^2_{\mathsf{exp}}$.*

Proof. Suppose $\{U_C\}_C \in \mathsf{UnitaryBQP}^0_{\mathsf{exp}}$, when C is interpreted as log-depth circuits. We will give a deterministic algorithm for solving $\oplus\mathsf{P}$, the set of languages decideable by polynomial-time non-deterministic Turing machines, where acceptance means that the number of accepting paths is odd. Thus, $\oplus\mathsf{P} \subseteq \mathsf{P}$. Then recall that $\mathsf{PH} \subseteq \mathsf{BPP}^{\oplus P}$ [Tod91]. Thus, if $\{U_C\}_C \notin \mathsf{UnitaryBQP}^0_{\mathsf{exp}}$, we have that $\mathsf{PH} \subseteq \mathsf{BPP}^P = \mathsf{BPP}$, contradicting the assumption of Theorem 47.

We now give the algorithm. Set $\epsilon = 2^{-n}/10$. Since we assume $\{U_C\}_C \in \mathsf{UnitaryBQP}^0_{\mathsf{exp}}$, there is a deterministic Turing machine $M(C)$ which outputs a circuit $\hat{U}_C$ computing U_C up to error ϵ in polynomial time.

First, we show $\det(\hat{U}_C)$ is very close to $\det(U_C)$. Indeed, recall the inequality of ([IR08], Corollary 2.14), which states that for matrices A, B of dimension N,

$$\frac{|\det(A) - \det(B)|}{|\det(A)|} \leq (1 + \|A\|^{-1}\|B - A\|)^N - 1.$$

where $\|\cdot\|$ denotes the operator norm. Set $N = 2^n$, $A = U_C$, and $B = \hat{U}_C$. Since U_C is unitary, $\|A\| = \|A\|^{-1} = 1$. Recall that $\|\hat{U}_C - U_C\| \leq \epsilon = 2^{-n}/10$. Then

$$|\det(U_C) - \det(\hat{U}_C)| \leq (1+\epsilon)^{2^n} - 1 = (1+\epsilon)^{1/10\epsilon} - 1 \leq e^{1/10} - 1 \leq 0.11.$$

Observe that $\det(U_C)$ is exactly $(-1)^p$ where p is the number of accepting inputs of C; in particular $\det(U_C) \in \{1, -1\}$. Since $\det(\hat{U}_C)$ is within 0.11 of $\det(U_C)$, rounding the real part to the nearest integer gives $\lfloor \mathsf{Re}(\det(\hat{U}_C)) \rceil = \det(\hat{U}_C)$. Hence by computing $\det(\hat{U}_C)$ we learn $(-1)^p$ and hence $p \bmod 2$.

This gives us a polynomial-time algorithm for computing $p \bmod 2$ for any log-depth circuit C. In particular, this captures formula, allowing us to solve $\oplus\mathsf{SAT}$, which is $\oplus\mathsf{P}$-complete. Thus, $\oplus\mathsf{P} = \mathsf{P}$. This completes the proof of Theorem 47. □

References

[AEH+24] Chris Akers, Netta Engelhardt, Daniel Harlow, Geoff Penington, and Shreya Vardhan. The black hole interior from non-isometric codes and complexity. *Journal of High Energy Physics*, 2024(6):155, 2024.

[AK07] Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3(7):129–157, 2007.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Berlin, Heidelberg, December 2011.

[BEM+23] John Bostanci, Yuval Efron, Tony Metger, Alexander Poremba, Luowen Qian, and Henry Yuen. Unitary complexity and the uhlmann transformation problem, 2023.

[BFV20] Adam Bouland, Bill Fefferman, and Umesh V. Vazirani. Computational pseudorandomness, the wormhole growth paradox, and constraints on the AdS/CFT duality (abstract). In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 63:1–63:2. LIPIcs, January 2020.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, August 2001.

[BGT21] Adam Bouland and Tudor Giurgica-Tiron. Efficient universal quantum compilation: An inverse-free solovay-kitaev algorithm, 2021.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002.

