Quantum Catalytic Space

Harry Buhrman
Quantinuum London, UK
QuSoft, Amsterdam, The Netherlands

Marten Folkertsma
CWI, Amsterdam, The Netherlands
QuSoft, Amsterdam, The Netherlands

Ian Mertz
Charles University, Prague, Czech Republic

Florian Speelman
University of Amsterdam, The Netherlands
QuSoft, Amsterdam, The Netherlands

Sergii Strelchuk
University of Oxford, UK

Sathyawageeswar Subramanian
University of Cambridge, UK

Quinten Tupker
CWI, Amsterdam, The Netherlands
QuSoft, Amsterdam, The Netherlands

Abstract

Space complexity is a key field of study in theoretical computer science. In the quantum setting there are clear motivations to understand the power of space-restricted computation, as qubits are an especially precious and limited resource.

Recently, a new branch of space-bounded complexity called catalytic computing has shown that reusing space is a very powerful computational resource, especially for subroutines that incur little to no space overhead. While quantum catalysis in an information theoretic context, and the power of "dirty" qubits for quantum computation, has been studied over the years, these models are generally not suitable for use in quantum space-bounded algorithms, as they either rely on specific catalytic states or destroy the memory being borrowed.

We define the notion of catalytic computing in the quantum setting and show a number of initial results about the model. First, we show that quantum catalytic logspace can always be computed quantumly in polynomial time; the classical analogue of this is the largest open question in catalytic computing. This also allows quantum catalytic space to be defined in an equivalent way with respect to circuits instead of Turing machines. We also prove that quantum catalytic logspace can simulate log-depth threshold circuits, a class which is known to contain (and believed to strictly contain) quantum logspace, thus showcasing the power of quantum catalytic space. Finally we show that both unitary quantum catalytic logspace and classical catalytic logspace can be simulated in the one-clean qubit model.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Complexity classes

Keywords and phrases quantum computing, quantum complexity, space-bounded algorithms, catalytic computation, one clean qubit

Digital Object Identifier 10.4230/LIPIcs.TQC.2025.11

Related Version *Full Version*: `https://arxiv.org/abs/2506.16324`

20th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2025).
Editor: Bill Fefferman; Article No. 11; pp. 11:1–11:24
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding *Marten Folkertsma*: Supported by the Dutch Ministry of Economic Affairs and Climate Policy (EZK), as part of the Quantum Delta NL programme.
Ian Mertz: Supported by the Grant Agency of the Czech Republic under the grant agreement no. 24-10306S and by the Center for Foundations of Contemporary Computer Science (Charles Univ. project UNCE 24/SCI/008).
Florian Speelman: Supported by the Dutch Ministry of Economic Affairs and Climate Policy (EZK), as part of the Quantum Delta NL program, and the project Divide and Quantum "D&Q" NWA.1389.20.241 of the program "NWA-ORC", which is partly funded by the Dutch Research Council (NWO).
Sergii Strelchuk: Supported by a Royal Society University Research Fellowship and the EPSRC (RoaRQ), Investigation 005 [grantreference EP/W032635/1].
Sathyawageeswar Subramanian: Supported by a Royal Commission for the Exhibition of 1851 Research Fellowship.
Quinten Tupker: Supported by the Dutch National Growth Fund (NGF), as part of the Quantum Delta NL program.

Acknowledgements We thank [24] for pointing us to [3]. Part of this work was done while HB was at CWI, IM was at University of Warwick, SS (1) was at Cambridge University and University of Warwick, and SS (2) was at University of Warwick. SS (2) acknowledges support from the Royal Society University Research Fellowship.

1 Introduction

Space is one of the cornerstones of theoretical computer science, and the study of space-bounded computations has been crucial in the development of complexity theory. Investigating logspace computations revealed the limits of efficient computation under memory constraints and has led to striking results such as Savitch's theorem [38] and $\mathsf{NL} = \mathsf{coNL}$ [25, 41]. Logspace reductions are essential in classifying problems as NL-complete or P-complete, and leading to techniques for efficient parallelization and algorithm design.

Many graph and database problems rely on logspace techniques, making them relevant for query optimization, data retrieval, and formal verification. Furthermore, logspace computations have practical applications in streaming algorithms, embedded systems, cryptography, and model checking, where minimizing memory usage is critical.

The emergence of quantum computing has led to remarkable theoretical speedups over the best known classical algorithms. The promise of exponential computational advantage in using principles of quantum mechanics to process information comes with formidable experimental challenges of building and maintaining quantum computers that can implement long sequences of coherent operations. This led to a renewed interest in the structure of quantum space.

1.1 Space in quantum computation

Understanding the true extent of the power of quantum computing in a variety of space-constrained settings is a major challenge. In contrast to the classical setting where adding a reasonable amount of extra memory to support computations is routinely achievable, producing and maintaining multiple qubits is exceptionally difficult due to several fundamental physical, engineering, and scalability issues. Qubits are fragile and susceptible to decoherence, and maintaining long coherence times becomes significantly harder as the number of qubits increases. Furthermore, quantum error rates scale with the number of qubits, making fault-tolerant quantum computing a major challenge. In the quantum computational setting, space thus comes at a premium, and increasing the amount of space available for computation requires overcoming fundamental challenges to reduce error rates, increase control precision, and maintain entanglement across multiple systems, to name but a few.

The characterization of quantum logspace (QL) and the study of the computational power of bounded-error quantum logarithmic space (BQL) and its relationship to classical complexity classes was first done by Watrous [44], where it was established that $\mathsf{BQL} \subseteq \mathsf{P}$. This showed that any problem solvable in quantum logspace with bounded error is also solvable in polynomial time by a classical deterministic machine. In later work, Watrous [43] showed that $\mathsf{QSPACE}(s) \subseteq \mathsf{SPACE}[O(s^2)]$ for all $s \geq \log n$, even when the quantum machine is allowed to err with probability arbitrarily close to $1/2$; this confirms that quantum logspace computations remain simulable within polynomial space, and is consistent with classical space complexity results such as Savitch's theorem. His work also established that quantum logspace can efficiently solve certain algebraic problems, including the *group word problem for solvable groups*, which lacks efficient classical logspace algorithms [43].

These above obstacles prompted the search for extra ingredients which could lift restricted models of quantum computation (for example – realized by quantum circuits which are classically efficiently simulatable) to regain the power of universal quantum computation. These extra ingredients (e.g. magic state injection) are usually studied in the context of unrestricted space and there has as of yet been no attempt to investigate them under space restrictions.

On the other hand, there have been several notable results that illuminate various properties of quantum logspace. One of the earliest findings shows that any quantum computation that can be performed with logarithmic space can also be efficiently simulated using matchgate circuits of polynomial width, and vice versa [26]. Following this characterisation, there have been a series of further results indicating that quantum logspace describes a non-trivial class of computations. Ta-Shma [42] showed that given a matrix with a bounded condition number, a quantum logspace algorithm can efficiently approximate its inverse or solve linear systems. Girish, Raz, and Zhan [21] described a quantum logspace algorithm to compute powers of an matrix with bounded norm and prove that deterministic logspace is equal to reversible logspace. Recently, it was shown by the same authors that the class of decision problems solvable by a quantum computer in logspace admits an efficient verification procedure [22]; moreover, they also show that every language in BQL has an (information-theoretically secure) streaming proof with a quantum logspace prover and a classical logspace verifier. This hints at a curious interplay between the powers of classical and quantum logspace.

1.2 Catalysis and space

Catalysis is a concept well-studied in the context of quantum information and is widely recognized for its counterintuitive abilities to enable (state) transformations that are otherwise infeasible (see survey by Lipka et al. [30]). A related concept, known as catalytic embedding, was recently introduced in the context of circuit synthesis as an alternative to traditional gate approximation methods in quantum circuit design [4]. Here the goal is to implement a desired unitary operation *more efficiently* (e.g., with fewer gates, lower depth, or using a restricted gate set) than would be possible without assistance. It involves a specific, known, and often small catalyst state that is chosen to aid a particular unitary implementation.

These foregoing lines of work focus on the idea that a specific unitary may be implemented more efficiently if a special state (i.e. catalyst) is available, often discussing resource theories, and do not dwell on complexity theoretic implications.

In this work, we initiate the complexity-theoretic study of the effect of catalytic space in quantum computations. Much like magic state injection is able to promote and increase quantum computational power in the space-unrestricted setting, the presence of a catalyst in the form of an extra register of quantum memory – albeit memory that already contains some stored quantum information – holds a similar promise for space-bounded quantum computations. The notion of catalytic space can be regarded as a theoretical model of qubit reusal.

The first step towards a rigorous study of catalytic logspace quantum computations is to formalize the model and means of interaction with the catalytic space. Identifying new computational capabilities endowed by the presence of a catalyst in the form of additional quantum memory, which however contains an arbitrary unknown quantum state, appears to be a significantly more challenging task due to the nature of quantum information and the inherent limitations of quantum resources. For example, any framework for quantum catalytic space must incorporate the possibility of entanglement and its inherent limitations (e.g. monogamy) between the catalytic memory and the rest of the work space. It has to further account for the irreversibile nature of quantum measurement.

Remarkably, it was recently shown that the addition of a similar notion of catalytic space has major implications even in the classical logspace setting. Buhrman et al. [10] introduced a model of space, called *catalytic computing*, which studies the power of "imperfect" memory. In addition to the usual Turing machine work tape, a catalytic machine is equipped with a much larger *catalytic* work tape, which is filled with an arbitrary initial string τ and which must be reset to the configuration τ at the end of its computation.

