

Quantum and Classical Complexity Classes: Separations, Collapses, and Closure Properties*

Holger Spakowski[†]

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
spakowsk@cs.uni-duesseldorf.de

Mayur Thakur[‡]

Department of Computer Science
University of Missouri–Rolla
Rolla, MO 65409, USA
thakurk@umr.edu

Rahul Tripathi[§]

Department of Computer Science
University of Rochester
Rochester, NY 14627, USA
rahult@cs.rochester.edu

Abstract

We study the complexity of quantum complexity classes such as EQP, BQP, and NQP (quantum analogs of P, BPP, and NP, respectively) using classical complexity classes such as ZPP, WPP, and $C=P$. The contributions of this paper are threefold. First, via oracle constructions, we show that no relativizable proof technique can improve the best known classical upper bound for BQP ($BQP \subseteq AWPP$ [FR99]) to $BQP \subseteq WPP$ and the best known classical lower bound for EQP ($P \subseteq EQP$) to $ZPP \subseteq EQP$. Second, we prove that there are oracles A and B such that, relative to A , $coRP$ is immune to NQP and relative to B , BQP is immune to $P^{C=P}$. Extending a result of de Graaf and Valiant [dGV02], we construct a relativized world where EQP is immune to $Mod_p^k P$. Third, motivated by the fact that counting classes (e.g., LWPP, AWPP, etc.) are the best known classical upper bounds on quantum complexity classes, we study properties of these counting classes. We prove that WPP is closed under polynomial-time truth-table reductions, while we construct an oracle relative to which WPP is not closed under polynomial-time Turing reductions. The latter result implies that proving the equality of the similar appearing classes LWPP and WPP would require nonrelativizable proof techniques. We also prove that both AWPP and APP are closed under \leq_T^{UP} reductions. We use closure properties of WPP and AWPP to

*A preliminary version of this paper was presented at the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '03). Research supported in part by grant NSF-INT-9815095/DAAD-315-PPP-gü-ab, a grant from the DAAD, and by DFG project RO 1202/9-1.

[†]Work done in part while visiting the University of Rochester.

[‡]Work done in part while the author was affiliated with the University of Rochester.

[§]Corresponding author.

prove interesting consequences, in terms of the complexity of the polynomial-hierarchy, of the following hypotheses: $\text{NQP} \subseteq \text{BQP}$ and $\text{EQP} = \text{NQP}$.

Keywords: computational complexity, quantum complexity classes, gap-definable counting classes, relativization theory, reduction closure properties.

1 Introduction

Quantum complexity classes such as EQP, BQP [BV97] (quantum analogs, respectively, of P and BPP [Gil77]), and NQP [ADH97] (quantum analog of NP) are defined using quantum Turing machines [BV97], the quantum analog of classical Turing machines. EQP is the class of languages L accepted by a quantum Turing machine M running in polynomial time such that, for each $x \in \Sigma^*$, if $x \in L$, then the probability that $M(x)$ accepts is 1, and if $x \notin L$, then the probability that $M(x)$ accepts is 0. BQP is the class of languages L accepted by a quantum Turing machine M running in polynomial time such that, for each $x \in \Sigma^*$, if $x \in L$, then the probability that $M(x)$ accepts is at least $2/3$, and if $x \notin L$, then the probability that $M(x)$ accepts is at most $1/3$. NQP is the class of languages L accepted by a quantum Turing machine M running in polynomial time such that, for each $x \in \Sigma^*$, $x \in L$ if and only if the probability that $M(x)$ accepts is nonzero.

Quantum complexity classes represent the computational power of quantum computers. Some fundamental computational problems—for example, factoring, discrete logarithm [Sho97], Pell’s equation, and the principal ideal problem [Hal02]—are not believed to be in BPP, and yet have been shown to be in BQP. One of the key issues in quantum complexity theory is studying the relationship between classical and quantum complexity classes. The inclusion relationships of BQP with some natural classical complexity classes are known. Bernstein and Vazirani [BV97] showed that $\text{BPP} \subseteq \text{BQP} \subseteq \text{P}^{\#P}$. Adleman, DeMarrais, and Huang [ADH97] improved that to $\text{BQP} \subseteq \text{PP}$. Fortnow and Rogers [FR99] showed that the investigation of counting classes can give us insights into the classical complexity of quantum complexity classes. In particular, they studied the complexity of BQP using gap-definable counting classes [FFK94]. (See section 2 for definitions of complexity classes not defined in this section.) Loosely speaking, gap-definable counting classes capture the power of computing via counting the gap (i.e., difference) between the number of accepting and rejecting paths in a nondeterministic polynomial-time Turing machine. Fortnow and Rogers proved that $\text{BQP} \subseteq \text{AWPP}$, where AWPP is a gap-definable counting class. Since $\text{AWPP} \subseteq \text{PP}$, this gives a better upper bound for BQP than that of Adleman, DeMarrais, and Huang. Thus, the best known lower and upper bounds for BQP in terms of classical complexity classes are, respectively, BPP and AWPP: $\text{BPP} \subseteq \text{BQP} \subseteq \text{AWPP} \subseteq \text{PP}$. Similarly, the best known classical lower and upper bounds for EQP are, respectively, P and LWPP: $\text{P} \subseteq \text{EQP} \subseteq \text{LWPP} \subseteq \text{AWPP} \subseteq \text{PP}$. The quantum complexity class NQP coincides with coC=P [FGHP98,YY99].

