Oracle Separations for Non-adaptive Collapse-free Quantum Computing

Henrique Hepp[b,*], Murilo V. G. da Silva[b], L. M. Zatesko[a,*]

[a]*Academic Department of Informatics, Federal University of Technology — Paraná, Brazil*
[b]*Department of Informatics, Federal University of Paraná, Brazil*

Abstract

Aaronson et al. (2016) introduced the quantum complexity class naCQP (*non-adaptive Collapse-free Quantum Polynomial time*), also known as PDQP (*Product Dynamical Quantum Polynomial time*), aiming to define a class larger than BQP (*Bounded-error Quantum Polynomial time*), but not large enough to include NP-complete problems. Aaronson et al. showed that SZK (*Statistical Zero Knowledge*) is contained in naCQP and that there is an oracle A for which $\mathsf{NP}^A \not\subseteq \mathsf{naCQP}^A$. We prove that: there is an oracle A for which $\mathsf{P}^A = \mathsf{BQP}^A = \mathsf{SZK}^A = \mathsf{naCQP}^A \neq (\mathsf{UP} \cap \mathsf{coUP})^A = \mathsf{EXP}^A$, where UP (*Unambiguous Polynomial time*) is an important subclass of NP; relative to an oracle A chosen uniformly at random, it holds $(\mathsf{UP} \cap \mathsf{coUP})^A \not\subseteq \mathsf{naCQP}^A$ with probability 1. Our results are a strengthening of the results by Bennett et al. (1997), Fortnow and Rogers (1999), Tamon and Yamakami (2001), and Aaronson et al. (2016).

Keywords: Computational Complexity, Oracle Separation, Unambiguous Polynomial-time, Collapse-free Quantum Computing.

1. Introduction

Aaronson et al. [1] introduced the complexity classes CQP (*Collapse-free Quantum Polynomial time*) and naCQP (*non-adaptive Collapse-free Quantum*

[*]Corresponding author

Email addresses: `hhepp@inf.ufpr.br` (Henrique Hepp), `murilo@inf.ufpr.br` (Murilo V. G. da Silva), `zatesko@utfpr.edu.br` (L. M. Zatesko)

Polynomial time). The former is the class of the problems that can be solved by a polynomial-time quantum algorithm allowing measurements not to cause the collapse of the states. The latter is the class CQP with the restriction that the quantum operations do not depend on the results of non-collapsing measurements. We refer the reader to Section 2 for technical details in the definition of naCQP. Note that the aim of defining such complexity classes in this line of research is not to propose alternative models of physically realizable computation, but to investigate classes of problems that seem to be larger than BQP, but not large enough to include NP-complete problems. The class naCQP is also called as PDQP (*Product Dynamical Quantum Polynomial time*) in the preprint version [2] and in three subsequent articles [3, 4, 5]. The class naCQP, or PDQP, is a subclass of the class DQP (*Dynamical Quantum Polynomial-Time*) introduced by Aaronson [6]. The class DQP was defined assuming a *hidden variable theory*, and that is possible to access in real time the evolution of the hidden variables. The complexity relation between DQP and CQP is unknown.

Aaronson et al. [1] showed that naCQP includes not only BQP, but also SZK, which is the class of the problems that admit zero-knowledge interactive proof systems, such as the Graph Isomorphism Problem. Assuming derandomisation hypotheses, we would have $\mathsf{P} = \mathsf{BPP}$ and $\mathsf{NP} = \mathsf{MA} = \mathsf{AM}$ [7], which would imply $\mathsf{SZK} \subseteq \mathsf{NP} \cap \mathsf{coNP}$. Moreover, Aaronson et al. showed that there is an oracle A for which $\mathsf{NP}^A \not\subseteq \mathsf{naCQP}^A$, which indicates that naCQP, although a superclass of BQP and SZK, seems not to be large enough to contain NP-complete problems. Remark that, since there is an oracle A for which $\mathsf{BQP}^A \not\subseteq \mathsf{NP}^A$ (in fact, $\mathsf{BQP}^A \not\subseteq \mathsf{PH}^A$ [8]), we also have $\mathsf{naCQP}^A \not\subseteq \mathsf{NP}^A$ for this oracle A.

The class UP (*Unambiguous Polynomial-Time*), a subclass of NP, is the class of problems Π for which there is a non-deterministic Turing machine M such that: if x is a positive instance of Π, then there is a single accepting computation path of M on x; if x is a negative instance of Π, then all computations of M on x end in rejection. Fortnow and Rogers [9] showed that $\mathsf{P}^A = \mathsf{BQP}^A \neq \mathsf{UP}^A \cap \mathsf{coUP}^A$ for the same oracle A. The relation between $\mathsf{UP} \cap \mathsf{coUP}$ and other complexity classes has also been explored. For instance, Menda and Watrous [10] showed that there is an oracle A for which $\mathsf{UP}^A \cap \mathsf{coUP}^A \not\subseteq \mathsf{QSZK}^A$, which implies $\mathsf{UP}^A \cap \mathsf{coUP}^A \not\subseteq \mathsf{SZK}^A$. Beigel et al. [11] also showed that there is an oracle A for which $\mathsf{P}^A = \oplus\mathsf{P}^A$ and $\mathsf{RP}^A = \mathsf{EXP}^A$ (recall that $\mathsf{P}^A \neq \mathsf{EXP}^A$ for all oracle A). As noted by Tamon and Yamakami [12], since $\mathsf{UP}^A \subseteq \oplus\mathsf{P}^A$ and $\mathsf{RP}^A \subseteq \mathsf{BQP}^A$, we have $\mathsf{P}^A = \mathsf{UP}^A$

and $\mathsf{BQP}^A = \mathsf{EXP}^A$. Curiously, Tamon and Yamakami [12] also claimed that there is an oracle A for which $\mathsf{P}^A = \mathsf{BQP}^A$ and $\mathsf{UP}^A = \mathsf{EXP}^A$, presenting a proof sketch for this. We prove the following strengthening of their claim.

Theorem 1. *There is an oracle A for which $\mathsf{P}^A = \mathsf{naCQP}^A$ and $(\mathsf{UP} \cap \mathsf{coUP})^A = \mathsf{EXP}^A$.*

Tamon and Yamakami's sketch of proof is not easy to follow and contains some technical details that we are not sure that are correct, so we present the full proof of our more general result, roughly inspired by their ideas.

Figure 1 illustrates the known relationship between the main complexity classes relevant to this paper, highlighting the result stated in Theorem 1.