The setting of most interest to us is *catalytic logspace* (CL), wherein a logspace machine is given access to a polynomial size catalytic tape. On the positive side, [10] showed that such machines have significantly greater power than traditional logspace, capturing the additional power of both non-determinism (NL) and randomness (BPL); in fact, they showed that CL can simulate the much larger class of *logarithmic-depth threshold circuits* (TC^1). On the negative side, they also showed that CL can be simulated by *(zero-error) randomized polynomial-time machines* (ZPP), which are strongly believed to be much weaker than e.g. polynomial space.

Since then, many works have studied classical catalytic space from a variety of angles, including further results on the power of CL [12, 1, 2] augmenting catalytic machines with other resources such as randomness or non-determinism [11, 15, 12, 29], considering non-uniform models such as catalytic branching programs or catalytic communication complexity [36, 13, 37], analyzing the robustness of classical catalytic machines to alternate conditions [9, 8, 23], and so on. Many properties of catalytic computation have emerged that appear ripe for use in the quantum setting, such as *reversibility* [18, 12], *robustness* [23, 20], and *average-case runtime bounds* [10].

Perhaps most important to motivate our current study, the utility of classical catalytic computation has been strikingly demonstrated in its use as a subroutine in an ordinary space-bounded computation: avoiding linear blowups in space when solving many instances of a problem. The most impactful result is the Tree Evaluation algorithm of Cook and Mertz [14], which was the key piece in Williams' recent breakthrough on time and space [45]. Catalytic subroutines of this kind are even more relevant in the quantum setting, as they may lead to a persistent reduction of the qubit count when executing a quantum algorithm.

1.3 Summary of results

In this paper we initiate the systematic study of catalytic techniques in the quantum setting. To this end we codify a concrete definition of quantum catalytic space ($\mathsf{QCSPACE}$), explore the degrees to which the definition is robust, and establish the relationship of quantum catalytic logspace (QCL) to various classical and quantum complexity classes.

Our main technical contribution is to show that, somewhat surprisingly, quantum Turing machines and quantum circuits are equivalent even in the catalytic space setting:

▶ **Theorem 1.** *Let L be a language, and let $s := s(n)$ and $c := c(n)$. Then L is computable by a quantum catalytic Turing machine with work space $O(s)$ and catalytic space $O(c)$ iff L is computable by a family of quantum catalytic circuits with work space $O(s)$ and catalytic space $O(c)$.*

While this translation is straightforward in other settings, QCL has no *a priori* polynomial time bound, and so there is no obvious way to define the length of a catalyic circuit without running into trouble. However, we prove that the result of Buhrman et al. [10] which shows that CL takes polynomial time on average can be strengthened in the quantum case, to show that QCL *always* takes polynomial time without any error:

▶ **Theorem 2.** $\mathsf{QCL} \subseteq \mathsf{EQP}$

We find Theorem 2 intriguing for many reasons. Naturally it is exciting to be able to solve the "holy grail" of catalytic computing in the quantum setting. The story of classical catalytic computing has been the ability of clever algorithms to circumvent the resetting condition of the catalytic tape and use it for powerful purposes, but Theorem 2 shows that conversely, the additional power of quantum techniques in such algorithms does not offset the additional restrictiveness of resetting a quantum state. Quantum computation is a model fundamentally built on reversible instructions, with the one exception being the final measurement with which we obtain our answer; Theorem 2 shows that this measurement is a massive obstruction to reversibility, as having access to such a huge resource with only the reversible restriction – something which is taken care of in the intermediate computation already – gives less power than we initially assumed.

In terms of class containments, we focus on two questions: the relationship of quantum and classical catalytic space, and the relationship of catalytic space to the one-clean qubit model (DQC_1), a pre-existing object of study in quantum complexity which bears a strong resemblance to catalysis. We show that, while $\mathsf{CL} \subseteq \mathsf{QCL}$ is surprisingly out of reach at the moment, this can be shown for an important subclass of CL, one which captures the strongest known classical containment:

▶ **Theorem 3.** $\mathsf{TC}^1 \subseteq \mathsf{QCL}$

As a consequence, we show that TC^1 constitutes a natural class of functions for which catalysis gives additional power to quantum computation.

We also show that unitary QCL ($\mathsf{Q_UCL}$) and classical CL are both contained in DQC_1:

▶ **Theorem 4.** $\mathsf{BQ_UCL} \subseteq \mathsf{DQC}_1$

▶ **Theorem 5.** $\mathsf{CL} \subseteq \mathsf{DQC}_1$

Note that we use a version of DQC_1 defined using a logspace controller instead of a polynomial time controller as may also be done. These results show how much of the power of DQC_1 comes from avoiding the limitation of the resetting condition on the "dirty" work space.

1.4 Open problems

We identify a number of interesting avenues to further explore the power of quantum catalytic space, and understand its relation to various (quantum) complexity classes.

QCL subroutines

Remarkably, classical catalytic subroutines can already be used to achieve analogous space savings in QCL. Is it possible to identify genuinely quantum subroutines to achieve savings beyond those attained by classical generalizations? This is not so straightforward because the subset of qubits being reused in a catalytic subroutine could become entangled with qubits that cannot be accessed by the subroutine. Therefore, there might be a non-trivial and inaccessible reference system with respect to which the catalytic property must hold. While we show the presence of such an inaccessible reference system does not change the model we define, designing quantum catalytic subroutines (cf. classical results in [14, 45]) stands out as a fertile direction for future work.

QNC^1 vs QCL

Starting with Barrington's Theorem [5], a landmark result in space complexity, a classical line of work [6, 10] has shown that polynomial-size formulas over many different gatesets can be computed using only logarithmic space, using a reversible, algebraic characterization of computation. Such a result in the quantum case, i.e. $\mathsf{QNC}^1 \subseteq \mathsf{QL}$, appears far out of reach, as this would imply e.g. novel derandomizations in polynomial time. However, such techniques are also key to the study of catalytic computation, and so perhaps we can show QNC^1 or a similar quantum circuit class is contained in QCL. This would give a clear indication of the power of quantumness in catalytic computation.

QCL vs DQC_1

While we seem to find that $\mathsf{Q_UCL}$ or QCL without intermediate measurements is contained in DQC_1, it is unclear if this still holds when we allow intermediate measurements.

QCL with errors

One aspect of our results which is discordant with the usual mode of quantum computation is that we require the catalytic tape be *exactly* reset by the computation. On the other hand, many basic primitives in quantum computing, such as converting between gatesets, can introduce errors into the computation, and in practice even the ambient environment can be assumed to cause such issues. Thus it seems natural to study the power of QCL when we allow a small, potentially exponentially small, trace distance between the initial and final catalytic states. This model is well-understood in the classical world [23, 20], but it would be interesting to see whether our techniques can be made robust to this small error or, to the contrary, whether this slight relaxation is enough to overcome the barriers in our work, chiefly the inability to show $\mathsf{CL} \subseteq \mathsf{QCL}$.

2 Preliminaries

2.1 Quantum computation

For this work we will consider complex *Hilbert* spaces $\mathcal{H} \cong \mathbb{C}^d$ of dimension d, that will form the state space for a quantum system. Multiple quantum systems are combined by taking the tensor product of their Hilbert spaces, such as $\mathcal{H}_1 \otimes \mathcal{H}_2$. We will often write $\mathcal{H}_s$ to denote the Hilbert space $\left(\mathbb{C}^2\right)^{\otimes s}$ of s qubits, where the dimension is given by function $d(\mathcal{H}_s) = 2^s$. We will also often use the abbreviation $[n] = \{1, \ldots, n\}$. Below, we recall some of the important background required for this article, referring the reader to [32] for more details.

▶ **Definition 6** (Quantum states). *A* pure quantum states *is a unit vector of the Hilbert space $|\psi\rangle \in \mathcal{H}$, with the normalization condition $\langle\psi|\psi\rangle = 1$. We also make use of more general states represented by* density matrices *ρ which are positive semi definite operators on a Hilbert space with unit trace, $Tr[\rho] = 1$. Density matrices describe* mixed states *which, beyond pure quantum states, can also capture classical uncertainty. In other words, they correspond to classical mixtures of pure quantum states. The density matrix of a pure state is $\rho = |\psi\rangle\langle\psi|$. Given an ensemble of states $\{|\psi_i\rangle\}$ and corresponding probabilities $\{p_i\}$, with $p_i \geq 0$ and $\sum_i p_i = 1$, it can be represented by a mixed state of the form $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. We will denote the set of mixed states a Hilbert space $\mathcal{H}$ by $D(\mathcal{H})$.*

▶ **Definition 7** (Quantum channels). *A* quantum channel *is a linear operator that maps density matrices to density matrices, $\Phi : D(\mathcal{H}_1) \to D(\mathcal{H}_2)$ (also known as superoperators or CPTP maps). It is also required to have two additional properties: 1) it must be completely positive; and 2) it must be trace preserving. We denote the set of channels from $D(\mathcal{H})$ to itself by $\mathcal{C}(D(\mathcal{H}))$.*

We denote the identity channel on d qubits by $\mathcal{I}_d$, or just $\mathcal{I}$ when d is clear from context. The *Choi matrix* of a channel Φ that acts on an input space $\mathcal{H}$ of dimension d is defined by the action of Φ on the first register of a maximally entangled state in $\mathcal{H} \otimes \mathcal{H}$

$$J(\Phi) := (\Phi \otimes \mathcal{I}_d)\left(\frac{1}{d}\sum_{i,j=1}^{d} |i\rangle\langle j| \otimes |i\rangle\langle j|\right) = \frac{1}{d}\sum_{i,j=1}^{d} \Phi\Big(|i\rangle\langle j|\Big) \otimes |i\rangle\langle j|\,.$$

▶ **Definition 8.** *The trace distance between two density matrices $\rho, \sigma \in D(\mathcal{H})$ is defined by:*

$$||\rho - \sigma||_1 = Tr[\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}],$$

where $A^\dagger$ denotes the conjugate transpose of the matrix $A^\dagger = \bar{A}^T$.