[BM24] Zvika Brakerski and Nir Magrafta. Real-valued somewhat-pseudorandom unitaries. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part II*, volume 15365 of *LNCS*, pages 36–59. Springer, Cham, December 2024.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[Cle91] Richard Cleve. Complexity theoretic issues concerning block ciphers related to DES. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 530–544. Springer, Berlin, Heidelberg, August 1991.

[CMSZ22] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: Breaking the quantum rewinding barrier. In *62nd FOCS*, pages 49–58. IEEE Computer Society Press, February 2022.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, August 2008.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 89–98. ACM Press, June 2011.

[IR08] Ilse C. F. Ipsen and Rizwana Rehman. Perturbation bounds for determinants and characteristic polynomials. *SIAM Journal on Matrix Analysis and Applications*, 30(2):762–776, 2008.

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 126–152. Springer, Cham, August 2018.

[KTP20] Isaac Kim, Eugene Tang, and John Preskill. The ghost in the radiation: robust encodings of the black hole interior. *Journal of High Energy Physics*, 2020(6):31, 2020.

[LQS+24] Chuhan Lu, Minglong Qin, Fang Song, Penghui Yao, and Mingnan Zhao. Quantum pseudorandom scramblers. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part II*, volume 15365 of *LNCS*, pages 3–35. Springer, Cham, December 2024.

[Mec19] Elizabeth S. Meckes. *The Random Matrix Theory of the Classical Compact Groups*. Cambridge Tracts in Mathematics. Cambridge University Press, 2019.

[MH25] Fermi Ma and Hsin-Yuan Huang. How to construct random unitaries. In *STOC'25*, 2025.

[MPSY24] Tony Metger, Alexander Poremba, Makrand Sinha, and Henry Yuen. Simple constructions of linear-depth t-designs and pseudorandom unitaries. In *65th FOCS*, pages 485–492. IEEE Computer Society Press, October 2024.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Berlin, Heidelberg, February 2004.

[MZ03] Michele Mosca and Christof Zalka. Exact quantum fourier transforms and discrete logarithm algorithms, 2003.

[PRV24] Alexander Poremba, Seyoon Ragavan, and Vinod Vaikuntanathan. Cloning games, black holes and cryptography. Cryptology ePrint Archive, Paper 2024/1826, 2024.

[Tod91] Seinosuke Toda. Pp is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20:865–877, 1991.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Berlin, Heidelberg, April 2012.

[YE23] Lisa Yang and Netta Engelhardt. The complexity of learning (pseudo)random dynamics of black holes and other chaotic systems, 2023.

A Gates and identities

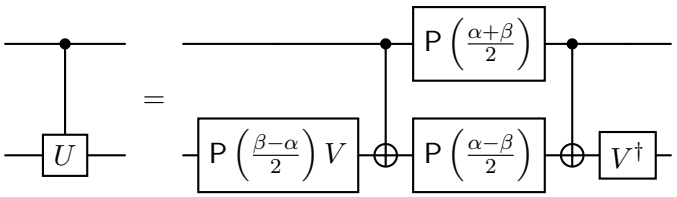
$$\mathsf{P}(\theta) = \begin{pmatrix} 1 & \\ & e^{i\theta} \end{pmatrix} \qquad \mathsf{T} = \begin{pmatrix} 1 & \\ & e^{i\pi/4} \end{pmatrix} \qquad \mathsf{C}U = \begin{pmatrix} \mathbf{I} & \\ & U \end{pmatrix}$$

$$\mathsf{X} = \mathsf{NOT} = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \qquad \mathsf{Y} = \begin{pmatrix} & -i \\ i & \end{pmatrix} \qquad \mathsf{Z} = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \qquad \text{(Pauli)}$$

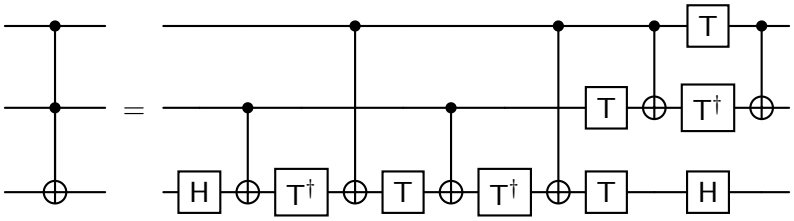
$$\mathsf{CNOT} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{pmatrix} \qquad \mathsf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \mathsf{S} = \begin{pmatrix} 1 & \\ & i \end{pmatrix} \qquad \text{(Clifford)}$$

Above, U is a unitary matrix, and $\mathbf{I}$ is the identity matrix of the same dimensions as U. $\mathsf{C}U$ is called a "controlled" gate.