In light of these connections between quantum and counting complexity classes, it is natural to ask if there are counting (or other classical) complexity classes that give better lower (or upper) bounds for BQP. More formally, is there a counting class \mathcal{C} that lies

nontrivially between BPP and BQP? Is there a counting class \mathcal{D} that lies nontrivially between BQP and AWPP? Similar questions can be asked about EQP. Unfortunately, these questions are often difficult and out of reach of relativizable proof techniques. Green and Pruim [GP01] constructed an oracle relative to which $\text{EQP} \not\subseteq \text{P}^{\text{NP}}$ and thus they showed that proving $\text{EQP} \subseteq \text{P}^{\text{NP}}$ is outside the scope of relativizable proof techniques. Furthermore, for each prime p and integer $k \geq 1$, de Graaf and Valiant [dGV02] constructed an oracle relative to which $\text{EQP} \not\subseteq \text{Mod}_{p^k}\text{P}$.

In this paper, we use counting classes to study the *relativized* complexity of EQP, BQP, and NQP. In particular, we study the classes EQP, BQP, and NQP by separating counting classes relative to an oracle. We construct oracles A and B such that $\text{ZPP}^A \not\subseteq \text{WPP}^A$ and $\text{RP}^B \not\subseteq \text{C}_{=} \text{P}^B$. It follows immediately from known inclusions that $\text{ZPP}^A \not\subseteq \text{EQP}^A$, $\text{BQP}^A \not\subseteq \text{WPP}^A$, and $\text{coRP}^B \not\subseteq \text{NQP}^B$. Note that $\text{WPP} \subseteq \text{AWPP}$, $\text{P} \subseteq \text{ZPP} \subseteq \text{BPP} \subseteq \text{BQP}$, and $\text{EQP} \subseteq \text{LWPP} \subseteq \text{WPP} \subseteq \text{C}_{=} \text{P}$. In fact, WPP is the largest known natural gap-definable subclass of AWPP and ZPP is the smallest known natural probabilistic complexity class that contains P. Thus, even though NQP ($= \text{coC}_{=} \text{P}$) contains RP, and hence contains ZPP ($= \text{RP} \cap \text{coRP}$), in every relativized world, using relativizable proof techniques it is impossible to show that NQP contains all sets in coRP.

The oracle separations of counting classes mentioned above, for example, $\text{RP}^B \not\subseteq \text{C}_{=} \text{P}^B$, leaves open the possibility that each set in RP^B can be approximated by a set in $\text{C}_{=} \text{P}^B$ in the following sense: For each infinite set $L \in \text{RP}^B$, there exists an infinite subset $L' \subseteq L$ such that $L' \in \text{C}_{=} \text{P}^B$. A strong (or immunity) separation of RP^B from $\text{C}_{=} \text{P}^B$ will preclude this possibility. Strong oracle separations have been studied in many different settings, e.g., for the polynomial hierarchy [Ko90, Bru92], for the boolean hierarchy over RP [BJY90], and for counting classes [Rot99]. We prove strong oracle separations between counting classes, and from these we get strong oracle separations involving quantum and counting complexity classes. For example, we prove that there are oracles A and A' such that RP^A is $\text{C}_{=} \text{P}^A$ -immune and $\text{BPP}^{A'}$ is $\text{P}^{\text{C}_{=} \text{P}^{A'}}$ -immune. That implies that coRP^A is NQP^A -immune and $\text{BQP}^{A'}$ is $\text{P}^{\text{C}_{=} \text{P}^{A'}}$ -immune. For each prime p and integer $k \geq 1$, we construct an oracle relative to which EQP is Mod_{p^k}P -immune. This extends an oracle separation of EQP from Mod_{p^k}P by de Graaf and Valiant [dGV02].