It is well known that no physical process can increase the trace distance between two states:

▶ **Lemma 9** (Contractivity under CPTP maps [32, Theorem 9.2]). *Let $\Phi \in \mathcal{C}(D(\mathcal{H}))$ and $\rho, \sigma \in D(\mathcal{H})$ then the trace distance between ρ and σ can not increase under application of Φ:*

$$||\Phi(\rho) - \Phi(\sigma)||_1 \leq ||\rho - \sigma||_1$$

2.1.1 Quantum Turing machines

Our fundamental computation model in quantum computing will be the quantum analogue of Turing machines [17, 7], which we define informally below.

▶ **Definition 10** (Quantum Turing machine). *A* quantum Turing machine *is a classical Turing machine with an additional quantum tape and quantum register. The quantum register does not affect the classical part of the machine in any way, except in that the qubits in the quantum register can be measured in the computational basis. On doing so, the values read from the measurement are copied into the classical registry, from where they can be used to affect the operation of the machine. The quantum Turing machine can perform any gate from its quantum gate set on its quantum registry. We assume this gate set is fixed and universal. Finally, the tape head on the quantum tape can swap qubits between the quantum registry and the position that the quantum tape head is located at. This applies a two-qubit* SWAP *gate.*

We define a number of complexity classes with respect to efficient computation by quantum Turing machines [7, 35][1].

▶ **Definition 11** (BQP). BQP *is the set of all languages $L = (L_{yes}, L_{no}) \subset \{0,1\}^* \times \{0,1\}^*$ for which there exists a quantum Turing machine M using $t = \mathsf{poly}(n)$ time such that for every input $x \in L$ of length $n = |x|$,*

- *if $x \in L_{yes}$ then the probability that M accepts input x is $\geq c$,*
- *if $x \in L_{no}$ then the probability that M accepts input x is $\leq s$.*

▶ **Definition 12** (BQL). BQL *is the set of all languages $L = (L_{yes}, L_{no}) \subset \{0,1\}^* \times \{0,1\}^*$ for which there exists a quantum Turing machine M using $r = O(\log(n))$ quantum and classical space such that for every input $x \in L$ of length $n = |x|$,*

- *if $x \in L_{yes}$ then the probability that M accepts input x is $\geq c$,*
- *if $x \in L_{no}$ then the probability that M accepts input x is $\leq s$.*

The completeness and soundness parameters in both the above definitions can be chosen to be $c = 2/3$ and $s = 1/3$ without affecting the set of languages.

▶ **Definition 13** (EQP). EQP *is the set of all languages $L = (L_{yes}, L_{no}) \subset \{0,1\}^* \times \{0,1\}^*$ for which there exists a quantum Turing machine M using $t = \mathsf{poly}(n)$ time such that for every input $x \in L$ of length $n = |x|$,*

- *if $x \in L_{yes}$ then M outputs one with certainty on measurement,*
- *if $x \in L_{no}$ then M output zero with certainty on measurement.*

▶ Remark 14. Note that the definition of EQP is gateset dependent; this is due to the fact that quantum gatesets only allow universality up to approximation, which means that if a quantum complexity class requires perfect soundness and completeness, as does EQP, it also has to be gateset dependent.

2.1.2 Quantum circuits

We may also define quantum complexity classes using uniform quantum circuits. For this we use similar definitions to those provided by [19], which readers may refer to for more details.

▶ **Definition 15.** *Let $s := s(n), t := t(n), k := k(n)$, let $\mathcal{K}$ be a family of machines, and let $\mathcal{G}$ be a set of k-local operators. A $\mathcal{K}$-uniform space-s time-t family of quantum circuits over $\mathcal{G}$ is a set $\{Q_x\}_{x \in \{0,1\}^n}$, where each Q_x is a sequence of tuples $\langle i, g, j_1 \dots j_k \rangle \in [t] \times \mathcal{G} \times [s]^k$ such that there is a deterministic TM $M \in \mathcal{K}$ which, on input $x \in \mathcal{X}$, outputs a description of Q_x.*

[1] We do not attempt to provide an exhaustive list of references to the vast literature on this topic, and refer the interested reader to the Complexity Zoo for such a list.

The execution of Q_x consists of initializing a vector $|\psi\rangle$ to $|0^s\rangle$ within $\mathcal{H}_s$ and applying, for each step $i \in [t]$ in order, each gate g to qubits $j_1 \dots j_k$ such that $\langle i, g, j_1 \dots j_k\rangle \in Q_x$. The output of Q_x is the value obtained by measuring the first qubit at the end of the computation.

If $\mathcal{G}$ consists of unitary operators, we call these unitary circuits and call each g a gate. If $\mathcal{G}$ additionally consists of measurements together with postprocessing and feed forward by (classical) $\mathcal{K}$-machines, we call these general circuits and call each g a channel.

It is known that polynomial-time uniform general quantum circuits over n qubits with $\mathsf{poly}(n)$ gates can be used to provide an alternative definition of BQP [46]. Similarly, logspace uniform general quantum circuits of logarithmic width can be used as an alternative to define classes such as BQL [19].

2.2 Catalytic computation

We finally recall the known classical definitions of catalytic classical computation.

▶ **Definition 16** ([10]). *A catalytic Turing Machine with space $s := s(n)$ and catalytic space $c := c(n)$ is a Turing Machine M with a work tape of length s and a catalytic tape of length c. We require that for any $\tau \in \{0,1\}^c$, if we initialize the catalytic tape to τ, then on any given input x, the execution of M on x halts with τ on the catalytic tape.*

This definition gives rise to a natural complexity class $\mathsf{CSPACE}[s, c]$, which is a variant of the ordinary class $\mathsf{SPACE}[s]$. The most well-studied variant is *catalytic logspace*, where s is logarithmic and c is polynomial.

▶ **Definition 17.** *We define $\mathsf{CSPACE}[s, c]$ to be the class of all functions f for which there exists a catalytic Turing Machine M with space s and catalytic space c such that on input x, $M(x) = f(x)$. We further define catalytic logspace as*

$$\mathsf{CL} := \bigcup_{k\in\mathbb{N}} \mathsf{CSPACE}(k \log n, n^k)$$

3 Quantum catalytic space

The first goal of this paper is to find a proper definition of quantum catalytic space. There are many choices that have to be made in the model, but we begin with our general definition up front, leaving questions of machine model, uniformity, gateset, and initial catalytic tapes. These will be discussed and clarified in the rest of this section.

▶ **Definition 18** (Quantum catalytic machine). *A quantum catalytic machine with work space $s := s(n)$, catalytic space $c := c(n)$, uniformity $\mathcal{K}$, gateset $\mathcal{G}$, and catalytic set $\mathcal{A}$ is a $\mathcal{K}$-uniform quantum machine M with operations from $\mathcal{G}$ acting on two Hilbert spaces, $\mathcal{H}_s$ and $\mathcal{H}_c$, of dimensions 2^s and 2^c respectively. The latter space, called the catalytic tape, will be initialized to some $\rho \in \mathcal{A} \subseteq D(\mathcal{H}_c)$. We require that for any $\rho \in \mathcal{A}$, if we initialize the catalytic tape to state ρ, then on any given input $x \in \{0,1\}^n$, the execution of $M(x)$ halts with ρ on the catalytic tape. Furthermore, we require that the output state on the worktape is independent of the catalytic state ρ.*[2] *The final action of the machine can be represented by a quantum channel $\Phi_x : |0\rangle\langle 0| \otimes \rho \mapsto \eta_x \otimes \rho$, for any catalytic state ρ and input $x \in \{0,1\}^n$, and some output state η.*

[2] We justify this final requirement in Lemma 42.

This gives rise to the following complexity classes:

▶ **Definition 19** (Quantum catalytic complexity). *$\mathsf{QCSPACE}[s, c]$ is the class of Boolean functions which can be decided with probability 1 by a quantum catalytic machine with work memory s and catalytic memory c.*

$\mathsf{BQCSPACE}[s, c]$ is the class of Boolean functions which can be decided with probability 2/3 by a quantum catalytic machine with work memory s and catalytic memory c.

We further specify to the case of quantum catalytic logspace:

▶ **Definition 20** (Quantum catalytic logspace).

$$\mathsf{QCL} = \bigcup_{k \in \mathbb{N}} \mathsf{QCSPACE}[k \log n, n^k]$$

$$\mathsf{BQCL} = \bigcup_{k \in \mathbb{N}} \mathsf{BQCSPACE}[k \log n, n^k]$$

3.1 Machine model

We begin by defining the two natural choices of base model for quantum catalytic machines, namely *Turing machines* and *circuits*.