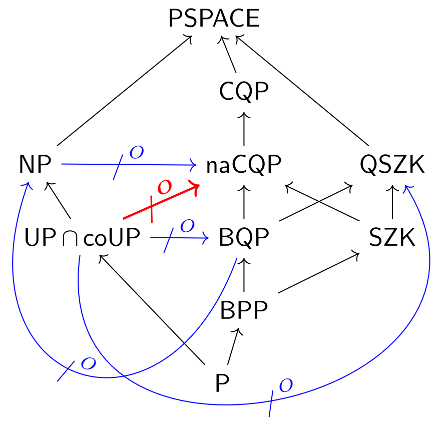

Figure 1: Relationship between complexity classes in this paper. The black arrows indicate inclusion between the classes. The striked-out arrows indicate that there is an oracle for which the class is not included in the other, being the blue ones results from the literature, and the red thicker one representing the result of Theorem 1 and (for a random oracle) Theorem 2.

A *random oracle* is an oracle chosen uniformly at random. Bennett et al. [13] showed that, relative to a random oracle A, we have $\mathsf{NP}^A \not\subseteq \mathsf{BQP}^A$ with probability 1. In fact, the authors further show that $(\mathsf{NP} \cap \mathsf{coNP})^A \not\subseteq \mathsf{BQP}^A$ with probability 1. We also show the following.

Theorem 2. *Relative to a random oracle A, we have $(\mathsf{UP} \cap \mathsf{coUP})^A \not\subseteq \mathsf{naCQP}^A$ with probability* 1*.*

The proof for Theorem 2 follows the structure of the proof by Bennett et al. for BQP, but using properties for naCQP that we have also used in the proof of Theorem 1. Some of these properties are stated in Theorem 3 below, which is a strengthening of the well-known *BBBV Theorem* for BQP by Bennett et al. [13].

Theorem 3. *Let M^A be an naCQP algorithm with an oracle A and let $p(n)$ be the running time of M^A. Let x be a n-bit string given as input to M. For every $\varepsilon > 0$ there is a set of strings S with $|S| \leq 200(p(n))^4/\varepsilon^2$ such that, for any oracle A', if A' differs from A only on a single string y and $y \notin S$, then*

$$\left|\Pr\left[M^{A'} \textit{ accepts } x\right] - \Pr\left[M^{A} \textit{ accepts } x\right]\right| \leq \varepsilon \,.$$

This paper is organised as follows. In the remaining of this section, we discuss further motivation concerning our results. In Section 2, we present the technical definition of naCQP. In Section 3, we present the proof of Theorem 3, needed for the proofs of Theorems 1 and 2 presented in Section 4.

Further motivation

According to Aaronson [14], most of the techniques that existed in the 1960s and 1970s for demonstrating relationships between complexity classes, such as $\mathsf{C} \subseteq \mathsf{D}$ or $\mathsf{C} \not\subseteq \mathsf{D}$, were so general that, when proved, they also proved $\mathsf{C}^A \subseteq \mathsf{D}^A$ or $\mathsf{C}^A \not\subseteq \mathsf{D}^A$ for all possible oracles A. An example is the demonstration of $\mathsf{P}^A \neq \mathsf{EXP}^A$ for every oracle A. However, there is an oracle A for which $\mathsf{P}^A = \mathsf{NP}^A$ and another oracle B for which $\mathsf{P}^B \neq \mathsf{NP}^B$ [15]. This means that a proof for $\mathsf{P} \neq \mathsf{NP}$ (or $\mathsf{P} = \mathsf{NP}$) must use more sophisticated techniques that do not relativise.

Oracle separations between NP and any class of efficient computation in some model (such as P, BPP, and BQP) are an evidence for the hardness of deciding solvability of NP-complete problems in that model [13]. In particular, oracle separations between NP and naCQP, which we address, raise the hardness of deciding whether quantum computers are able to solve NP-complete problems to another level. Having oracle separations between NP and a non-collapse quantum computing class such as naCQP suggests that state collapsing may not be the only limitation of quantum computers for solving NP-complete problems.

We now discuss how our oracle separations results help to better understand the related classes. Concerning our Theorem 1, remark that:

- there is an oracle A for which $\mathsf{P}^A = \mathsf{BQP}^A = \mathsf{SZK}^A = \mathsf{naCQP}^A \neq \mathsf{UP}^A \cap \mathsf{coUP}^A = \mathsf{EXP}^A$.

- our result is a strengthening of the result by Aaronson et al., concerning naCQP and NP, and also of the result by Fortnow and Rogers, concerning P, BQP, and $\mathsf{UP} \cap \mathsf{coUP}$;

- concerning SZK and $\mathsf{UP} \cap \mathsf{coUP}$, our result implies not only that there is an oracle A for which $\mathsf{UP}^A \cap \mathsf{coUP}^A \not\subseteq \mathsf{SZK}^A$, something which is already implied by Menda and Watrous, but also that $\mathsf{UP}^A \cap \mathsf{coUP}^A \not\subseteq \mathsf{P}^A = \mathsf{SZK}^A$.

Concerning Theorem 2, remark that:

- our result is a strengthening of the result by Bennett et al. concerning $\mathsf{UP} \cap \mathsf{coUP}$ and BQP;

- our result is also a strengthening of the result by Aaronson et al. that there is an oracle A for which $\mathsf{NP}^A \not\subseteq \mathsf{naCQP}^A$.

Concerning Theorem 3, the original theorem for BQP (BBBV Theorem) is an important tool used by Bennet et al. [13], Fortnow et al. [9] and Tamon and Yamakami [12] to show the oracular separations between BQP and NP and between BQP and UP. All these results can be extended to naCQP.

Finally, we would like to emphasise the importance of the results obtained in this work. Concerning naCQP, we studied the limits of quantum computing along the lines of the papers by [13] and [1]. When it is shown that BQP does not contain NP relative to an oracle, it is suggested that quantum computers are not capable of solving NP-complete problems. We showed that even if a quantum computer could make measurements without collapse and even with a subclass of NP, the limitations of the quantum model remain valid.

2. Definition of naCQP and technical assumptions

Consider a quantum circuit C with n qubits, constructed depending on an n-bit input string x, defined by the following sequence of operators $C = (U_1, \mathsf{M}_1, U_2, \mathsf{M}_2, \ldots, U_R, \mathsf{M}_R)$, where each U_t is a unitary operator of n qubits, and each M_t is a quantum measurement, in the computational basis, of m_t qubits, being $0 \leq m_t \leq n$. The initial state of the circuit is $|\psi_0\rangle = |0\rangle^{\otimes n}$, and after the t-th measurement the state is $|\psi_t\rangle = \mathsf{M}_t \, U_t \, |\psi_{t-1}\rangle$, so that the states at the different stages of the circuit are given by the sequence of random variables $\{|\psi_t\rangle\}_{t=0}^{R}$, subject to a probability distribution that depends on the circuit C. Consider a procedure that takes as input the circuit C and samples the sequence $\{|\psi_t\rangle\}_{t=1}^R$ from this probability distribution. Then, the procedure measures, on the computational basis, the states $|\psi_t\rangle$ for each t independently. The R results of these measurements are returned, named $v_1, \ldots, v_R$. Assuming that this procedure is done in only one step, we call this procedure the *oracle* Q. Note that if non-collapsing measurements were allowed, the result of Q would be equivalent to the case wherein, for each t, the m_t qubits measured in M_t collapse, being the remaining $n - m_t$ qubits measured without collapsing. Recall that it is possible that the measurements $\mathsf{M}_1, \mathsf{M}_2, \ldots, \mathsf{M}_R$ are of less than n (even possibly zero) qubits (if $m_t = 0$, then the measurement M_t is full non-collapsing). Observe that Q samples all states $\{|\psi_t\rangle\}$ at once, yielding a *non-adaptive* model of non-collapsing measurements.