Results by Fortnow and Rogers [FR99], de Graaf and Valiant [dGV02], and those of this paper show the connection between quantum and counting complexity classes. This motivates the investigation of reduction closure properties of these counting classes. Fenner, Fortnow, and Kurtz [FFK94] showed that the counting classes SPP and LWPP are closed under polynomial-time Turing reductions. (In fact, they proved that $\text{SPP}^{\text{SPP}} = \text{SPP}$ and $\text{SPP}^{\text{LWPP}} = \text{LWPP}$.) They asked whether the same holds for WPP. We give a partial answer to their question. We prove that WPP is closed under polynomial-time truth-table reductions, and show that this cannot be improved to closure under polynomial-time Turing reductions using relativizable proof techniques: There is an oracle A such that $\text{P}^{\text{WPP}^A} \not\subseteq \text{WPP}^A$. Thus, it follows that relative to oracle A , WPP strictly contains LWPP. For the counting classes AWPP and APP, we prove a stronger closure property, namely that both AWPP and APP are closed under \leq_T^{UP} (unambiguous nondeterministic polynomial-

time Turing) reductions.

Vyalyi [Vya03] proved using Toda’s theorem [Tod91] that QMA, the class of languages such that a “yes” answer can be verified by a 1-round quantum interactive proof, is unlikely to contain PP, since if it does then PP contains PH. That paper implicitly contains the observation that Toda’s theorem [Tod91] implies $\text{PH} \subseteq \text{UP}^{\text{C}=\text{P}}$. Using this result and the reduction closure results mentioned above, we prove consequences of the “ $\text{NQP} \subseteq \text{BQP}$ ” and “ $\text{NQP} \subseteq \text{EQP}$ ”¹ hypotheses. Note that these hypotheses are quantum counterparts of the “ $\text{NP} \subseteq \text{BPP}$ ” and the “ $\text{NP} \subseteq \text{P}$ ” hypotheses. Zachos [Zac88] proved that if $\text{NP} \subseteq \text{BPP}$, then $\text{PH} \subseteq \text{BPP}$. We prove that if $\text{NQP} \subseteq \text{BQP}$, then $\text{PH} \subseteq \text{AWPP}$, and if $\text{NQP} \subseteq \text{EQP}$, then $\text{PH} \subseteq \text{EQP}$ and so, as an immediate consequence, $\text{PH} \subseteq \text{LWPP}$. Since it is unlikely that the complexity of PH is restricted to classes such as AWPP and LWPP, and since AWPP and LWPP are low for PP, our results suggest that these hypotheses are less likely to be true.

The paper is organized as follows. Section 2 contains the definitions and notations. In Section 3, we prove relativized separations between counting classes, which in turn lead to relativized separations between quantum and counting classes. In Section 4, we prove our immunity separation results. In Section 5, we prove the closure and collapse results. Section 6 contains some open problems.

2 Preliminaries

Let \mathbb{C} , \mathbb{N} , \mathbb{Q} , and \mathbb{Z} denote the set of complex numbers, the set of nonnegative integers, the set of rational numbers, and the set of integers, respectively. Our alphabet is $\Sigma = \{0, 1\}$. Σ^* denotes the set of all finite length strings over the alphabet Σ and for every $n \in \mathbb{N}$, Σ^n denotes the set of all strings of length n in Σ^* . For any $n \in \mathbb{N}$ and any $x \in \Sigma^*$, $x\Sigma^n = \{xw \mid w \in \Sigma^n\}$. For any $x \in \Sigma^*$, $|x|$ denotes the length of the string x , while $|x|_0$ and $|x|_1$ denote, respectively, the number of 0’s and the number of 1’s in x . For every set $L \subseteq \Sigma^*$, $|L|$ denotes the cardinality of L and let χ_L denote the characteristic function of L , i.e., for each $x \in L$, $\chi_L(x) = 1$ and for each $x \notin L$, $\chi_L(x) = 0$. For any $x \in \Sigma^*$, the integer $\text{num}(x)$ corresponding to string x is defined as the value of the binary number $1x$. Let $\langle \cdot, \dots, \cdot \rangle$ denote a standard, fixed, easily computable, and invertible pairing function.

For general complexity-theoretic background and for the definition of complexity classes such as P, NP, FP etc., we refer the reader to the handbook [HO02]. NPTM stands for “nondeterministic polynomial-time Turing machine” and DPTM stands for “deterministic polynomial-time Turing machine.” For a complexity class \mathcal{C} , $\text{co}\mathcal{C}$ is defined by $\text{co}\mathcal{C} = \{L \mid \bar{L} \in \mathcal{C}\}$. Throughout this paper, for any (nondeterministic or deterministic or quantum) Turing machine N and for any $x \in \Sigma^*$, we use $N(x)$ as a shorthand for “the computation of N on input x .”

A computation path ρ' in an NPTM is a string in Σ^* representing the sequence of nondeterministic guesses. An *augmented* computation path ρ in an oracle NPTM is a string

¹Note that $\text{EQP} \subseteq \text{NQP}$ follows trivially from the definitions of these classes.

$\langle \rho', \sigma \rangle$ such that σ represents the sequence of answers of the oracle to queries along the path ρ' . We say that an augmented computation path $\rho = \langle \rho', \sigma \rangle$ occurs in the computation of $N^A(x)$ if ρ' is a computation path in $N^A(x)$ and σ is consistent with the memberships of all strings queried along ρ' in $N^A(x)$. Given an augmented path $\rho = \langle \rho', \sigma \rangle$, let $\text{naked}(\rho)$ denote ρ' and let $\text{ans}(\rho)$ denote σ . In this paper, we will use “path” to mean both “unaugmented path” and “augmented path” (which sense is being used will be clear from the context).