▶ **Definition 21** (Quantum catalytic Turing machine). *A quantum catalytic Turing machine is defined as in Definition 18 with quantum Turing machines as our machine model. We write $\mathsf{QCSPACEM}$ (respectively $\mathsf{BQCSPACEM}$, QCLM, and BQCLM) to refer to $\mathsf{QCSPACE}$ with quantum Turing machines.*

▶ **Definition 22** (Quantum catalytic circuits). *A quantum catalytic circuit is defined as in Definition 18 with time-$2^{O(s)}$ quantum circuits as our machine model. We write $\mathsf{QCSPACEC}$ (respectively $\mathsf{BQCSPACEC}$, QCLC, and BQCLC) to refer to $\mathsf{QCSPACE}$ with quantum catalytic circuits.*

Given that CL and related classes are defined in terms of (classical) Turing machines, the option of circuits seems surprising and perhaps unnatural. For example, Definition 22 imposes a time bound as part of its definition, while for CL there is no known containment in polynomial time. For quantum circuits and Turing machines without access to the catalytic tape, a simple equivalence has been known for a long time [46]; however, Definition 22 only allows for circuits of length $2^{O(s)}$, while a generic transformation on $s + c$ qubit registers would give a circuit of length $2^{O(s+c)}$, i.e. requiring an exponential overhead.

The main result of this paper is to show that these models are in fact equivalent:

▶ **Theorem 23.** *For $s = \Omega(\log n), c = 2^{O(s)}$*

$$\mathsf{QCSPACEM}[O(s), O(c)] = \mathsf{QCSPACEC}[O(s), O(c)]$$

$$\mathsf{BQCSPACEM}[O(s), O(c)] = \mathsf{BQCSPACEC}[O(s), O(c)]$$

For the rest of this section we will deal with all auxiliary issues, namely the choice of catalytic tapes and gateset, for quantum circuits alone; while all proofs can be made to hold for quantum Turing machines without much issue, this is also obviated by Theorem 23, which we will prove in Section 4.

3.2 Catalytic tapes

We now move to discussing the choice of initial catalytic tapes $\mathcal{A}$. Perhaps the most immediate choice would be to put no restrictions on $\mathcal{A}$ and allow our catalytic tapes to come from the set of all density matrices in $D(\mathcal{H}_c)$; this will ultimately be our definition.

▶ **Definition 24.** *We fix the catalytic set in Definition 18 to be $\mathcal{A} = D(\mathcal{H}_c)$.*

While this is a natural option, encompassing every possible state on c qubits, there are other choices one can make. We propose four natural options – density matrices and three others – and show that all four are equivalent, thus justifying our choice.

▶ **Definition 25.** *We define the following catalytic sets:*

- Density *is the set of all density matrices $\rho \in D(\mathcal{H}_c)$.*
- Pure *is the set of all pure states $|\psi\rangle \in \mathcal{H}_c$.*
- PauliProd $= \{|\mathrm{PP}\rangle : |\mathrm{PP}\rangle = \bigotimes_{i=1}^{c} |\phi\rangle_i\}$ *is the set of tensor products of eigenstates of the single-qubit Pauli operators, where $|\phi\rangle_i \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\circlearrowright\rangle, |\circlearrowleft\rangle\} \subset \mathcal{H}_2$.*
- EPR $= \{\frac{1}{\sqrt{2^c}} \sum_{i=0}^{2^c-1} |i\rangle |i\rangle\} \subset \mathcal{H}_c \otimes \mathcal{H}_c$ *is the unique state of c EPR pairs, where the catalytic tape will be formed of one half of each EPR pair; the other halves are retained as a reference system which cannot be operated on by the quantum circuit. the quantum circuit is of the form $Q_x = \tilde{Q}_x \otimes \mathcal{I}_c$, acting as the Identity on the second set of halves of the EPR pairs that is inaccessible to the circuit.*

▶ Remark 26. We briefly comment on the fourth set, i.e. EPR. Using classical catalytic techniques as a subroutine has proven to be very useful, for instance in giving an algorithm for tree evaluation in $\mathcal{O}(\log n \log(\log n))$ space [14]. One can also consider using analogous quantum catalytic techniques as subroutines for quantum computations, albeit this does not appear straightforward due to inherent quantum limitations. We will see that this complication can be effectively modeled by considering the initial state of the catalytic tape to be the halves of c EPR pairs.

We will now prove that the four classes of quantum catalytic circuits with initial catalytic states restricted to one of the four sets $D(\mathcal{H}_c)$, $\mathcal{H}_c$, PauliProd, and EPR respectively, are all equivalent. For this we first require the following lemma.

▶ **Lemma 27.** *Any $2^d \times 2^d$ complex matrix can be written as a linear combination of rank-1 outer products of states from* PauliProd *over d qubits.*

In other words, the complex span of the set of d-qubit tensor products of Pauli eigenstates equals the set of $2^d \times 2^d$ complex matrices.

Proof. Note that all four Pauli matrices can be written as a linear combination of two of the Pauli eigenstates:

$$\begin{aligned} I &= |0\rangle\langle 0| + |1\rangle\langle 1|, \quad & X &= |+\rangle\langle +| - |-\rangle\langle -|, \\ Z &= |0\rangle\langle 0| - |1\rangle\langle 1|, \quad & Y &= |\circlearrowright\rangle\langle \circlearrowright| - |\circlearrowleft\rangle\langle \circlearrowleft|. \end{aligned}$$

The four Pauli matrices form a basis for 2×2 complex matrices. Consequently, Pauli strings of length d – i.e., tensor products of d Pauli matrices – form a basis for $2^d \times 2^d$ matrices. ◀

Now we can state the theorem:

▶ **Theorem 28.** *Let* QCC_A *denote quantum catalytic circuits with initial catalytic tapes coming from* A*. Then The following four classes of quantum catalytic circuits are equivalent:*

$$\mathsf{QCC}_{\mathsf{Density}} = \mathsf{QCC}_{\mathsf{Pure}} = \mathsf{QCC}_{\mathsf{PauliProd}} = \mathsf{QCC}_{\mathsf{EPR}}$$

Proof. First note the obvious implications: for any quantum catalytic circuit Φ,

$$\begin{aligned} \Phi \in \mathsf{QCC}_{\mathsf{Density}} &\implies \Phi \in \mathsf{QCC}_{\mathsf{Pure}} \\ \Phi \in \mathsf{QCC}_{\mathsf{Pure}} &\implies \Phi \in \mathsf{QCC}_{\mathsf{PauliProd}} \end{aligned}$$

these follow due to the fact that $\mathsf{PauliProd} \subset \mathsf{Pure} \subset \mathsf{Density}$. To finish the proof, we will further show the following two implications.

$$\begin{aligned} &(1) \quad \Phi \in \mathsf{QCC}_{\mathsf{PauliProd}} \implies \Phi \otimes \mathcal{I}_c \in \mathsf{QCC}_{\mathsf{EPR}} \\ &(2) \quad \Phi \otimes \mathcal{I}_c \in \mathsf{QCC}_{\mathsf{EPR}} \implies \Phi \in \mathsf{QCC}_{\mathsf{Density}} \end{aligned}$$

We first prove implication (1). Let Φ be a circuit from $\mathsf{QCC}_{\mathsf{PauliProd}}$ and consider the action of $\Phi \otimes \mathcal{I}_c$ (where the Identity operator acts on the inaccessible halves of the EPR pairs) on the state $\frac{1}{2^c} |0\rangle \langle 0| \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j|$:

$$\Phi \otimes \mathcal{I}_c \left(\frac{1}{2^c} |0\rangle \langle 0| \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j| \right) = \frac{1}{2^c} \sum_{i,j} \Phi\Big(|0\rangle \langle 0| \otimes |i\rangle \langle j| \Big) \otimes |i\rangle \langle j|,$$

because Φ being a channel is a linear operator. By Lemma 27, $|i\rangle \langle j|$ can be written as a linear combination of rank-1 projectors onto $\mathsf{PauliProd}$ states. Since Φ is catalytic with respect to $\mathsf{PauliProd}$, it follows that

$$\frac{1}{2^c} \sum_{i,j} \Phi\Big(|0\rangle \langle 0| \otimes |i\rangle \langle j| \Big) \otimes |i\rangle \langle j| = \eta \otimes \frac{1}{2^c} \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j|,$$

for some state in $\eta \in D(\mathcal{H}_s)$. This shows that $\Phi \in \mathsf{QCC}_{\mathsf{EPR}}$.

Implication (2) requires a similar approach. Let $\tilde{\Phi} \in \mathsf{QCC}_{\mathsf{EPR}}$, then we can write $\tilde{\Phi} = \Phi \otimes \mathcal{I}_c$. For a given input state $|0\rangle \langle 0| \in \mathcal{H}_s$ the action of $\Phi \otimes \mathcal{I}_c$ must satisfy

$$\Phi \left(\frac{1}{2^c} \sum_{i,j} |0\rangle \langle 0| \otimes |i\rangle \langle j| \right) \otimes |i\rangle \langle j| = \eta \otimes \frac{1}{2^c} \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j|,$$

for some state in $\eta \in D(\mathcal{H}_s)$. Since the catalytic state of c EPR pairs is returned perfectly unaffected for every choice of input state, the effective channel of Φ can also be written as a tensor product channel: $\Phi = \Gamma_s \otimes \Xi_c$[3], with the action of Ξ_c being

$$\frac{1}{2^c} \sum_{i,j} \Xi_c \Big(|i\rangle \langle j| \Big) \otimes |i\rangle \langle j| = \frac{1}{2^c} \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j| \ .$$

[3] It seems that the catalyst does not offer any improvement, because we can write Φ as a tensor product of the action on the logspace clean qubits and the action of the catalyst, however this does not need to hold. Only the action as a whole is writable as a tensor product, it might actually consist of intermediate steps that are not of tensor product form, therefor Γ_s might only have an efficient circuit description in the presence of a catalyst.