Useful Quantum Gate Identities. Let U be a 1-qubit unitary. U is diagonalizeable, with eigenvalues of norm 1. Thus we can write U as $U = V^\dagger \begin{pmatrix} e^{i\alpha} & \\ & e^{i\beta} \end{pmatrix} V$. Then:



(6)



(7)

B Proof of Theorem 45

The proof follows the usual proof for showing the universality of 2-qubit quantum gates, but we are careful to implement the circuit in such a way so as to not use ancillas, and also to make sure we preserve the global phase. We will assume we have access to CNOT gates, plus arbitrary single-qubit gates, which can be approximated from any universal gate set using Solovay-Kitaev, up to global phase. Since our model allows for specifying a global phase, we can therefore implement any single-qubit gate even with phase.

Implementing C^nU for single-qubit U. Let U be a single-qubit unitary. We now explain how to implement C^nU, the circuit controlled on n qubits, which applies U to a target qubit if and only if the control qubits are identically $|0^n\rangle$. This is the main part of the proof where we need to be careful regarding ancillas, and where we depart from typical treatments.

Our construction is inductive. If $n = 0$, the task is just to implement U, which requires only one single-qubit gate. Now assume we have written $\mathsf{C}^{n-1}U$ as a circuit of size $T(n-1)$ comprising CNOT gates and single qubit gates. To get C^nU, we simply control every single gate in $\mathsf{C}^{n-1}U$ on the n-th control qubit. Thus, we have a circuit of size $T(n-1)$ consisting of CCNOT and $\mathsf{C}U'$ gates for various single-qubit U' gates. We can implement CCNOT using a constant number of CNOT and single-qubit gates in a standard way (provided in Equation 7 for completeness). We can likewise implement an arbitrary $\mathsf{C}U'$ gate using a constant number of CNOT and single-qubit gates (provided in Equation 6 for completeness). Thus, we have a circuit of size at most $c * T(n-1)$ for computing C^nU for some constant c, giving the recurrence $T(n) \leq c*T(n-1)$, or $T(n) \leq c^n = 2^{O(n)}$. The running time to construct the circuit is proportional to the size of the circuit, meaning the overall running-time is exponential.

Implementing "2-level" Unitaries. A "2-level" unitary U is one for which there exist computational basis vectors $x, y \in \{0,1\}^n$ such that:

- On computational basis vectors $|z\rangle$ such that $z \notin \{x, y\}$, $U|z\rangle = |z\rangle$.
- On the computational basis vectors $|x\rangle$ and $|y\rangle$, the action of U is that of a single-qubit unitary U'.

$U = \mathsf{C}^{n-1}U'$ is an example where $x = 1^{n-1}0$ and $y = 1^{n-1}1$. Viewing U as a matrix, a "2-level" unitary is one that is the identity except for two rows and the corresponding two columns.

We can easily implement the 2-level unitary U using $\mathsf{C}^{n-1}U'$. First, by applying an X gate if needed, we can assume that x, y are both not 0^n. The idea is to find a linear bijection V over $\mathbb{F}_2^n$ such that $Vx = 1^{n-1}0$ and $Vy = 1^{n-1}1$. We will overload notation and define V also as the unitary such that $V|z\rangle = |Vz\rangle$. The elementary row operations making up V (as an invertible linear transformation over $\mathbb{F}_2^n$) will consist of adding one row to another (mod 2) or swapping two rows (which in turn can be decomposed into three operations of adding one row to another). Adding one row to another corresponds to applying a CNOT gate on the appropriate qubits. Thus, the map $V|z\rangle = |Vz\rangle$ can be implemented with just $O(n^2)$ CNOT gates. Then $U = V^\dagger \mathsf{C}^{n-1}U'V$.

Implementing General Unitaries. We now explain how to implement a general unitary U on n qubits. This will roughly correspond to performing Gaussian elimination on the $2^n \times 2^n$ matrix representing U, except modified so that the elementary row operations are "2-level" unitaries.