Given NPTM N , $A \subseteq \Sigma^*$, and $x \in \Sigma^*$, let $\text{PATH}(N^A, x)$ denote the set of augmented computation paths that occur in $N^A(x)$. Let $\text{ACCEPT}(N^A, x)$ denote the set of paths in $\text{PATH}(N^A, x)$ that accept. Given NPTM N , $x \in \Sigma^*$, and augmented path $\rho \in \Sigma^*$, we let $\text{sign}(N, x, \rho) = +1$ if $\text{naked}(\rho)$ with query answers $\text{ans}(\rho)$ is an accepting path in $N(x)$, $\text{sign}(N, x, \rho) = -1$ if $\text{naked}(\rho)$ with query answers $\text{ans}(\rho)$ is a rejecting path in $N(x)$, and $\text{sign}(N, x, \rho) = 0$, otherwise (i.e., if ρ is not a valid augmented path in $N(x)$). Note that the “sign” value does not depend on the oracle because the oracle answers are already included in the augmented path ρ .

Next, we define the notions of counting in nondeterministic Turing machines. Given an NPTM N and a string x in Σ^* , we use $\#\text{acc}_N(x)$ ($\#\text{rej}_N(x)$) to denote the number of accepting (respectively, rejecting) paths of N on x . If N is an oracle NPTM and A is a set, then we use $\#\text{acc}_{N^A}(x)$ ($\#\text{rej}_{N^A}(x)$) to denote the number of accepting (respectively, rejecting) paths of N on x with oracle A . $\#\text{P}$ [Val76] is the class of functions f such that there exists an NPTM N such that, for each $x \in \Sigma^*$, $f(x) = \#\text{acc}_N(x)$. We now define a function gap_N that represents the “gap” between the number of accepting and the number of rejecting paths of N .

Definition 2.1 [FFK94] *If N is an NPTM, define the function $\text{gap}_N : \Sigma^* \rightarrow \mathbb{Z}$ as follows: For all $x \in \Sigma^*$, $\text{gap}_N(x) =_{\text{df}} \#\text{acc}_N(x) - \#\text{rej}_N(x)$. If N is an oracle NPTM, then for every set A , define the function $\text{gap}_{N^A} : \Sigma^* \rightarrow \mathbb{Z}$ as follows: For all $x \in \Sigma^*$, $\text{gap}_{N^A}(x) =_{\text{df}} \#\text{acc}_{N^A}(x) - \#\text{rej}_{N^A}(x)$.*

GapP is the class of all functions f such that there exists an NPTM N such that $f = \text{gap}_N$. We define the gap-definable counting classes [FFK94] that will be used in the rest of the paper.

- Definition 2.2**
1. [Sim75, Gil77] $\text{PP} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[x \in L \iff g(x) > 0]\}$.
 2. [Sim75, Wag86] $\text{C}_{=}\text{P} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[x \in L \iff g(x) = 0]\}$.
 3. [CH90, Her90, BG92] For each $k \geq 2$, $\text{Mod}_k\text{P} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[x \in L \iff g(x) \not\equiv 0 \pmod{k}]\}$. Equivalently, for each $k \geq 2$, $\text{Mod}_k\text{P} = \{L \mid (\exists f \in \#\text{P})(\forall x \in \Sigma^*)[x \in L \iff f(x) \not\equiv 0 \pmod{k}]\}$.
 4. [PZ83, GP86] $\oplus\text{P} = \text{Mod}_2\text{P}$.
 5. [OH93, FFK94] $\text{SPP} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[(x \in L \implies g(x) = 1) \wedge (x \notin L \implies g(x) = 0)]\}$.

6. [FFK94] LWPP = $\{L \mid (\exists g \in \text{GapP})(\exists h \in \text{FP} : 0 \notin \text{range}(h))(\forall x \in \Sigma^*)[(x \in L \implies g(x) = h(0^{|x|})) \wedge (x \notin L \implies g(x) = 0)]\}$.
7. [FFK94] WPP = $\{L \mid (\exists g \in \text{GapP})(\exists h \in \text{FP} : 0 \notin \text{range}(h))(\forall x \in \Sigma^*)[(x \in L \implies g(x) = h(x)) \wedge (x \notin L \implies g(x) = 0)]\}$.

The counting classes AWPP [FFKL03] and APP [Li93] were defined to study the sets that are low for PP. The original definition of these classes included the amplification property.