Note that although the effective channel factorises into a tensor product across the work and catalytic registers, without the catalytic tape much larger circuits may be required to implement Γ_c. Moving forward, this implies that the Choi matrix of Ξ_c is

$$J(\Xi_c) = \sum_{i,j} \Xi_c\Big(|i\rangle\langle j|\Big) \otimes |i\rangle\langle j| = \sum_{i,j} |i\rangle\langle j| \otimes |i\rangle\langle j| = J(\mathcal{I}),$$

and therefore the effective channel Ξ_c is the identity channel. This gives that for any state $\rho \in \mathcal{H}_c$ it must hold that on input $|0\rangle\langle 0|$, the channel Φ must act as follows:

$$\Phi(|0\rangle\langle 0| \otimes \rho) = \eta \otimes \rho \qquad ◀$$

▶ Remark 29. In the proof that these channel definitions are equivalent we actually showed that any channel under one definition also furnishes an instance of the other definitions. This means that they are also operationally equivalent. These equivalence proofs therefore have to hold for any type of machine model that has to adhere to the same restrictions of resetting the input state in the catalytic space. In particular it also holds for quantum Turing machines.

3.3 Gateset

When discussing quantum circuits, a fundamental issue is the underlying gate set. Unlike the classical case, unitary operations form a continuous space, and finite-sized circuits over finite gate sets cannot implement arbitrary unitaries. However, there do exist finite gate sets of constant locality (that is, fan-in) which are quantum universal, in the sense that any n-qubit unitary may be approximated to any desired precision ϵ in ℓ_2-distance by a product of $l = O(\mathsf{poly}\log\frac{1}{\epsilon})$ gates from the universal gate set; this is the celebrated Solovay-Kitaev theorem [27, 16, 32]. From the standpoint of complexity classes, Nishimura and Ozawa [34] also showed that polynomial-time quantum Turing machines are exactly equivalent to finitely generated uniform quantum circuits.

We note that Definitions 15 and 22 do not make reference any fixed universal gate set. A potential issue that arises in this regard is that the complexity class being defined may depend in an intricate way on the chosen universal gate set, since it may not be possible to perfectly reset every initial catalytic state under our uniformity and resource constraints. If we relax the notion of catalyticity to mean that the initial catalytic state only has to be reset to within ϵ trace distance at the end of the computation, one can use the Solovay-Kitaev theorem to see that every choice of gate set leads to the same complexity class in Definition 22. This interesting model resembles classical catalytic space classes with small errors in resetting, and we leave it as an open question to determine how it relates to the exact resetting model.

Returning to our setting that requires the quantum catalytic machine to perfectly reset the catalytic space to its initial state at the end of the computation, we will restrict out attention to the case of universal quantum gate sets that are infinite (for the complexity theoretic properties of circuit families over such gate sets, see e.g. [33]). In this case, our definition is robust to the choice of gate set since any unitary may be implemented exactly by finite-sized circuits over such gate sets. Consequently changing the gate set does not change the set of catalytic states that can be reset exactly by the machine. This results in well-defined catalytic complexity classes independent of the specific choice of gate set.

3.4 Uniformity

Similar to gatesets, the question of uniformity is quite subjective, as different uniformity conditions will lead to different levels of expressiveness for our machines.

▶ **Definition 30.** *We fix the uniformity in Definition 18 to be $\mathcal{K} = \mathsf{SPACE}[O(s)]$.*

We choose $\mathsf{SPACE}[O(s)]$ as it is the largest class of classical machines a $\mathsf{QCSPACE}[s, c]$ machine should seemingly contain by default. Thus we believe the choice of $\mathsf{SPACE}[O(s)]$-uniformity is best suited to removing classical uniformity considerations from taking the forefront of the discussion regarding quantum catalytic space.

The question of how uniformity affects the power of QCSPACE is left to future work; we only comment briefly here on natural alternative choices. Perhaps the most immediate would be to consider $\mathsf{CSPACE}[s, c]$ uniformity, as it mirrors our quantum machine. As we will see later, it is not clear how to prove $\mathsf{QCSPACE}[s, c]$ contains $\mathsf{CSPACE}[s, c]$ directly, an interesting technical challenge that would be rendered moot by building it into the uniformity. Similarly we avoid P-uniformity because it is not known, and even strongly disbelieved, that CL contains P.

4 QCL upper bounds

In this section we will finally return to the question of our quantum machine model, showing that Turing machines and circuits are equivalent. One major stepping stone is to show that quantum catalytic Turing machines adhere to a polynomial runtime bound for *all* possible initializations of the catalytic tape.

Before all else, a remark is in order as to why such a restriction should hold for a seemingly stronger model, i.e. QCLM, when it is not in fact known for CL. While quantum catalytic space has access to more powerful computations, i.e. quantum operations, it also has the much stronger restriction of resetting arbitrary density matrices rather than arbitrary bit strings. This restriction gives rise to a much stronger upper bound argument, and in fact rules out one of the main techniques available to classical Turing machines, namely compression arguments (see c.f. [18, 12]).

4.1 Polynomial average runtime bound

We begin by showing an analogue of the classical result of [10], i.e. the average runtime of a quantum catalytic machine for a random initial catalytic state ρ is polynomial in the number of work qubits. We note that the runtime of a quantum Turing machine need not be a deterministic function of the input; M has access to quantum states and intermediate measurements, from which it is possible to generate randomness which might influence the time that machine takes to halt.

▶ **Definition 31.** *Given a quantum catalytic Turing machine M, a fixed input $x \in \{0, 1\}^n$, and an initial catalytic tape ρ, we denote by $T(M, x, \rho)$ the distribution of runtimes of M on input x and initial catalytic tape ρ.*

For an averaging argument to hold, we need to have a quantum notion of non-overlapping configuration graphs.

▶ **Lemma 32.** *Let M be a quantum catalytic Turing machine, and let $\{\tau_i\}_i$ form an orthonormal basis for $D(\mathcal{H}_c)$. For all i and t, let $\rho_{i,t}$ be the density matrix describing the state of the classical tape, quantum tape, and internal state of M at time step t on initial catalytic tape τ_i. Then if M is absolutely halting, all elements of the set $\{\rho_{i,t}\}_{i,t}$ are orthogonal.*

Proof. We first consider the states $\rho_{i,t}$ for a fixed i. Assume instead that there exists some times t and t' where the states are not orthogonal. This means that the state at time step t can be written as a superposition between the state in time step t' and the state

$\rho_{i,t} = p\rho_{i,t'} + (1-p)\eta$ for some $p > 0$. This forms a loop in the configuration graph where part of the state is back at time step t'. The amplitude of the part of the state in this loop will shrink over time, but never go to zero. The part of the state that is stuck in the loop will never reach the halting state, therefore this is in contradiction with the assumption that the quantum Turing machine is absolutely halting.

Next we consider the states $\rho_{i,t}$ for different i. By definition of a quantum Turing machine, the transformations M can apply to the entire state of the machine is given by some quantum channel. By Lemma 9 we know that the trace distance between the entire state of the machine for separate instances of the catalytic tape can only decrease by this quantum channel. Therefore we know that if two instances start out to be orthogonal and end to be orthogonal, they have to remain orthogonal through the entire calculation. ◀

▶ **Lemma 33.** *Let M be a quantum catalytic Turing machine with work space s and catalytic space c, let $\{\rho_i\}_i$ form an orthonormal basis for $D(\mathcal{H}_c)$, and define $T_{max}(M, x, \rho)$ to be the maximum runtime of machine M on input x on starting catalytic tape ρ. Then*

$$\mathbb{E}_i[T_{max}(M, x, \rho_i)] \leq 2^{O(s)}$$

Proof. Our catalytic machine is defined by a $\mathsf{SPACE}[O(s)]$ machine, defined by a tape of length $O(s)$ and an internal machine of size $O(1)$, which acts on $\mathcal{H}_s$ and $\mathcal{H}_c$, which can be addressed into using $\log s$ and $\log c$ bits respectively. Since these quantities plus the Hilbert spaces $\mathcal{H}_s$ and $\mathcal{H}_c$ define the dimensionality of our machine, by Lemma 32 we have that

$$\sum_{\rho\in\{\rho_i\}} T_{max}(M, x, \rho) \leq \mathcal{O}(2^{2^{2(s+c+O(s)+O(1)+\log s+\log c)}})$$

and therefore the lemma follows because $|\{\rho_i\}| \leq 2^{2c}$ and $2(s + O(s) + O(1) + \log s + \log c) = O(s)$. ◀

This already gives us a nice containment for our $\mathsf{QCSPACE}[s, c]$ classes.

▶ **Corollary 34.** $\mathsf{QCLM} \subseteq \mathsf{ZQP}$

▶ **Corollary 35.** $\mathsf{BQCLM} \subseteq \mathsf{BQP}$

4.2 Equal running times

We now take a further leap, showing that the initial catalytic tape does not affect the (distribution of the) runtime of our machine M for a fixed input x.

We can first show that given M and only one single copy of a state $\eta \in \mathcal{H}_c$, this probability distribution can be approximated up to arbitrary precision for any x.