We define naCQP (*non-adaptive Collapse-free Quantum Polynomial*) as the class of promise problems that can be solved by an naCQP *algorithm*, i.e. a polynomial-time Turing machine with error probability less than or equal to 1/3 that, receiving an n-bit input string x, writes the description of the quantum circuit C and makes a single query to the oracle Q as defined above. An $\mathsf{naCQTime}(T(n))$ algorithm is defined similarly to an naCQP algorithm, but being $O(T(n))$ the running time of the Turing machine, and being $\mathsf{naCQTime}(T(n))$ the corresponding complexity class.

Clearly, $\mathsf{BQP} \subseteq \mathsf{naCQP}$, since we can provide the oracle with a polynomial quantum circuit and use only the measurement result at the end of the circuit. On the other hand, as pointed by Aaronson et al.[1] "the oracle Q gives us information about the intermediate stages of the quantum computation without collapsing the state; this is what gives naCQP additional power over BQP", even making quantum computers in this model capable of solving SZK-complete problems (recall that it is not known if $\mathsf{SZK} \subseteq \mathsf{BQP}$).

In order to simplify our proofs, we show an equivalent circuit for naCQP using a procedure in which the measurements are delayed. This procedure is also described by Aaronson et al. [1, Appendix C]. As explained before, given a circuit C, the states at the different stages of the circuit are given by the sequence $\{|\psi_t\rangle\}_{t=0}^R$. Then, the oracle Q returns measurements of the sequence $\{|\psi_t\rangle\}_{t=1}^R$ named $v_1, \ldots, v_R$. Note that $v_1, \ldots, v_R$ is a sequence of random variables governed by a Markov distribution. That is, for all $1 \leq t \leq R$, we have that v_t is independent of $v_0, \cdots, v_{t-2}$ conditioned on a particular value of v_{t-1}. More precisely, in our case, the variable v_t depends on the qubits of $|\psi_{t-1}\rangle$ that have been measured by $\mathsf{M}_1, \ldots, \mathsf{M}_{t-1}$ and thus have collapsed. Now, suppose that the oracle Q does not sample all the states

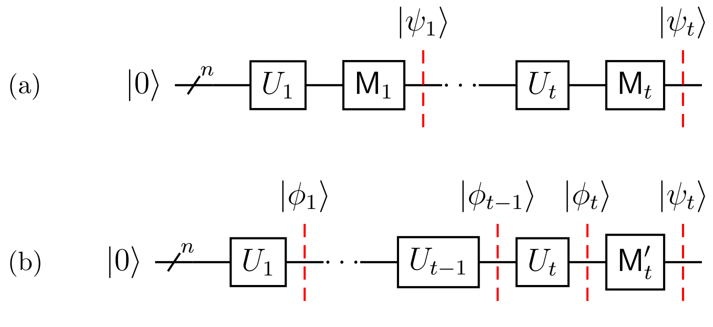

Figure 2: (a) The original naCQP circuit used by Q for sampling all $|\psi_1\rangle, \ldots, |\psi_t\rangle$. (b) The circuit constructed by Q for sampling only $|\psi_t\rangle$ with the measurements being delayed.

$\{|\psi_t\rangle\}$ at once, but, instead, the following procedure is performed. The oracle Q samples each $|\psi_t\rangle$ one at a time. At each sampling, the oracle builds an equivalent circuit in which all measurements $\mathsf{M}_1, \ldots, \mathsf{M}_{t-1}$ are delayed, according to the well-known Principle of Deferred Measurement (PDM)[16], and an equivalent measurement M'_t takes place immediately before the state $|\psi_t\rangle$. However, since the construction of this equivalent circuit is conditioned on the value of v_{t-1} in the previous sampling, M'_t depends not only on the deferred measurements $\mathsf{M}_1, \ldots, \mathsf{M}_{t-1}$, but also on the former collapsed qubits, with which the current sampling must be consistent. Hence, M'_t is designed so that the measurements that would have taken place up to time $t-1$ now collapse to the same values as they did in the last sampling. Throughout this text, therefore, we assume without loss of generality that each state is sampled using the procedure above, with all the collapsing measurements delayed, conditioned on the value of the previous sampling. Also, for a fixed t, we use $|\phi_t\rangle$ to denote the state immediately before the single measurement M'_t performed by Q to obtain $|\psi_t\rangle$, and $|\phi_0\rangle, \ldots, |\phi_{t-1}\rangle$ are used to denote the previous states, with no measurements performed. Moreover, although applying PDM modifies gates $U_1, \ldots, U_t$, by abuse we maintain the names $U_1, \ldots, U_t$, assuming that the gates have been modified so that the measurements are deferred (see Figure 2).

Being A a language, an naCQP *algorithm with oracle*[2] A is an naCQP algorithm that, receiving an input x of length n, can query a function f :

[2]Please do not mistake this oracle A for the also called *oracle* Q in the definition of an naCQP algorithm. Although using the term *oracle* in both situations may be confusing, we still do so to follow what is standard in the literature.

$\{0,1\}^n \to \{0,1\}$ such that $f(x) = 1$ if and only if $x \in A$. That is, for the construction of the unitary operators $U_1, \cdots, U_R$, we are given access to the n-qubit unitary transformation defined as $U_f : |x, b\rangle \mapsto |x, b \oplus f(x)\rangle$, for $x \in \{0,1\}^n$ and $b \in \{0,1\}$, where the operation $\oplus$ indicates addition modulo 2, or, equivalently,

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle \ , \quad \text{for } x \in \{0,1\}^n.$$

Throughout this text, we assume without loss of generality that in an naCQP algorithm with an oracle query to f, each of the unitary transformations $U_1, \ldots, U_R$ is either a copy of U_f, or an n-qubit circuit constructed using only gates from a fixed finite universal gate set.

Let M^A be an naCQP algorithm with oracle A, obeying our assumption on the transformations $U_1, \ldots, U_R$ and U_f as above. Being y a fixed binary string, let $q_y(|\phi_t\rangle)$ be the modulus squared of the amplitude of $|y\rangle$ in $|\phi_t\rangle$ if U_t is a copy of U_f, and 0 otherwise. Note that, since $|\phi_t\rangle$ can be viewed as a superposition of all the configurations of the algorithm at time t, we have that $q_y(|\phi_t\rangle)$ is the sum of modulus squared of amplitudes in $|\phi_t\rangle$ corresponding only to the configurations at time t which are querying the oracle A on string y.