Definition 2.3 [FFKL03] *A language $L \subseteq \Sigma^*$ is in AWPP if for every polynomial r , there exist a function $g \in \text{GapP}$ and a polynomial p such that, for all $x \in \Sigma^*$,*

$$\begin{aligned} x \in L &\implies 1 - 2^{-r(|x|)} \leq \frac{g(x)}{2^{p(|x|)}} \leq 1, \text{ and} \\ x \notin L &\implies 0 \leq \frac{g(x)}{2^{p(|x|)}} \leq 2^{-r(|x|)}. \end{aligned}$$

Definition 2.4 [Li93] *A language $L \subseteq \Sigma^*$ is in APP if for every polynomial r , there exist $g, h \in \text{GapP}$ such that, for all $n \in \mathbb{N}$ and x with $n \geq |x|$, $h(0^n) > 0$ and*

$$\begin{aligned} x \in L &\implies 1 - 2^{-r(n)} \leq \frac{g(x, 0^n)}{h(0^n)} \leq 1, \text{ and} \\ x \notin L &\implies 0 \leq \frac{g(x, 0^n)}{h(0^n)} \leq 2^{-r(n)}. \end{aligned}$$

Fenner [Fen03] simplified the definition of these classes and showed that AWPP \subseteq APP.

Theorem 2.5 [Fen03]

1. *A language $L \subseteq \Sigma^*$ is in AWPP if and only if there exist a function $g \in \text{GapP}$ and a polynomial p such that, for all $x \in \Sigma^*$,*

$$\begin{aligned} x \in L &\implies 3/4 \leq \frac{g(x)}{2^{p(|x|)}} \leq 1, \text{ and} \\ x \notin L &\implies 0 \leq \frac{g(x)}{2^{p(|x|)}} \leq 1/4. \end{aligned}$$

2. *A language $L \subseteq \Sigma^*$ is in APP if and only if there exist $g, h \in \text{GapP}$, $h > 0$, such that, for all $x \in \Sigma^*$,*

$$\begin{aligned} x \in L &\implies 3/4 \leq \frac{g(x)}{h(0^{|x|})} \leq 1, \text{ and} \\ x \notin L &\implies 0 \leq \frac{g(x)}{h(0^{|x|})} \leq 1/4. \end{aligned}$$

We also define certain classes that we will relate to the gap-definable classes mentioned in Definitions 2.2, 2.3, and 2.4 in this paper.

Definition 2.6 [Val76] *A language $L \subseteq \Sigma^*$ is in UP if there is an NPTM N such that, for all $x \in \Sigma^*$, $x \in L \implies \#acc_N(x) = 1$ and $x \notin L \implies \#acc_N(x) = 0$.*

Definition 2.7 [All86,AR88] A language $L \subseteq \Sigma^*$ is in FewP if there is an NPTM N and a polynomial p such that, for all $x \in \Sigma^*$, $x \in L \implies 1 \leq \#acc_N(x) \leq p(|x|)$ and $x \notin L \implies \#acc_N(x) = 0$.

Definition 2.8 [Gil77] A language $L \subseteq \Sigma^*$ is in RP if there is a polynomial-time predicate R , a polynomial p , and $0 < \epsilon \leq 1$ such that, for all $x \in \Sigma^*$,

$$\begin{aligned} x \in L &\implies \|\{y \mid |y| \leq p(|x|) \wedge R(x, y)\}\| \geq \frac{(1+\epsilon)}{2} \cdot 2^{p(|x|)}, \text{ and} \\ x \notin L &\implies \|\{y \mid |y| \leq p(|x|) \wedge R(x, y)\}\| = 0. \end{aligned}$$

Definition 2.9 [Gil77] $ZPP = RP \cap \text{coRP}$.

Definition 2.10 [Gil77] A language $L \subseteq \Sigma^*$ is in BPP if there is a polynomial-time predicate R , a polynomial p , and $0 < \epsilon \leq 1$ such that, for all $x \in \Sigma^*$,

$$\begin{aligned} x \in L &\implies \|\{y \mid |y| \leq p(|x|) \wedge R(x, y)\}\| \geq \frac{(1+\epsilon)}{2} \cdot 2^{p(|x|)}, \text{ and} \\ x \notin L &\implies \|\{y \mid |y| \leq p(|x|) \wedge R(x, y)\}\| \leq \frac{(1-\epsilon)}{2} \cdot 2^{p(|x|)}. \end{aligned}$$

For background information on quantum complexity theory and for the definition of quantum Turing machine, we recommend [NC00,Gru99]. We define the quantum complexity classes that will be used in this paper.

Definition 2.11 [BV97,ADH97] EQP (BQP, NQP) is the class of all languages $L \subseteq \Sigma^*$ such that there is a polynomial-time quantum Turing machine M such that, for each $x \in \Sigma^*$, $x \in L \implies \Pr[M(x) \text{ accepts}] = 1$ (respectively, $\geq 2/3$, $\neq 0$) and $x \notin L \implies \Pr[M(x) \text{ accepts}] = 0$ (respectively, $\leq 1/3$, $= 0$).