▶ **Lemma 36.** *Given catalytic Turing machine M and a single copy of a quantum state $\eta \in \mathcal{H}_c$, $T(M, x, \eta)$ can be approximated up to arbitrary precision for any x.*

Proof. Because M is a quantum catalytic Turing machine it has to reset the quantum state initialized in its catalytic tape perfectly. Therefore we can use the following approach: first fix some input x, then run the catalytic machine given x as input and η on its catalytic tape and record the running time. When the machine halts, η should be returned in the catalytic tape. This means the test can be performed again given the same inputs. This test can be run arbitrarily often giving an arbitrary approximation to $T(M, x, \eta)$. ◀

This gives us the following observation about states with different halting times:

▶ **Lemma 37.** *Let M be a quantum catalytic Turing machine, and let $\rho_1, \rho_2 \in D(\mathcal{H}_c)$. Assume there exists $x \in \{0,1\}^n$ such that $T(M, x, \rho_1) \neq T(M, x, \rho_2)$. Then $||\rho_1 - \rho_2||_1 = 1$, where $|| \cdot ||_1$ is the trace distance.*

Proof. The Helstrom bound states that the optimal success probability of any state discrimination protocol given one copy of an unnown state is:

$$P_{success} = \frac{1}{2} + \frac{1}{2} \cdot ||\rho_1 - \rho_2||_1$$

By Lemma 36, we know that $T(M, x, \rho)$ can be approximated to any precision with only one copy of ρ. Given a copy of either ρ_1 or ρ_2 at random, one can estimate $T(M, x, \rho)$ and perfectly discriminate between the cases $\rho = \rho_1$ and $\rho = \rho_2$ giving a protocol with $P_{success} = 1$. Therefore it follows that

$$\frac{1}{2} + \frac{1}{2}||\rho_1 - \rho_2||_1 = 1$$

and hence $||\rho_1 - \rho_2||_1 = 1$. ◀

Lemma 37 is sufficient to show that the halting time of a quantum catalytic Turing machine is independent of the initial state in the catalytic tape:

▶ **Theorem 38.** *Let M be a quantum catalytic Turing machine with s-qubit work space and c-qubit catalytic space, and let $x \in \{0,1\}^n$. Then there exists some value $t := t(n)$ such that $T(M, x, \rho) = t$ for all $\rho \in D(\mathcal{H}_c)$.*

Proof. Assume for contradiction that there exist ρ_1, ρ_2 such that $T(M, x, \rho_1) \neq T(M, x, \rho_2)$. By Lemma 37 it holds that $||\rho_1 - \rho_2||_1 = 1$. Consider the state $\rho' = \frac{1}{2}\rho_1 + \frac{1}{2}\rho_2$, and note that only one of $T(M, x, \rho') = T(M, x, \rho_1)$ or $T(M, x, \rho') = T(M, x, \rho_2)$ can hold, by transitivity. Without loss of generality, let us assume $T(M, x, \rho') = T(M, x, \rho_2)$, thereby $T(M, x, \rho') \neq T(M, x, \rho_1)$ and so $||\rho' - \rho_1||_1 = 1$ by Lemma 37. However, by definition we have that

$$||\rho' - \rho_1||_1 = ||(\frac{1}{2}\rho_1 + \frac{1}{2}\rho_2) - \rho_1||_1 = \frac{1}{2}$$

which is a contradiction. ◀

Putting Lemma 33 and Theorem 38 together immediately shows that the runtime of M is bounded by a polynomial in n for every input x and initial catalytic state ρ:

▶ **Theorem 39.** *Let M be a quantum catalytic Turing machine with work space s and catalytic space c. Then the maximum halting time is bounded by $2^{\mathcal{O}(s)}$.*

This strengthens Corollary 34 to remove the randomness in the output probability; this is the quantum equivalent of showing CL ∈ P, considered the holy grail of open problems in classical catalytic computing:

▶ **Corollary 40.** QCLM ⊆ EQP

4.3 Turing machines and circuits

We finally prove Theorem 23 and show the equivalence of our two definitions of quantum catalytic machines. To do this, we observe, without proof, that Theorem 38 extends to any *classical observable feature* of the initial catalytic state by the same proof. We will apply this to one other aspect, namely the transition applied at a given timestep t:

▶ **Lemma 41.** *Let M be a quantum catalytic Turing machine, and let $x \in \{0,1\}^n$. Then for every time t, there exists a fixed operation g applied by M at time t for every $\rho \in \mathcal{H}_c$.*

This is sufficient to prove Theorem 23:

Proof of Theorem 23. We only prove the equivalence between QCSPACEC and QCSPACEM; the same proof applies to BQCSPACEC and BQCSPACEM. Certainly QCSPACEC$[s,c]$ is contained in QCSPACEM$[O(s),O(c)]$, since QCSPACEC circuits are SPACE$[O(s)]$ uniform and can be directly simulated by a QCSPACEM machine.

Conversely, given a QCSPACEM$[s,c]$ machine M, we wish to find an equivalent quantum catalytic circuit in QCSPACEC$[O(s),O(c)]$. For this, we transform the transition function of the quantum Turing machine into a quantum channel; since the transition only takes a finite number of (qu)bits as input, this can be always be done, and we have our transitions act on the same space $\mathcal{H}_s \otimes \mathcal{H}_c$ as M. Then, by using a method similar to that from the proof of Lemma 56, to make the machine oblivious, the tape head movement of the quantum Turing machine will be fixed. If our circuit is the transition function channel copied to all locations where the tape heads end up, we completely simulate the quantum Turing machine. We know that $T_{max}(M,x,\rho)$ is always at most $2^{O(s)}$ for a machine M by Theorem 39, and so the number of such transition function channels is also at most $2^{O(s)}$. Therefore, we can simulate M using a quantum circuit of length $2^{O(s)}$ as claimed. ◀

As an afterword, we also resolve one other aspect of our initial definition of quantum catalytic space, namely the requirement that the output state be the same for every initial catalytic state. As mentioned above, Lemma 36 extends to all classically observable characteristics, but a similar argument clearly holds for approximating the output state as well:

▶ **Lemma 42.** *Given catalytic Turing machine M and a single copy of a quantum state $\eta \in \mathcal{H}_c$, the output qubit $|\phi_{out}\rangle$ can be approximated up to arbitrary precision for any x.*

Thus again we can appeal to the inststinguishability of nearby catalytic states to claim that $|\phi_{out}\rangle$ must be equal for all inital $|\tau\rangle$.

References

1 Aryan Agarwala and Ian Mertz. Bipartite matching is in catalytic logspace. *Electron. Colloquium Comput. Complex.*, TR25-048, 2025. URL: https://eccc.weizmann.ac.il/report/2025/048/.

2 Yaroslav Alekseev, Yuval Filmus, Ian Mertz, Alexander Smal, and Antoine Vinciguerra. Catalytic computing and register programs beyond log-depth. *Electron. Colloquium Comput. Complex.*, TR25-055, 2025. URL: https://eccc.weizmann.ac.il/report/2025/055/.

3 Noga Alon. Problems and results in extremal combinatorics—i. *Discrete Mathematics*, 273(1):31–53, 2003. EuroComb'01. doi:10.1016/S0012-365X(03)00227-9.

4 Matthew Amy, Matthew Crawford, Andrew N Glaudell, Melissa L Macasieb, Samuel S Mendelson, and Neil J Ross. Catalytic embeddings of quantum circuits. *arXiv preprint arXiv:2305.07720*, 2023.

5 David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences (J.CSS)*, 38(1):150–164, 1989. doi:10.1016/0022-0000(89)90037-8.

6 Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing (SICOMP)*, 21(1):54–58, 1992. doi:10.1137/0221006.

7 Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/s0097539796300921.

8 Sagar Bisoyi, Krishnamoorthy Dinesh, Bhabya Rai, and Jayalal Sarma. Almost-catalytic computation. *CoRR*, abs/2409.07208, 2024. doi:10.48550/arXiv.2409.07208.

9 Sagar Bisoyi, Krishnamoorthy Dinesh, and Jayalal Sarma. On pure space vs catalytic space. *Theor. Comput. Sci.*, 921:112–126, 2022. doi:10.1016/J.TCS.2022.04.005.

10 Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: Catalytic space. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 857–866, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591874.

11 Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: Nondeterminism and hierarchy. *Theory Comput. Syst.*, 62(1):116–135, 2018. doi:10.1007/S00224-017-9784-7.

12 James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. The structure of catalytic space: Capturing randomness and time via compression. In *ACM Symposium on Theory of Computing (STOC)*, 2025.

13 James Cook and Ian Mertz. Trading time and space in catalytic branching programs. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 8:1–8:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.CCC.2022.8.

14 James Cook and Ian Mertz. Tree evaluation is in space $O(\log n \cdot \log\log n)$. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1268–1278, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649664.

15 Samir Datta, Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Randomized and symmetric catalytic computation. *Electron. Colloquium Comput. Complex.*, TR20-024, 2020. URL: https://eccc.weizmann.ac.il/report/2020/024.

16 Christopher M. Dawson and Michael A. Nielsen. The solovay-kitaev algorithm. *Quantum Info. Comput.*, 6(1):81–95, January 2006.

17 David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

18 Yfke Dulek. Catalytic space: on reversibility and multiple-access randomness. 2015.

19 Bill Fefferman and Zachary Remscrim. Eliminating intermediate measurements in space-bounded quantum computation. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1343–1356, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451051.

20 Marten Folkertsma, Ian Mertz, Florian Speelman, and Quinten Tupker. Fully characterizing lossy catalytic computation. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference, ITCS 2025, January 7-10, 2025, Columbia University, New York, NY, USA*, volume 325 of *LIPIcs*, pages 50:1–50:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.ITCS.2025.50.