3. Proof of Theorem 3

Before we present the proof of Theorem 3, we first need a few technical lemmas.

Lemma 4 (adapted from Bennett et al., 1997 [13])**.** *Let M^A be an naCQP algorithm with an oracle A as in Section 2. For some fixed t, let T be the number of copies of U_f amongst gates $U_1, \ldots, U_t$, and, for $\varepsilon > 0$, let $F \subseteq [1, t] \times \Sigma^*$*

1. *for each $(i, y) \in F$, U_i is a copy of U_f;*
2. *$\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \leq \varepsilon^2/2T$ (recall the definition of $|\phi_i\rangle$ in Section 2).*

Now suppose that the answer to each query $(i, y) \in F$ is modified to some arbitrary fixed bit $a_{i,y}$, being these answers not necessarily consistent with an oracle. Let $|\phi'_i\rangle$ be defined as $|\phi_i\rangle$, but with respect to oracle A modified as above. Then, $\||\phi_t\rangle - |\phi'_t\rangle\| \leq \varepsilon$. □

The proof of Lemma 4 follows by inspection on the same proof showed by Bennett et al. for BQP, but adapting the arguments for naCQP (reproduced in Appendix A for the sake of completeness). Recall that we have assumed that no collapsing measurements have occurred before time t (according to the definition of $|\phi_i\rangle$ in Section 2), which means that, until that point, our naCQP algorithm is behaving like a BQP algorithm, similarly as Aaronson et al. [1] also pointed out in their paper. Furthermore, the reader may notice that the authors originally stated $\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \leq \varepsilon^2/T$, but we remark that the same proof also shows that $\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \leq \varepsilon^2/2T$.

Lemma 5 (Aaronson et al., 2016 [1]). *Let $R \geq 1$, and let $v = (v_0, \cdots, v_R)$ and $w = (w_0, \cdots, w_R)$ two random variables governed by two Markov distributions. Then, the total variation distance between these random variables is $d_{TV}(v, w) \leq 2\sum_{i=1}^{R} d_{TV}((v_{i-1}, v_i), (w_{i-1}, w_i))$.* $\square$

Aaronson et al. [1] showed, in Theorem 6.1 in their paper, an $\Omega(2^{n/4})$ lower bound for a non-structured search with an naCQP algorithm, analogous to the well-known $\Omega(2^{n/2})$ lower bound for a non-structured search with a BQP algorithm [13]. The main argument in their proof can be summarised in what is stated below in Lemma 6.

Lemma 6. *Let M^A be an naCQP algorithm with an oracle A as in Section 2. Being A' an oracle which answers arbitrarily to any query, let v and w be the results of the non-collapsing measurements for M^A and $M^{A'}$ respectively. Let $d_i = d_{TV}((v_{i-1}, v_i), (w_{i-1}, w_i))$, then $d_i \leq 5\||\phi_{i-1}\rangle - |\phi'_{i-1}\rangle\|$.* $\square$

Now we present Theorem 7, from which follows Corollary 8 and Theorem 3.

Theorem 7. *Let M^A be an naCQP algorithm with an oracle A as in Section 2. Let T be the number of copies of U_f amongst gates $U_1, \ldots, U_R$, and, for $\varepsilon > 0$, let $F \subseteq [1, R] \times \Sigma^*$ be a set of time-string pairs such that:*

1. *for each each $(i, y) \in F$, U_i is a copy of U_f;*
2. *$\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \leq \varepsilon^2/2T$.*

Now suppose that the answer to each query $(i, y) \in F$ is modified to some arbitrary fixed $a_{i,y}$, being these answers not necessarily consistent with an oracle. Let $|\phi'_i\rangle$ be defined as $|\phi_i\rangle$, but with respect to oracle A modified *as above. Then, being $v = (v_1, \dots, v_R)$ and $w = (w_1, \dots, w_R)$ the random variables returned by the sampling of $\{|\psi_t\rangle\}$ and $\{|\psi'_t\rangle\}$, respectively, the total variation distance between v and w is $d_{TV}(v,w) \leq 2\sum_{i=1}^{R} 5\varepsilon \leq 10R\varepsilon$.*

Proof. First, let us fix some i. Since we can delay all collapsing measurements to occur immediately before the sampling of v_i and w_i, we have, by Lemma 6, $d_i = d_{TV}((v_{i-1}, v_i), (w_{i-1}, w_i)) \leq 5\||\phi_{i-1}\rangle - |\phi'_{i-1}\rangle\|$. Also, by Lemma 4, since $\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \leq \varepsilon^2/2T$, we have $\| \, |\phi_{i-1}\rangle - |\phi'_{i-1}\rangle \, \| \leq \varepsilon$. Therefore, by Lemma 5,

$$d_{TV}(v,w) \leq 2\sum_{i=1}^{R} d_{TV}\left((v_{i-1}, v_i), (w_{i-1}, w_i)\right) \leq 2\sum_{i=1}^{R} 5\varepsilon = 10R\varepsilon\,.$$

□

Corollary 8. *Let M^A be an* naCQP *algorithm with an oracle A as in Section 2. Let T be the number of copies of U_f amongst gates $U_1, \dots, U_R$. For every $\varepsilon > 0$, there is a set of strings S with $|S| \leq 200T^2R^2/\varepsilon^2$ such that, for any oracle A', if A' differs from A only on a single string y and $y \notin S$, then, being $v = (v_1, \dots, v_R)$ and $w = (w_1, \dots, w_R)$ the random variables returned by M^A and $M^{A'}$, we have $d_{TV}(v,w) \leq \varepsilon$.*

Proof. First, we follow the proof by Bennett et al. [13] for the analogous result concerning BQP. Since each $|\phi_i\rangle$ has unit length, $\sum_{i=1}^{R}\sum_y q_y(|\phi_i\rangle) \leq T$. Let S be the set of strings y such that $\sum_{i=1}^{R} q_y(|\phi_i\rangle) \geq \frac{\varepsilon^2}{2T}$. Since for any y in S we have $|S|\sum_{i=1}^{R} q_y(|\phi_i\rangle) \leq T$, we can conclude that $|S| \leq \frac{2T^2}{\varepsilon^2}$. If $y \notin S$ then

$$\sum_{i=1}^{R} q_y(|\phi_i\rangle) < \frac{\varepsilon^2}{2T}\,. \tag{1}$$

Now, from (1), we apply Theorem 7, obtaining that for all $y \notin S$, $d_{TV}(v,w) \leq 10R\varepsilon$. Considering $\varepsilon = \frac{\varepsilon'}{10R}$, we have

$$|S| \leq \frac{2T^2}{(\varepsilon'/10R)^2} = \frac{200T^2R^2}{\varepsilon'^2}\,.$$

Hence, for all $y \notin S$, we have $d_{TV}(v,w) \leq \varepsilon'$ as desired. □

Proof of Theorem 3. Follows directly from Corollary 8. □

4. Oracle separations for naCQP

Proof of Theorem 1. We prove that there is an oracle A for which $\mathsf{P}^A = \mathsf{naCQP}^A$ and $\mathsf{UP}^A \cap \mathsf{coUP}^A = \mathsf{EXP}^A$. Let $M^?$ be a naCQP algorithm which can query an oracle such that given the input $(0^j, x)$, $j \leq |x|$: runs in linear time on $j + |x|$ for any oracle; makes a query of size smaller than $|x|$; has error probability smaller than $2^{-j-|x|}$. Let $\{M_k^?\}_{k\in\mathbb{N}}$ be an enumeration of such algorithms.