The following proposition gives known inclusion relationships among the classes defined above.

Proposition 2.12 [Fen03,FFK94,FFKL03,FGHP98,FR99,Gil77,KSTT92,YY99] *The following inclusion relations hold relative to all oracles.*

1. $P \subseteq ZPP \subseteq RP \subseteq BPP \subseteq AWPP$.
2. $P \subseteq UP \subseteq \text{FewP} \subseteq \text{SPP} \subseteq \text{LWPP} \subseteq \text{WPP} \subseteq \text{C=P} \subseteq \text{PP}$.
3. $ZPP \subseteq \text{coRP} \subseteq \text{coNP} \subseteq \text{C=P}$.
4. $\text{WPP} \subseteq \text{AWPP} \subseteq \text{APP} \subseteq \text{PP}$.
5. $P \subseteq \text{EQP} \subseteq \text{LWPP} \subseteq \text{WPP} \subseteq \text{coC=P}$.
6. $\text{EQP} \subseteq \text{BQP} \subseteq \text{AWPP}$.
7. $\text{SPP} \subseteq \oplus P$.
8. $\text{FewP} \subseteq \text{NP} \subseteq \text{coC=P} = \text{NQP}$.
9. $\text{BPP} \subseteq \text{BQP}$.

Next we define the reductions used in this paper. We say that $A \leq_T^p B$ (*A polynomial-time Turing reduces to B*) if there exists an oracle DPTM M such that $L(M^B) = A$. We say that $A \leq_{tt}^p B$ (*A polynomial-time truth-table reduces to B*) if there exists a DPTM M and a polynomial-time computable function f such that, for each $x \in \Sigma^*$, there exists an integer m such that

1. $f(x) = \langle q_1, q_2, \dots, q_m \rangle$, and
2. $M(\langle x, \chi_B(q_1), \chi_B(q_2), \dots, \chi_B(q_m) \rangle)$ accepts if and only if $x \in A$.

3 Separating Quantum Classes Using Counting Class Separations

One way to study the power of quantum complexity classes is to search for well-known complexity classes that provide a good lower bound for these quantum classes. The best known lower bound for EQP is P. In fact, EQP is not known to contain even a single problem that is not already known to be in P. Bennett et al. [BBBV97] showed that relative to a random oracle, NP is not contained in EQP with probability one, and relative to a permutation oracle chosen uniformly at random, $\text{NP} \cap \text{coNP}$ is not contained in EQP with probability one. Thus it is interesting to ask the following questions. Are there natural classes between P and $\text{NP} \cap \text{coNP}$ that are contained in EQP? Are there natural classes between P and $\text{NP} \cap \text{coNP}$ that are not contained in EQP in some relativized world? We prove that the latter is true by showing that there is a relativized world where ZPP is not contained in EQP. In fact, we prove a slightly stronger statement. We prove, as the next theorem, that there is an oracle relative to which ZPP is not contained in WPP, a superclass of EQP [FR99]. It is interesting to note that there is an oracle, due to Fortnow [For99], relative to which SPP, a subclass of WPP, strictly contains an infinite polynomial hierarchy. In contrast, our oracle provides a completely different picture of WPP in a relativized world: A world in which WPP sets are not powerful enough to capture a seemingly small subclass, namely ZPP, of NP.

On the Proof Technique: The oracle constructions in Theorems 3.1 and 3.13 use a “gap analog” of the counting technique used by Torán [Tor91] in the relativized separation of counting classes from each other, for example in the construction of an oracle A such that $\text{NP}^A \not\subseteq \text{C}_{=}P^A$. Many of the powerful state-of-the-art oracle construction techniques have the following flavor in proof steps: (1) abstract the oracle construction problem in terms of some combinatorial object, such as a boolean circuit over some base set of gates or a polynomial over some ring, satisfying certain properties, and (2) prove existence of an appropriate oracle segment based on the limitations of the combinatorial object satisfying those properties. Because of this transition in proof from machines and oracles to pure combinatorial objects with implicit dependence on behavior of machines, these oracle construction techniques are more combinatorial than complexity-theoretic in nature. In contrast, the proof techniques by Torán [Tor91] and Beigel [Bei91], which we use in Theorems 3.1, 3.13, and 4.4, do not purely abstract the problem from machines and oracles to combinatorial objects, and depend heavily on the behavior of machines with different

oracles. From the point of view of traditional and perfectionist complexity theorists, the proof techniques by Torán [Tor91] and Beigel [Bei91] might be more appealing, because of being more complexity-theoretic in nature, than the other known combinatorial proof techniques for oracle constructions.