21 Uma Girish, Ran Raz, and Wei Zhan. Quantum logspace algorithm for powering matrices with bounded norm. *arXiv preprint arXiv:2006.04880*, 2020.

22 Uma Girish, Ran Raz, and Wei Zhan. Quantum logspace computations are verifiable. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 144–150. SIAM, 2024.

23 Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Lossy catalytic computation. *Computing Research Repository (CoRR)*, abs/2408.14670, 2024.

24 Dustin G. Mixon (https://mathoverflow.net/users/29873/dustin-g mixon). How many non-orthogonal vectors fit into a complex vector space? MathOverflow. URL:https://mathoverflow.net/q/458508 (version: 2023-11-16). arXiv:https://mathoverflow.net/q/458508.

25 Neil Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on computing*, 17(5):935–938, 1988.

26 Richard Jozsa, Barbara Kraus, Akimasa Miyake, and John Watrous. Matchgate and space-bounded quantum computations are equivalent. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2115):809–830, 2010.

27 A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, December 1997. doi:10.1070/rm1997v052n06abeh002155.

28 E. Knill and R. Laflamme. Power of one bit of quantum information. *Physical Review Letters*, 81(25):5672–5675, December 1998. doi:10.1103/physrevlett.81.5672.

29 Michal Koucký, Ian Mertz, Ted Pyne, and Sasha Sami. Collapsing catalytic classes. *Electronic Colloquium on Computational Complexity (ECCC)*, TR25-018, 2025. URL: https://eccc.weizmann.ac.il/report/2025/018.

30 Patryk Lipka-Bartosik, Henrik Wilming, and Nelly HY Ng. Catalysis in quantum information theory. *Reviews of Modern Physics*, 96(2):025005, 2024.

31 Ian Mertz. Reusing space: Techniques and open problems. *Bulletin of the EATCS (B.EATCS)*, 141:57–106, 2023.

32 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2010. doi:10.1017/CBO9780511976667.

33 Harumichi Nishimura and Masanao Ozawa. Computational complexity of uniform quantum circuit families and quantum turing machines. *Theor. Comput. Sci.*, 276(1–2):147–181, April 2002. doi:10.1016/S0304-3975(01)00111-6.

34 Harumichi Nishimura and Masanao Ozawa. Perfect computational equivalence between quantum turing machines and finitely generated uniform quantum circuit families. *Quantum Information Processing*, 8(1):13–24, January 2009. doi:10.1007/s11128-008-0091-8.

35 Tetsuro Nishino. Mathematical models of quantum computation. *New Generation Computing*, 20(4):317–337, December 2002. doi:10.1007/bf03037370.

36 Aaron Potechin. A note on amortized branching program complexity. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 4:1–4:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.CCC.2017.4.

37 Edward Pyne, Nathan S. Sheffield, and William Wang. Catalytic communication. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference, ITCS 2025, January 7-10, 2025, Columbia University, New York, NY, USA*, volume 325 of *LIPIcs*, pages 79:1–79:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.ITCS.2025.79.

38 Walter J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970.

39 Dan Shepherd. Computation with unitaries and one pure qubit, 2006. arXiv:quant-ph/0608132.

40 Peter W. Shor and Stephen P. Jordan. Estimating jones polynomials is a complete problem for one clean qubit, 2008. arXiv:0707.2831.

41 Róbert Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta informatica*, 26:279–284, 1988.

42 Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 881–890, 2013.

43 John Watrous. Quantum algorithms for solvable groups. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 60–67, 2001.

44 John Harrison Watrous. *Space-bounded quantum computation*. The University of Wisconsin-Madison, 1998.

45 Ryan Williams. Simulating time in square-root space. *Electron. Colloquium Comput. Complex.*, TR25-017, 2025. URL: https://eccc.weizmann.ac.il/report/2025/017/.

46 Andrew Chi-Chih Yao. Quantum circuit complexity. In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 352–361. IEEE Computer Society, 1993. doi:10.1109/SFCS.1993.366852.

A Simulation of TC^1

In this section we show that QCL can simulate Boolean threshold circuits. As in the classical world, the ability to simulate TC^1 is also a reason to believe that catalytic logspace is strictly more powerful than logspace. This follows from the fact that $\mathsf{QL} = \mathsf{PL}$ [44], which is itself contained in TC^1:

▶ **Lemma 43.** $\mathsf{QL} \subseteq \mathsf{TC}^1$

Since TC^1 can compute powerful functions such as determinant, this containment is largely believed to be strict. Thus Theorem 3 gives us a candidate class of problems for separating QL from QCL.

A.1 Reversibility and obliviousness

In [10] the authors showed that TC^1 can be simulated by *transparent register programs*, which themselves are computable in CL; thus our goal is to extend the CL simulation of transparent programs to QCL. More broadly, we show that *reversible, oblivious, time-bounded* CL is enough to simulate transparent programs, and such a model is structured enough that, while we cannot show that all of CL is in QCL, we can at least prove the containment for this small fragment.

We first make the following definitions which we use for our simulations. We begin by recalling a result of Dulek [18] which shows that catalytic Turing machines can be made *reversible* (see c.f. [12] for a proof)

▶ **Theorem 44.** *For every catalytic machine M with space s and catalytic space c, there exist catalytic machines $M_{\rightarrow}$, $M_{\leftarrow}$ with space $s+1$ and catalytic space c such that for any pair of configurations (τ_1, v_1), (τ_2, v_2) of $M_{\rightarrow}$ and $M_{\leftarrow}$, if $M_{\rightarrow}$ transitions from (τ_1, v_1) to (τ_2, v_2) on input x, then $M_{\leftarrow}$ transitions from (τ_2, v_2) to (τ_1, v_1) on input x.*

We will also need to consider *oblivious* machines, i.e. ones where the tape head movement is solely a function of the input length $|x|$ and does not depend at all on the content of the catalytic tape c. While any Turing machine can be made oblivious, it requires relaxing the definition of obliviousness to not forcing the machine to halt at the same time on every input; we simply require that every machine that continues to run carries out its execution in an oblivious manner. We will bar this restriction in this section.

▶ **Definition 45.** *We say that a CL machine is totally oblivious if the following holds. Let t, q, h be special registers on the free work tape, all initialized to 0, representing the time, state, and tape heads of the machine. At each point in time our machine consist of one mega-step: for every setting of t, q, h there is a fixed transformation, computable in logspace, which the machine applies to the catalytic tape and to q, h, and a mega-step consists of applying each of these operations, conditioned on the values of t, q, h on the free work tape, in order. At the end of every mega-step we increment t, and our machine halts iff t reaches a predetermined step T.*

Totally oblivious machines are ones that in essence apply the same bundle of transformations at every time step, with the information about which one to to actually apply being written on the free work tape, and the halting behavior being determined only by the clock.

Such machines are clearly in poly-time bounded CL (see c.f. [12] for a discussion of this class), since the clock must fit on the free work tape. This causes issues when we seek total obliviousness in tandem with reversibility; in general it is not known, and is highly unlikely, that a polynomially time-bounded Turing machine can be made reversible while remaining polynomially time-bounded.

However, there is an important class of algorithms which is both reversible and totally oblivious: *clean register programs.* For our purposes we will use a very restricted version of clean register programs (see c.f. [31] for a discussion).

▶ **Definition 46.** *A register program $\mathcal{P}$ is a list of instructions $P_1 \dots P_t$ where each P_i either has the form $R_j \mathrel{+}= x_k$ for some input variable x_k or has the form $R_j \mathrel{+}= q_i(R_1 \dots R_m)$ for some polynomial q_i. A register program cleanly computes a value v if for any initial values $\tau_1 \dots \tau_m$, the net result of running $\mathcal{P}$ on the registers $R_1 \dots R_m$, where each R_j is initialized to the value τ_j, is that $R_1 = \tau_1 + v$ and $R_j = \tau_j$ for all $j \neq 1$.*

If we think of these registers as being written on the catalytic tape, it is clear that clean register programs are totally oblivious, as the instruction at every moment in time is based only on the timestep. This is nearly immediate, although we note a few minor complications here. We need to preprocess the catalytic tape to ensure our registers have values over the same ring as our register program; for example, if we represent numbers mod p using $\lceil \log p \rceil$ bits, some initial values will exceed p. This can be handled obliviously by observing that for either τ or $\overline{\tau}$, half the registers are already correct, and so we take one full pass over τ to keep a count of which case we are in, store this as a bit b (1 iff we need to flip τ), and XOR τ with b at the beginning and end of the computation. We subsequently ignore all blocks which are initialized to improper values; when we go to operate on register R_j, say, as we obliviously pass over the whole catalytic tape we will count how many *valid* registers we have seen, and act only when we see the counter reach j.

Besides being totally oblivious, however, such programs are also *reversible*, as every step of the form $R_j \mathrel{+}= c$ can be inverted by a step of the form $R_j - = c$. Thus such programs appear highly constrained in terms of what they can and cannot achieve. Nevertheless, such programs are sufficient to compute TC^1.

▶ **Lemma 47** ([10]). *Let L be a language in TC^1. Then L can be decided by a clean register program, and, hence, by a totally oblivious reversible* CL *machine.*

A.2 Simulation by QCL machines

We now show that reversibility plus total obliviousness is sufficient for simulation by QCL.