Claim 1.1. *Let $\{M_k^A\}_{k\in\mathbb{N}}$ be the enumeration of the naCQP^A algorithms as defined and restricted above. Then, naCQP^A can be given as the set of the languages* decided *by an algorithm in the set $\{M_k^A\}_{k\in\mathbb{N}}$.*

Proof of the claim. The proof uses a padding argument. Let $W^?$ be an naCQP algorithm that queries an oracle and runs in time at most $|x|^c$ for some $c \in \mathbb{N}$. Then, $M_k^?$ is the naCQP algorithm that queries an oracle and that, on input $(0^{|x|}, x0^{|x|^c-|x|})$, emulates $W^?(x)$. Assuming that $M_k^?$ rejects if the input does not fit in this format, we have clearly that $M_k^?$, on input $(0^j, y)$, runs in linear time on the length of its input (i.e. $j + |y|$) and makes oracle queries only of length smaller than $|y|$. Also, we assume w.l.o.g. that if $x \in L$, then $\Pr[M_k(0^{|x|}, x0^{|x|^c-|x|}) = 1] \geq 1 - 2^{-|x|-|x|^c}$, and if $x \notin L$, then $\Pr[M_k(0^{|x|}, x0^{|x|^c-|x}) = 1] \leq 2^{-|x|-|x|^c}$. $\Diamond$

For the oracle A we want to construct, let L^A be a EXP^A-complete language and let E^A be a deterministic Turing machine which queries A that decides L^A in exponential time. We can also assume w.l.o.g that E^A runs in time $2^{|x|}$ given input $(0^{j\leq|x|}, x)$. Let $\langle\cdot,\cdot,\cdot,\cdot,\cdot\rangle$ be a function from Σ^5 to Σ such that $|\langle a,b,c,d,e\rangle| > \max\{|a|,|b|,|c|,|d|,|e|\}$ for all $a,b,c,d,e \in \Sigma$. Given $k > 0$ and $s \in \{0,1\}^*$, we denote s_k^n as the $(k+1)$-th string in $\{0,1\}^n$. If $k = 0$ we have $s_0^n = 0^n$. We construct A such that for any sufficiently large n and all strings x of size n and all k with $0 \leq k < n$,

(a) $\Pr[M_k^A(0^{j\leq n}, x) = 1] \geq 1 - 2^{-2n} \Rightarrow \langle 00, s_k^{\lceil\log(n)\rceil}, x, 1^{n^2}, 1^{n^2}\rangle \in A$;

(b) $\Pr[M_k^A(0^{j\leq n}, x) = 0] \geq 1 - 2^{-2n} \Rightarrow \langle 00, s_k^{\lceil\log(n)\rceil}, x, 1^{n^2}, 1^{n^2}\rangle \notin A$;

(c) $x \in L^A \Rightarrow |\{y \in \Sigma^{n^2} : \langle 01, 1^{\lceil\log(n)\rceil}, x, y, 1^{n^2}\rangle \in A\}| = 1$ and $|\{z \in \Sigma^{n^2} : \langle 10, 1^{\lceil\log(n)\rceil}, x, 1^{n^2}, z\rangle \in A\}| = 0$;

(d) $x \notin L^A \Rightarrow |\{y \in \Sigma^{n^2} : \langle 01, 1^{\lceil\log(n)\rceil}, x, y, 1^{n^2}\rangle \in A\}| = 0$ and $|\{z \in \Sigma^{n^2} : \langle 10, 1^{\lceil\log(n)\rceil}, x, 1^{n^2}, z\rangle \in A\}| = 1$.

Claim 1.2. *If conditions (a) and (b) are satisfied, then* $\mathsf{P}^A = \mathsf{naCQP}^A$.

Proof of the claim. A deterministic polynomial-time Turing machine can decide the same language as M_k^A by simply querying the oracle A about the string $\langle 00, s_k^{\lceil \log(n) \rceil}, x, 1^{n^2}, 1^{n^2} \rangle$. $\Diamond$

Claim 1.3. *If conditions (c) and (d) are satisfied, then* $\mathsf{UP}^A \cap \mathsf{coUP}^A = \mathsf{EXP}^A$.

Proof of the claim. A non-deterministic polynomial-time Turing machine can decide L^A by guessing an unambiguous $y \in \Sigma^{n^2}$ and querying the oracle about $\langle 01, 1^{\lceil \log(n) \rceil}, x, y, 1^{n^2} \rangle$, this machine can also decide coL^A by guessing an unambiguous $z \in \Sigma^{n^2}$ and querying the oracle about $\langle 10, 1^{\lceil \log(n) \rceil}, x, 1^{n^2}, z \rangle$. $\Diamond$

We say that:

- w is a 00-string, if w has the form $\langle 00, s_k^{\lceil \log(n) \rceil}, x, 1^{n^2}, 1^{n^2} \rangle$ with $0 \leq k < n$;
- w is a 01-string, if w has the form $\langle 01, 1^{\lceil \log(n) \rceil}, x, y, 1^{n^2} \rangle$ with $|y| = n^2$;
- w is a 10-string, if w has the form $\langle 10, 1^{\lceil \log(n) \rceil}, x, 1^{n^2}, z \rangle$ with $|z| = n^2$.

We construct A in stages, for each string x following the lexicographic order. At each stage, the oracle is described as a partial function σ_A. The initial stage is defined as $\sigma_A(\lambda) = 0$ for the empty string λ. For each stage x, we consider all possible strings w from $w = \langle 00, 0, x, 0, 0 \rangle$ to $w = \langle 10, 1^{\lceil \log(n) \rceil}, x, 1^{n^2}, 1^{n^2} \rangle$.

Let $p(w)$ be a qubit as recursively defined below. Note that we want $p(w)$ to represent the oracle's answer, that is, if the answer is 0 (1), than $p(w)$ collapses to $|0\rangle$ ($|1\rangle$) with high probability, i.e. $\geq 1 - 2^{-2n}$.

Step 1. For each string w that is not 00-string, 01-string, or 10-string: $p(w) = |0\rangle$.

Step 2. We assign for every 01-string w and 10-string w a variable v_w that indicates whether w belongs to the oracle A, and do $p(w) = |v_w\rangle$. The variable v_w will be valued later. In the case wherein $w \in \mathrm{dom}(\sigma_A)$, we will have $p(w) = |\sigma_A(w)\rangle$.