Consider the matrix corresponding to the U; the (i,j) entry will be denoted by $U_{i,j}$. By unitarity, some entry in the first column will be non-zero; by swapping into the first entry if necessary using a 2-level unitary, we can assume that $U_{1,1}$ is non-zero.[7] Then there will exist some 2-level unitary $V^{1,2}$ operating on just the first two rows such that $(V^{1,2}U)_{2,1} = 0$. Let $U' = V^{12}U$. Then there will exist some 2-level unitary $V^{1,3}$ operating on the first and third rows such that $(V^{1,3}U')_{3,1} = 0$. Observe that the second row is unaffected, so $(V^{1,3}U')_{2,1} = 0$. Define $U'' = V^{1,3}U'$.

Continuing in this way, we eventually arrive at $V^{1,n}V^{1,n-1}\cdots V^{1,2}U$ being a matrix whose entire first column except the first entry is 0. By unitarity, we therefore know that the entire first row except the first entry is 0 as well. By unitarity we also know that the first entry in the first row has norm 1; by putting an appropriate phase in $V^{1,n}$ we can assume the entry is exactly 1.

Then the sub-matrix of $V^{1,2^n}V^{1,2^n-1}\cdots V^{1,2}U$ excluding the first row and column must also be a unitary matrix, and we can apply the same procedure to it. By iterating this process, we eventually arrive at $(\prod_{1\leq i<j\leq 2^n} V^{i,j})U = \mathbf{I}$, where each $V^{i,j}$ is a 2-level unitary. Thus $U = \prod_{1\leq i<j\leq 2^n}(V^{i,j})^\dagger$, where each $(V^{i,j})^\dagger$ is also a 2-level unitary. Thus, we have expressed U as a product of $O(4^n)$ 2-level matrices; the overall running time of this procedure is also exponential time.

As each $V^{i,j}$ takes exponential time to construct and there are an exponential number of them, the overall running time to construct the circuit for U is exponential. Moreover, the resulting circuit requires no ancilla qubits.

C Implementing Controlled Gates

In this section, we recall known results showing that we can implement controlled gates for the gate set $\{\mathsf{H}, \mathsf{CNOT}, \mathsf{T}\}$ with a constant blow-up and zero error. Thus, for a Clifford+T circuit, controlling that circuit only gives a constant blow-up. This is in contrast to general gate sets, for which the best seems to be something like Proposition 22, which in turn employs Solovay-Kitaev, and thus incurs some error, with a blow-up that depends logarithmically on the error.

In the following, observe that $\mathsf{T}^2 = \mathsf{S}$, $\mathsf{T}^4 = \mathsf{Z} = \mathsf{Z}^\dagger$, $\mathsf{T}^6 = \mathsf{S}^\dagger$, and $\mathsf{T}^7 = \mathsf{T}^\dagger$, so we can assume we have the gates $\mathsf{S}, \mathsf{Z} = \mathsf{Z}^\dagger, \mathsf{S}^\dagger, \mathsf{T}^\dagger$ without introducing new gates to our gate set. Likewise, $\mathsf{X} = \mathsf{HZH}$, so we can assume we have the gate X.

Controlled CNOT (Toffoli). For a single-qubit gate $\mathsf{U} \in \{\mathcal{H}, \mathsf{T}\}$, write $\mathsf{U}_\mathcal{A}$ to denote the gate applied to qubit $\mathcal{A}$. Write $\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}$ to denote CNOT applied to the two qubits $\mathcal{A},\mathcal{B}$, NOT acting on qubit $\mathcal{B}$ and being controlled by qubit $\mathcal{A}$. Similarly write CCNOT as the controlled CNOT, also known as Tofolli, with the $\mathsf{CNOT}_{\mathcal{B},\mathcal{C}}$ being controlled by qubit $\mathcal{A}$. Then we have that:

$$\mathsf{CCNOT}_{\mathcal{A},\mathcal{B},\mathcal{C}} = \mathsf{H}_\mathcal{C}\mathsf{CNOT}_{\mathcal{B},\mathcal{C}}\mathsf{T}_\mathcal{C}^\dagger\mathsf{CNOT}_{\mathcal{A},\mathcal{C}}\mathsf{T}_\mathcal{C}\mathsf{CNOT}_{\mathcal{B},\mathcal{C}}\mathsf{T}_\mathcal{C}^\dagger\mathsf{CNOT}_{\mathcal{A},\mathcal{C}}\mathsf{T}_\mathcal{B}\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}\mathsf{T}_\mathcal{A}\mathsf{T}_\mathcal{B}^\dagger\mathsf{T}_\mathcal{C}\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}\mathsf{H}_\mathcal{C}.$$