For the above reason, we choose to use the “gap analog” of the proof technique by Torán [Tor91] and Beigel [Bei91] in Theorems 3.1, 3.13, and 4.4. In a subsequent paper [ST04a,ST04b], the first and the third author of this paper used the combinatorial polynomial degree bound technique to prove an extension of Theorem 3.1. They showed that there is a relativized world where $ZPP \not\subseteq WPP^{WPP}$ and claimed that their proof can easily be extended to construct a relativized world where ZPP is not contained in any level k , where $k \geq 1$, of the WPP hierarchy formed by composing WPP with itself up to k levels. It is illuminating to compare the power of the two vastly different proof techniques for the oracle construction in Theorem 3.1.

Theorem 3.1 $(\exists \mathcal{A}) [ZPP^{\mathcal{A}} \not\subseteq WPP^{\mathcal{A}}]$.

Proof For every set B , let $L_B =_{df} \{0^n \mid (\exists w \in \Sigma^n)[0w \in B]\}$. Define predicates “I-Promise” and “II-Promise” as follows.

$$\text{I-Promise}(B, n, k) \equiv \|B \cap 0\Sigma^n\| > k \text{ and } \|B \cap 1\Sigma^n\| = 0, \text{ and}$$

$$\text{II-Promise}(B, n, k) \equiv \|B \cap 0\Sigma^n\| = 0 \text{ and } \|B \cap 1\Sigma^n\| > k.$$

We say that a set B satisfies the promise at length n with threshold k if $[\text{I-Promise}(B, n, k) \vee \text{II-Promise}(B, n, k)]$ is true. Clearly, if B satisfies the promise at each length $n \geq 1$ with threshold 2^{n-1} , then $L_B \in ZPP^B$. We show that there is an oracle \mathcal{A} such that \mathcal{A} satisfies the promise at each length $n \geq 1$ with threshold 2^{n-1} and $L_{\mathcal{A}} \notin WPP^{\mathcal{A}}$.

Let $(N_s, M_s, p_s)_{s \geq 1}$ be an enumeration of all triples such that the first component of the triple is a nondeterministic polynomial-time oracle Turing machine, the second component is a deterministic polynomial-time oracle transducer, the third component is a polynomial, and the running time of both N_s and M_s is bounded by the polynomial p_s regardless of the oracle.

The oracle is constructed in stages. Before the start of stage 1, let $\mathcal{A} := \bigcup_{n \in \mathbb{N}} 0\Sigma^n$. In stage $s \geq 1$, the membership in \mathcal{A} of strings of length $n_s + 1$ is changed. Finally at the end of every stage, \mathcal{A} is assigned as the oracle.

Stage s , where $s > 0$: Let n_s be the smallest integer such that the following hold: $n_s > n_{s-1}$, $2^{n_s} > 4p_s(n_s) + 2$, and no machine considered in previous stages ever queries a string of length greater than n_s . Let $\mathcal{A} := \mathcal{A} - \Sigma^{n_s+1}$. In stage s , we diagonalize against the triple (N_s, M_s, p_s) . Note that the FP function h used in the definition of WPP (see Definition 2.2(7)) is non-zero for all $x \in \Sigma^*$. Therefore, without loss of generality we may assume that M_s never outputs zero with any oracle that satisfies the promise at length n_s with threshold 2^{n_s-1} .

Let T_0 (T_1) denote the set of queries of length $n_s + 1$ asked by $M_s^{\mathcal{A}}(0^{n_s})$ whose first bit is zero (respectively, one). Let the value computed by $M_s^{\mathcal{A}}(0^{n_s})$ be denoted by val . As explained above, we may assume that $val \neq 0$.

(\star) Choose a set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$ and one of the following conditions is satisfied:

$$\text{I-Promise}(B, n_s, 2^{n_s-1}) \quad \text{and} \quad \text{gap}_{N_s^{\mathcal{A} \cup B}}(0^{n_s}) \neq \text{val}, \quad \text{or} \quad (3.i)$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1}) \quad \text{and} \quad \text{gap}_{N_s^{\mathcal{A} \cup B}}(0^{n_s}) \neq 0. \quad (3.ii)$$

Let $\mathcal{A} := \mathcal{A} \cup B$ and move to stage $s + 1$.

End of Stage

Obviously, the construction guarantees that $L_{\mathcal{A}} \in \text{ZPP}^{\mathcal{A}}$ but $L_{\mathcal{A}} \notin \text{WPP}^{\mathcal{A}}$. The feasibility of the construction follows from the following claim.

Claim 1 *For each $s \geq 1$, there exists a set B satisfying (\star).*

Proof of Claim 1. In order to prove the above claim, we first describe a combinatorial tool similar to the one used by Torán [Tor91] to separate NP from C=P. Torán used his Q -notation to count the number of accepting paths that have certain restrictions on the queries asked along them. Our Q -notation represents the gap between the number of accepting and rejecting paths that have similar restrictions on the queries asked along them.