Step 3. If w is a 00-string, we define $p(w)$ as the quantum state corresponding to the output, just before the final measurement, of $M_k^A(0^{j \leq n}, x)$. Note that, since the size of each query is smaller than $|x|$, and therefore smaller than $\sqrt{|w|}$, the answers to these queries can be assumed to be defined by recursion. Then, we add w to A if and only if the majority output of $p(w)$ is 1.

To complete the construction of oracle A at stage x, we now need to value v_w when w is a 01-string or a 10-string. We denote a string $\mathbf{r}$ of length $2 \cdot 2^{n^2}$ containing in its first half the indexing of all strings $y \in \Sigma^{n^2}$, and in its second half the indexing of all strings $z \in \Sigma^{n^2}$. That is, each bit of the first half of $\mathbf{r}$ corresponds to a 01-string of the form $\langle 01, 1^{\lceil \log(n) \rceil}, x, y, 1^{n^2} \rangle$ and each bit of the second half of $\mathbf{r}$ corresponds to a 10-string of the form $\langle 10, 1^{\lceil \log(n) \rceil}, x, 1^{n^2}, z \rangle$. We define $C_x(\mathbf{r})$ to be the output of the emulation of $E^A(0^{j \leq n}, x)$ with the condition that whenever E makes a query w, we compute $p(w)$ and use its majority output.

Claim 1.4. *For each x, there is a string $\mathbf{r}$ given by $\mathbf{r} = \mathbf{d}_i = 0^i 1 0^{2^{n^2+1}-i-1}$ $(0 \leq i < 2^{n^2+1})$ that can be in one of two cases:*

1. *$C_x(\mathbf{d}_i) = 1$, if $i < 2^{n^2}$;*
2. *$C_x(\mathbf{d}_i) = 0$, if $i \geq 2^{n^2}$.*

We interpret this string $\mathbf{r}$ such as follows. We assign $v_w = 1$ to the string w indexed by the $(i+1)$-th bit of $\mathbf{r}$, and $v_w = 0$ to all w indexed by the other bits of $\mathbf{r}$.

Proof of the claim. Due to the recursion used to obtain $p(w)$ for the 00-strings, the number of levels needed to compute $C_x(\mathbf{r})$ is at most $\lfloor \log n \rfloor - 1$. Therefore, the total number of queries that $C_x(\mathbf{r})$ makes is at most $\prod_{i=0}^{\lfloor \log n \rfloor} (2^n)^{1/2^i}$. Note that $\prod_{i=0}^{\lfloor \log n \rfloor} (2^n)^{1/2^i} < 2^{2n-1}$. Therefore, for a sufficiently large n, the probability that $C_x(\mathbf{r})$ obtains the correct answer is greater than $(1 - 2^{-2n})^{2^{2n-1}} > \sqrt{1/e} > 2/3$, assuming that the probability that $p(w)$ returns a wrong result is less than 2^{-2n}.

We need, finally, to consider that a subcomputation $p(w)$ may return a wrong result with probability greater than 2^{-2n}. For such cases we can consider that the subcomputation $p(w)$ returns $|1\rangle$ or $|0\rangle$ in an arbitrary way. According to Theorem 3 we have a polynomially small set S of oracle answers that can change with probability greater than ε the output of a quantum algorithm naCQP. We conclude, then, that there is always a way to set the values of v_w for these "bad" w so that neither of the following cases occur:

1. $i < 2^{n^2}$ and $C_x(\mathbf{d}_i) = 0$;
2. $i \geq 2^{n^2}$ and $C_x(\mathbf{d}_i) = 1$. $\Diamond$

Step 3 corresponding to the construction of the 00-strings, together with Claim 1.4 corresponding to the construction of the 01-strings and 10-strings, fulfil Conditions (a), (b), (c) and (d), thus concluding the proof. $\square$

Proof of Theorem 2. The statement of the original version of Lemma 4 for BQP by Bennett et al. [13] coincides with the one that we have presented for naCQP, since in our statement we are considering a time t before which no collapsing measurements have occurred, meaning that, up to that point, our naCQP algorithm is behaving like a BQP algorithm. Bennett et al. [13] used the original version of Lemma 4 for BQP to prove that, relative to a random oracle A, we have $(\mathsf{UP} \cap \mathsf{coUP})^A \not\subseteq \mathsf{BQP}^A$ with probability 1 (actually they prove this for $\mathsf{NP} \cap \mathsf{coNP}$, but the same proof works for $\mathsf{UP} \cap \mathsf{coUP}$). We adapt the proof of Bennett et al. by using Theorem 7 instead of Lemma 4 for BQP, and show that $(\mathsf{UP} \cap \mathsf{coUP})^A \not\subseteq \mathsf{naCQP}^A$ with probability 1.

First, remark that an oracle A can be conveniently thought of as a length-preserving function f on Σ^*, which can be accomplished by interpreting the oracle answer on the pair (x, i) as i-th bit of $f(x)$. We can also define a permutation oracle as a length-preserving function that for each $n \geq 0$ gives a permutation on Σ^n. Thus, for any such oracle A, let $L_A = \{y :$ first bit of $A^{-1}(y)$ is $1\}$, which is clearly in $(\mathsf{UP} \cap \mathsf{coUP})^A$ — the certificate for y, regardless whether $y \in L_A$ or $y \notin L_A$, is the only x such that $A(x) = y$, since A is a permutation oracle.

Let $M^?$ be an $\mathsf{naCQTime}(n)$ algorithm which can query an oracle and runs in time at most $T(n)$. Let $T(n) = o(2^{n/5})$. By sampling uniformly at random a permutation oracle A, we have with probability 1 that M^A gives the wrong answer on input 1^n, for some large enough picked n, implying that M^A does not accept L_A, which suffices to show that $(\mathsf{UP} \cap \mathsf{coUP})^A \not\subseteq \mathsf{naCQP}^A$. This probability 1, taken over the choice of a random *permutation* oracle (i.e. thought as a length-preserving function, the oracle acts as a permutation on Σ^*), remains the same when taken over the choice of a random oracle.

In what follows, we argue that M^A gives the wrong answer on input 1^n, but adapting the technical details in the context of naCQP.

We can sample random permutations on $\{0,1\}^n$ as follows. Let x_0, x_1, $\ldots$, $x_{T(n)+1}$ be strings chosen uniformly at random in $\{0,1\}^n$. Let π_0 be a permutation uniformly sampled amongst permutations π satisfying $\pi(x_0) = 1^n$. Now, being τ the transposition (x_{i-1}, x_i), define $\pi_i = \pi_{i-1} \cdot \tau$, i.e. $\pi_i(x_i) = \pi_{i-1}(x_{i-1})$ and $\pi_i(x_{i-1}) = \pi_{i-1}(x_i)$. Clearly each π_i is a random permutation on $\{0,1\}^n$. Let $\{A_i\}$ be a sequence of permutation oracles such that $A_i(x) = \pi_i(x)$ if $x \in \{0,1\}^n$, and $A_i(x) = A_j(x)$ if $x \notin \{0,1\}^n$, for every j (recall that Σ may have other symbols than 0 and 1, so all we need is to define the answers of all oracles A_i to be consistent whenever x is not a binary string).