Controlled H. We can implement a Controlled H gate, denoted $\mathsf{CH}_{\mathcal{A},\mathcal{B}}$ with H acting on qubit $\mathcal{B}$ and controlled by qubit $\mathcal{A}$, as follows: $\mathsf{CH}_{\mathcal{A},\mathcal{B}} = e^{i\pi/4}\mathsf{H}_\mathcal{B}\mathsf{S}_\mathcal{A}\mathsf{S}_\mathcal{B}^\dagger\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}\mathsf{H}_\mathcal{B}\mathsf{T}_\mathcal{B}\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}\mathsf{T}_\mathcal{B}\mathsf{H}_\mathcal{B}\mathsf{S}_\mathcal{B}\mathsf{X}_\mathcal{B}$. We can eliminate the global phase using that $\mathsf{XT}^\dagger\mathsf{XT}^\dagger = e^{-i\pi/4}$.

Controlled T. We can implement a Controlled T gate, denoted $\mathsf{CT}_{\mathcal{A},\mathcal{B}}$ with T acting on qubit $\mathcal{B}$ and controlled by qubit $\mathcal{A}$, using an ancilla qubit $\mathcal{C}$, as follows: $\mathsf{CT}_{\mathcal{A},\mathcal{B}} = \mathsf{CCNOT}_{\mathcal{A},\mathcal{B},\mathcal{C}}\mathsf{T}_\mathcal{C}\mathsf{CCNOT}_{\mathcal{A},\mathcal{B},\mathcal{C}}$.

[7] It is not actually strictly necessary to make $U_{1,1}$ zero, and we could instead just re-index the rows of U so that the row we call "1" has a non-zero entry in the first column.

Introducing the gate $\sqrt{\mathsf{T}}$, we can get rid of the ancilla qubit and implement CT as $\mathsf{CT}_{\mathcal{A},\mathcal{B}} = \sqrt{\mathsf{T}}_{\mathcal{A}}\sqrt{\mathsf{T}}_{\mathcal{B}}\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}\sqrt{\mathsf{T}}^{\dagger}_{\mathcal{B}}\mathsf{CNOT}_{\mathcal{A},\mathcal{B}}$.

Also, note that CT cannot be implemented using just $\mathsf{CNOT}, \mathsf{H}, \mathsf{T}$ gates, without incurring a global phase. To see this, note that $\mathsf{CNOT}, \mathsf{H} \otimes \mathbf{I}, \mathsf{T} \otimes \mathbf{I}$ all have determinants in $\{0, i, -1, -i\}$, since the $e^{i\pi/4}$ term in T gets repeated in $\mathsf{T} \otimes \mathbf{I}$. Thus, any 2-qubit circuit comprised of the gates $\mathsf{CNOT}, \mathsf{H}, \mathsf{T}$ will have a determinant in this set. Meanwhile, CT has determinant $e^{i\pi/4}$. Note that the implementation above using an ancilla, the circuit over $\mathcal{A}, \mathcal{B}, \mathcal{C}$ also has determinant in this set for the same reason. However, the 4×4 sub-matrix corresponding to $\mathcal{C}$ being $|0\rangle$ will be exactly CT and have the correct determinant $e^{i\pi/4}$. The other 4×4 sub-matrix corresponding to $\mathcal{C}$ being $|1\rangle$ will actually be the unitary $e^{i\pi/4}\mathsf{CT}^{\dagger}$, which has determinant $e^{i3\pi/4}$. The overall determinant is the product of these two, giving -1, which is in the set of allowed determinants $\{0, i, -1, -i\}$.