▶ **Lemma 48.** *Let L be a language which can be computed be a totally oblivious reversible* CL *machine. Then $L \subseteq$* QCL*.*

Proof. Let M be a totally oblivious reversible CL machine. We will treat our quantum catalytic tape as a superposition over classical catalytic tapes, i.e. a superposition over computational basis states. It is thus sufficient to show that the operation of machine M can be simulated by a fixed quantum circuit containing Toffoli gates, as such a circuit will correctly operate on each of our catalytic basis states in each branch of the superposition.

By total obliviousness, every step that M takes is a fixed transformation conditioned on the value of t, q, and h; since we additionally know that such a step is reversible, it must be isomorphic to a Toffoli gate applied to a fixed position of the catalytic tape conditioned on some fixed mask applied to t, q, and h, and furthermore each transformation can be computed by our logspace controlling machine. Since these operations are fixed for each timestep, we can move t to our space controlling machine and have it construct a circuit, comprised of Toffoli gates on q, h, and the catalytic tape, of polynomial length. ◀

This is sufficient to prove our main result for this section:

Proof of Theorem 3. Combine Lemma 47 with Lemma 48. ◀

B Simulating catalytic space in DQC$_1$

Lastly we will discuss the relationship between catalytic computing and a pre-existing yet closely related quantum model, namely the one clean qubit setting. We will introduce the model and then prove that it can simulate unitary QCL. In the full version of the paper, we further show that classical CL is also contained in the one clean qubit model.

B.1 One clean qubit model

In the one-clean qubit model, first introduced by Knill and Laflamme [28], a quantum machine is given a single input qubit initialized in the zero state and n qubits initialized in the maximally mixed state. We will formalize the definition of this computational model:

▶ **Definition 49** (One clean qubit). *Let $\{Q_x\}_x$ be a log-space uniform family of unitary quantum circuits. The* one clean qubit model *is a model of computation in which Q_x is applied to the $n+1$-qubit input state*

$$\rho = |0\rangle\langle 0| \otimes \frac{I_n}{2^n},$$

where $n = |x|$ and I_n operator is the identity on n qubits. After execution of Q_x the first qubit is measured, giving output probabilities:

$$p_0 = 2^{-n}\,\mathrm{Tr}\big[(|0\rangle\langle 0| \otimes I)Q_x(|0\rangle\langle 0| \otimes I)Q_x^\dagger\big],$$
$$p_1 = 1 - p_0$$

▶ Remark 50. Two points stand out in this definition. First, note that Q_x are unitary circuits, and hence do not allow intermediate measurements; such measurements would allow for resetting the qubits initialized in the maximally mixed state, making the model significantly stronger. Second, in this paper we consider log-space uniform families of unitary circuits, rather than the more common deterministic polynomial-time uniform families, in order to align more closely with the QCL model that we study.

The one-clean qubit model is a probabilistic model of computation, and hence we typically talk about computing a function $f(x)$ in terms of success probability for computing $f(x)$ being bounded away from $1/2$. The exact bound on the error probability does not matter; while we often use $2/3$ in defining e.g. BQP, even a $1/\mathsf{poly}(n)$ gap is sufficient as there we can employ standard error-correction to boost our success, namely by running the algorithm multiple times. However, this is not known to be possible in the one-clean qubit model, as such a machine can only reliably run once.

▶ **Definition 51** ([28, 39]). DQC$_1$ *is the set of all languages $L = (L_{yes}, L_{no}) \subset \{0,1\}^* \times \{0,1\}^*$ for which there exists a one-clean qubit machine M and a polynomial $q(n)$ that on input $x \in L$ of length $n = |x|$,*

- *if $x \in L_{yes}$ then the output probability $p_1 \geq \frac{1}{2} + \frac{1}{q(n)}$*
- *if $x \in L_{no}$ then the output probability $p_0 \geq \frac{1}{2} + \frac{1}{q(n)}$*

On the other hand, somewhat surprisingly the one-clean qubit model is robust to the number of clean qubits allowed, up to a logarithmic number:

▶ **Lemma 52** ([40]). *$\mathsf{DQC}_k = \mathsf{DQC}_1$ for $k = \mathcal{O}(\log(n))$, where DQC_k means having access to k clean qubits instead of one.*

B.2 Containment of unitary QCL in DQC$_1$

We now move on to establishing a formal connection between QCL and DQC$_1$. A QCL machine is allowed to apply intermediate measurements to its quantum tape as well as its catalytic tape, which is not possible in DQC$_1$; however, if we restrict the QCL machine to not make any intermediate measurements we can show that such a machine can in fact be simulated by the one-clean qubit model.

▶ **Definition 53** (Q$_\mathsf{U}$CL). *A Q$_\mathsf{U}$CL machine is a QCL machine in which the quantum circuit is unitary. In the final step of the unitary the Q$_\mathsf{U}$CL machine measures the first qubit, which then gives the outcome of the calculation. Similarly we define BQ$_\mathsf{U}$CL to be BQCL with the unitary restriction.*

Using this definition we can give the following proof of containment:

Proof of Theorem 4. Let C be a log-space uniform BQ$_\mathsf{U}$CL quantum channel. Since C is unitary up until the last measurement step, it preserves all possible density matrices from the catalytic tape, and in particular it preserves the maximally mixed state I_n. Let U be the unitary part of C. The action of U on the work-tape and the catalytic tape, with the catalytic tape initialized in I_n, is:

$$U\left|0\right\rangle\left\langle 0\right|_w \otimes \frac{I_n}{2^n}U^\dagger = (\sqrt{p_0}\left|0\right\rangle\left\langle 0\right|_{w_0}\left|\psi_0\right\rangle\left\langle\psi_0\right|_w + \sqrt{p_1}\left|1\right\rangle\left\langle 1\right|_{w_0}\left|\psi_1\right\rangle\left\langle\psi_1\right|_w) \otimes \frac{I_n}{2^n}$$

with $|p_1| \geq 2/3$ in a "yes" instance and $|p_0| \geq 2/3$ in a "no" instance. Note that this calculation is of the exact form of a $\log(n)$-clean qubit machine and that the output probabilities are a constant bounded away from $1/2$; hence this problem is in DQC$_\mathsf{k}$, and by Lemma 52 is therefore in DQC$_1$ ◀

B.3 Containment of CL in DQC$_1$

We aim to show that CL $\subseteq$ DQC$_1$. The idea is that CL, as per Theorem 44, can always be made reversible. While as discussed before we cannot maintain reversibility and total obliviousness, a CL machine can also always be made "almost oblivious" while maintaining reversibility; the tape head movements are independent of the input, but the machine does not know when to halt. Instead, after any given amount of time, we know that the machine has halted on a fraction $1/\mathsf{poly}(n)$ of possible initial catalytic states. Since the DQC$_1$ model can be interpreted as sampling from a uniform distribution of computational basis states, this shows the probability of finding the correct output is $1/2 + 1/\mathsf{poly}(n)$, which is sufficient for the proof.

▶ **Definition 54.** *A* non-halting reversible oblivious *catalytic Turing machine is a reversible oblivious catalytic Turing machine that need not halt absolutely. In particular, for every input x and initial catalytic state c there exists a time $t(x, c)$ where the correct output has been written to the output tape and the catalytic tape has been reset to its initial state. In addition, the output state has an additional binary cell that indicates whether or not the output has been determined yet, or is still "unknown" by the machine.*

▶ **Definition 55.** *We say a reversible oblivious catalytic Turing machine* halts with polynomial success probability *if there exists polynomials p, q such that for any valid input x to a promise problem, after time $p(|x|)$ the output tape of the catalytic Turing machine contains the correct output to the problem on a fraction of at least $1/q(|x|)$ when the initial catalytic tapes are taken uniformily at random. After time $p(|x|)$, the output tape of the catalytic Turing machine never contains the wrong answer, but it may leave the output undetermined.*

We show that any CL machine can be transformed into a reversible oblivious catalytic Turing machine that halts with polynomial success probability. We defer the proof of this fact to the full version of the paper.

▶ **Lemma 56.** *Any catalytic Turing machine M that has a logarithmic clean space and polynomial size catalytic tape can be turned into a non-halting oblivious reversible catalytic Turing machine M^o with a logarithmic clean tape and polynomial catalytic tape.*

We call the machine formed this way M^o for oblivious M. Since the catalytic and clean tape are no more than polynomial length, this procedure adds at most a polynomial factor to the runtime. However, since the runtime of M may be super-polynomial and an oblivious machine has the same runtime for all inputs x of the same length and catalytic tapes c, the machine does not have enough clean space to keep a clock to know whether or not it has terminated. This means we cannot assume it to be halting. However, we can show that it is halting with sufficient probability (we again defer this proof to the full version of the paper):

▶ **Lemma 57.** *For any language L in CL that is recognized by a catalytic Turing machine M, there exists a reversible oblivious catalytic Turing machine N that halts with polynomial probability that also recognizes L. Furthermore, N also uses $O(\log |x|)$ clean space and polynomial catalytic space.*

This completes all technical components necessary to show that $\mathsf{CL} \subseteq \mathsf{DQC}_1$.

Proof of Theorem 5. The maximally mixed state of DQC_1 can be interpreted as uniformly randomly sampling computational basis states. If we take these basis states to be the catalytic tape and use the fact that DQC_1 is unchanged if we allow a logarithmic number of clean qubits, then we can run the machine N from Lemma 57 by using unitary gates instead of reversible, oblivious operations. When we measure the output bit at the end, we get either an indeterminate state or the correct output with certainty. If we get an indeterminate state, we output a random bit and thus output the correct answer with probability 1/2. If not, then we output the correct answer, which occurs with probability at least $1/\mathsf{poly}(n)$. ◀