Claim 2.1. *The probability of the string 1^n belongs to exactly one of the two languages $L_{A_{T(n)}}$ and $L_{A_{T(n)-1}}$ is $1/2$ (hence it is also $1/2$ the probability that 1^n belongs to none of the languages or to both).*

Proof of the claim. Both $A^{-1}_{T(n)}(1^n) = x_{T(n)}$ and $A^{-1}_{T(n)-1}(1^n) = x_{T(n)-1}$ have been sampled uniformly at random, so the probability that exactly one of them has the first bit equal to 1 is $1/2$. $\Diamond$

For naCQTime$(T(n))$, let $\{|\psi_t\rangle\}$ and $\{|\psi'_t\rangle\}$ be the states sampled by $M^{A_{T(n)}}$ and $M^{A_{T(n)-1}}$, respectively, on input 1^n, being $v = (v_1, \ldots, v_R)$ and $w = (w_1, \ldots, w_R)$ the corresponding random variables, as in Theorem 7.

Claim 2.2. $d_{VT}(v, w) \leq 1/50$

Proof of the claim. Consider that the naCQTime$(T(n))$ algorithm $M^?$ queries a different oracle A_t for each $t = 1, \ldots, R$, being $\{|\overline{\psi}_t\rangle\}$ the corresponding states and $q = (q_1, \ldots, q_R)$ the corresponding random variables. Consider the set of time-string pairs $S = \{(i, x_j) : j \geq i, 0 \leq i \leq T(n)\}$. By construction, the oracle queries described above and those of $M^A_{T(n)}$ and $M^A_{T(n)+1}$ differ only on the set S. Since each x_j, for $j \geq i$, may be thought of having been randomly chosen during step j, after the superposition of oracle queries to be performed has already been written on the oracle tape, the expected query magnitude of any pair in S is at most $1/2^n$. Hence, being α the sum of the query magnitudes after step $T(n) \geq 4$ for the elements of S,

$$E[\alpha] \leq \frac{|S|}{2^n} = \frac{\binom{T(n)+1}{2}}{2^n} \leq \frac{(T(n))^2}{2^n}.$$

Let ε be a random variable such that $\alpha = \frac{\varepsilon^2}{200R^2T(n)}$. Then by Theorem 7, and making a variable substitution of $\varepsilon = \frac{\varepsilon'}{10R}$, we have $d_{VT}(q,v) \leq \varepsilon$ and $d_{VT}(q,w) \leq \varepsilon$. We showed above that $E[\varepsilon^2/200R^2T(n)] = E[\alpha] \leq (T(n))^2/2^n$. But $E\left[\varepsilon/R\sqrt{200T(n)}\right]^2 \leq E\left[\varepsilon^2/200R^2T(n)\right]$. Therefore,

$$\begin{aligned} E[\varepsilon] &= \sqrt{200R^2T(n)}\, E\left[\frac{\varepsilon}{\sqrt{200R^2T(n)}}\right] \\ &\leq \sqrt{200R^2T(n)\, E\left[\frac{\varepsilon^2}{200R^2T(n)}\right]} \leq \sqrt{200R^2T(n)\frac{(T(n))^2}{2^n}}\,. \end{aligned}$$

As $T(n) \leq \frac{2^{n/5}}{200}$ and $R \leq \frac{2^{n/5}}{200}$, we have

$$E[\varepsilon] \leq \sqrt{200R^2T(n)\frac{(T(n))^2}{2^n}} \leq \sqrt{\frac{200}{200^5}} < \frac{1}{100}\,.$$

Therefore $d_{VT}(q,v) \leq E[\varepsilon] < 1/100$ and $d_{VT}(q,w) \leq E[\varepsilon] < 1/100$. It follows that $d_{VT}(v,w) < 1/50$. $\Diamond$

By Markov's bound, if $d_{VT}(v,w) \leq 1/50$, then $\Pr[d_{VT}(v,w) \leq 2/25] \geq 3/4$, i.e. with probability at least $3/4$ we have $d_{VT}(v,w) \leq 2/25 < 1/3$, meaning that both $M^{A_{T(n)}}$ and $M^{A_{T(n)-1}}$ either accept or reject 1^n.

As stated in Claim 2.1, the probability that the string 1^n belongs to exactly one of the languages $L_{A_{T(n)}}$ and $L_{A_{T(n)-1}}$ is $1/2$. Hence, with probability at least $3/4 - 1/2 = 1/4$, either $M^{A_{T(n)}}$ or $M^{A_{T(n)-1}}$ gives the wrong answer on input 1^n. Since $A_{T(n)}$ and $A_{T(n)-1}$ are chosen from the same distribution, $M^{A_{T(n)}}$ is the machine to give the wrong answer with probability at least $1/8$. This leads to the conclusion that M decides L_A with probability 0 for a uniformly random permutation oracle A. $\square$

References

[1] S. Aaronson, A. Bouland, J. Fitzsimons, M. Lee, The space "just above" BQP, in: Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, 2016, pp. 271–280. `doi:10.1145/2840728.2840739`.

[2] S. Aaronson, A. Bouland, J. Fitzsimons, M. Lee, The space "just above" BQP, Preprint (2014). `doi:10.48550/arXiv.1412.6507`.

[3] S. Aaronson, PDQP/qpoly = ALL, Quantum Information & Computation 18 (11-12) (2018) 901–909, doi: 10.5555/3370220.3370221.

[4] R. Hiromasa, A. Mizutani, Y. Takeuchi, S. Tani, Rewindable quantum computation and its equivalence to cloning and adaptive postselection, Preprint (2022). doi:10.48550/arXiv.2206.05434.

[5] S. Aaronson, S. Grewal, V. Iyer, S. C. Marshall, R. Ramachandran, PDQMA = DQMA = NEXP: QMA with hidden variables and non-collapsing measurements (2024). doi:10.48550/arXiv.2403.02543.

[6] S. Aaronson, Quantum computing and hidden variables, Physical Review A 71 (3) (2005) 032325. doi:10.1103/PhysRevA.71.032325.

[7] D. Gutfreund, R. Shaltiel, A. Ta-Shma, Uniform hardness versus randomness tradeoffs for Arthur-Merlin games, Computational Complexity 12 (3) (2003) 85–130. doi:10.1007/s00037-003-0178-7.

[8] R. Raz, A. Tal, Oracle separation of BQP and PH, Journal of the ACM 69 (4) (2022). doi:10.1145/3530258.

[9] L. Fortnow, J. Rogers, Complexity limitations on quantum computation, J. Comput. Syst. Sci. 59 (2) (1999) 240–252. doi:10.1006/jcss.1999.1651.

[10] S. Menda, J. Watrous, Oracle separations for quantum statistical zero-knowledge, Preprint (2018). doi:10.48550/arXiv.1801.08967.

[11] R. Beigel, H. Buhrman, L. Fortnow, NP might not be as easy as detecting unique solutions, in: Proceedings of the thirtieth annual ACM symposium on Theory of computing, 1998, pp. 203–208. doi:10.1145/276698.276737.

[12] C. Tamon, T. Yamakami, Quantum computation relative to oracles, in: Unconventional Models of Computation, UMC'2K, Springer, 2001, pp. 273–288. doi:10.1007/978-1-4471-0313-4_20.

[13] C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani, Strengths and weaknesses of quantum computing, SIAM Journal on Computing 26 (5) (1997) 1510–1523. doi:10.1137/S0097539796300933.

[14] S. Aaronson, $\mathsf{P} \overset{?}{=} \mathsf{NP}$, Springer International Publishing, Cham, 2016, pp. 1–122. `doi:10.1007/978-3-319-32162-2_1`.
URL https://doi.org/10.1007/978-3-319-32162-2_1

[15] T. Baker, J. Gill, R. Solovay, Relativizations of the $\mathsf{P} \overset{?}{=} \mathsf{NP}$ question, SIAM Journal on Computing 4 (4) (1975) 431–442. `doi:10.1137/0204037`.

[16] Y. Gurevich, A. Blass, Quantum circuits with classical channels and the principle of deferred measurements, Theoretical Computer Science 920 (2022) 21–32. `doi:10.1016/j.tcs.2022.02.002`.

Appendix A. Proof of Lemma 4

Lemma 4 (adapted from Bennett et al., 1997 [13]). *Let M^A be an* naCQP *algorithm with an oracle A as in Section 2. For some fixed t, let T be the number of copies of U_f amongst gates $U_1, \ldots, U_t$, and, for $\varepsilon > 0$, let $F \subseteq [1,t] \times \Sigma^*$*

1. *for each $(i,y) \in F$, U_i is a copy of U_f;*
2. *$\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \leq \varepsilon^2/2T$ (recall the definition of $|\phi_i\rangle$ in Section 2).*

Now suppose that the answer to each query $(i,y) \in F$ is modified to some arbitrary fixed bit $a_{i,y}$, being these answers not necessarily consistent with an oracle. Let $|\phi'_i\rangle$ be defined as $|\phi_i\rangle$, but with respect to oracle A modified as above. Then, $\||\phi_t\rangle - |\phi'_t\rangle\| \leq \varepsilon$.

Proof. Let $|\phi_i\rangle$ be the quantum state of M^A at time i given the input x and let $U_1, \ldots, U_t$ be the quantum gates of M^A. Let A_i be an oracle such that if $(i,y) \in F$ then $A_i(y) = a_{i,y}$ and if $(i,y) \notin F$ then $A_i(y) = A(y)$. Let U'_i be the gate of M^{A_i}, we define $|E_i\rangle$ as the error in the i-th step caused by exchanging the oracle A for the oracle A_i, thus,

$$|E_i\rangle = U'_{i+1}|\phi_i\rangle - U_{i+1}|\phi_i\rangle \tag{A.1}$$

or

$$U_{i+1}|\phi_i\rangle = U'_{i+1}|\phi_i\rangle - |E_i\rangle\ . \tag{A.2}$$

Considering that $|\phi_t\rangle = U_t\,|\phi_{t-1}\rangle$, we have, using (A.2),

$$\begin{aligned}
|\phi_t\rangle &= U'_t\,|\phi_{t-1}\rangle - |E_{t-1}\rangle \\
&= U'_t(U_{t-1}\,|\phi_{t-2}\rangle) - |E_{t-1}\rangle \\
&= U'_t(U'_{t-1}\,|\phi_{t-2}\rangle - |E_{t-2}\rangle) - |E_{t-1}\rangle \\
&= U'_tU'_{t-1}\,|\phi_{t-2}\rangle - U'_t\,|E_{t-2}\rangle - |E_{t-1}\rangle \\
&= U'_tU'_{t-1}\cdots U'_1\,|\phi_0\rangle - (U'_tU'_{t-1}\cdots U'_1\,|E_0\rangle + \cdots + U'_t\,|E_{t-2}\rangle + |E_{t-1}\rangle) \\
&= |\phi'_t\rangle - (U'_tU'_{t-1}\cdots U'_1\,|E_0\rangle + \cdots + U'_t\,|E_{t-2}\rangle + |E_{t-1}\rangle)\,;
\end{aligned}$$

then

$$\begin{aligned}
\||\phi_t\rangle - |\phi'_t\rangle\| &= \|U'_tU'_{t-1}\cdots U'_1\,|E_0\rangle + \cdots + U'_t\,|E_{t-2}\rangle + |E_{t-1}\rangle\| \\
&\le \|U'_tU'_{t-1}\cdots U'_1\,|E_0\rangle\| + \cdots + \|U'_t\,|E_{t-2}\rangle\| + \||E_{t-1}\rangle\|\,.
\end{aligned}$$

As all operators U'_i are unitary, $\|U'_{t-1}\cdots U'_i\,|E_i\rangle\| = \||E_i\rangle\|$, we have

$$\||\phi_t\rangle - |\phi_t{}'\rangle\,\| \le \sum_{i=0}^{t-1}\||E_i\rangle\|\,. \tag{A.3}$$

From (A.1), using the triangle inequality we have:

$$\begin{aligned}
\||E_i\rangle\|^2 &\le \|U'_i\,|\phi_i\rangle\|^2 + \|U_i\,|\phi_i\rangle\|^2 \\
&\le \||\phi_i\rangle\|^2 + \||\phi_i\rangle\|^2 = 2\||\phi_i\rangle\|^2\,;
\end{aligned} \tag{A.4}$$

Since $\||E_i\rangle\|^2$ is the result of the error caused by changing the result of the query $(i, y) \in F$, we can rewrite (A.4) as:

$$\||E_i\rangle\|^2 \le 2q_y(\phi_i)\,. \tag{A.5}$$

So, by Condition 2 in the statement, we have

$$\sum_{i=0}^{t-1}\||E_i\rangle\|^2 \le 2\sum_{(i,y)\in F} q_y(|\phi_i\rangle) \le \frac{\epsilon^2}{T}\,. \tag{A.6}$$

Using (A.3) and (A.6) we get

$$\||\phi_t\rangle - |\phi_t{}'\rangle\,\|^2 \le \left\|\sum_{i=0}^{t-1}\||E_i\rangle\|\right\|^2 \le T\sum_{(i,y)\in F}\||E_i\rangle\|^2 = T\frac{\epsilon^2}{T} = \epsilon^2\,.$$

Therefore, $\||\phi_t\rangle - |\phi'_t\rangle\| \le \epsilon$. □