Recall from the definition of “sign” given in Section 2 that $\text{sign}(N, x, \rho)$ is 0, +1, or -1 , if, respectively, ρ is not a valid augmented path in $N(x)$, $N(x)$ on path $\text{naked}(\rho)$ with query answers $\text{ans}(\rho)$ accepts, or $N(x)$ on path $\text{naked}(\rho)$ with query answers $\text{ans}(\rho)$ rejects.

Definition 3.2 *For every set $E \subseteq \Sigma^{n_s+1}$ and for each $a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_\ell \in \Sigma^*$, let $Q_{a_1, \dots, a_k, (b_1, \dots, b_\ell)}^E = \sum_{\rho \in R} \text{sign}(N_s, 0^{n_s}, \rho)$, where $R = \{\rho \mid \rho \in \text{PATH}(N_s^{\mathcal{A} \cup E}, 0^{n_s}) \text{ and } N_s^{\mathcal{A} \cup E}(0^{n_s}) \text{ on path } \text{naked}(\rho) \text{ queries each string in } \{a_1, a_2, \dots, a_k\} \text{ and } N_s^{\mathcal{A} \cup E}(0^{n_s}) \text{ on path } \text{naked}(\rho) \text{ does not query any string in } \{b_1, b_2, \dots, b_\ell\}\}$.*

Thus, for any strings $w_1, w_2, \dots, w_k, w'_1, w'_2, \dots, w'_\ell$, each accepting path of $N_s^{\mathcal{A} \cup E}(0^{n_s})$ that queries all of w_1, w_2, \dots, w_k and none of $w'_1, w'_2, \dots, w'_\ell$ has a contribution of +1 in $Q_{w_1, w_2, \dots, w_k, (w'_1, w'_2, \dots, w'_\ell)}^E$, and each rejecting path of $N_s^{\mathcal{A} \cup E}(0^{n_s})$ that queries all of w_1, w_2, \dots, w_k and none of $w'_1, w'_2, \dots, w'_\ell$ has a contribution of -1 in $Q_{w_1, w_2, \dots, w_k, (w'_1, w'_2, \dots, w'_\ell)}^E$.

The following lemma, which we state without a proof, is a “gap analog” of Lemma 5.2 by Torán [Tor91]. It says that the “ Q ” value corresponding to paths querying w_1, \dots, w_k can be split into 2 parts: (a) contribution from paths querying w_1, \dots, w_k that also query w_{k+1} and (b) contribution from paths querying w_1, \dots, w_k that do not query w_{k+1} .

Lemma 3.3 *For every set $E \subseteq \Sigma^{n_s+1}$ and for all $w_1, \dots, w_{k+1} \in \Sigma^{n_s+1}$, the following equality holds:*

$$Q_{w_1, \dots, w_k}^E = Q_{w_1, \dots, w_{k+1}}^E + Q_{w_1, \dots, w_k, (w_{k+1})}^E.$$

The following definition is similar to the one by Torán [Tor91] with the only exception that the Q -terms have the “gap” meaning as in Definition 3.2.

Definition 3.4 *For any $E, D \subseteq \Sigma^{n_s+1}$ with $E \cap D = \emptyset$,*

$$J_D^E = \sum_{i=0}^{\|D\|} (-1)^i \sum_{\substack{A \subseteq D \\ \|A\|=i}} Q_D^{E \cup A}.$$

Intuitively, $E \subseteq \Sigma^{n_s+1}$ represents an oracle segment and $D \subseteq \Sigma^{n_s+1}$ represents a set of query strings, which is disjoint from E , in the expression J_D^E . The expression J_D^E is an arithmetic sum of “ Q ” values, where the “ Q ” values correspond to oracles $E \cup A$, for all $A \subseteq D$, and correspond to computation paths with a fixed query set D . For example, for a set $D = \{w_1, w_2, w_3\}$ and an arbitrary set $E \subseteq \Sigma^{n_s+1} - \{w_1, w_2, w_3\}$,

$$J_D^E = Q_D^E - (Q_D^{E \cup \{w_1\}} + Q_D^{E \cup \{w_2\}} + Q_D^{E \cup \{w_3\}}) + (Q_D^{E \cup \{w_1, w_2\}} + Q_D^{E \cup \{w_2, w_3\}} + Q_D^{E \cup \{w_1, w_3\}}) - Q_D^{E \cup \{w_1, w_2, w_3\}}.$$

For notational convenience, we use the following shorthands: For every $E \subseteq \Sigma^{n_s+1}$, let $Q^E =_{df} \text{gap}_{N_s^{A \cup E}}(0^{n_s})$ and $J^E =_{df} J_\emptyset^E = Q^E$.

Lemma 3.5 (See [Tor91], Lemma 5.3.) *For every sequence of words w_1, \dots, w_k, w_{k+1} in Σ^{n_s+1} and for every set $E \subseteq \Sigma^{n_s+1}$ with $E \cap \{w_1, \dots, w_k, w_{k+1}\} = \emptyset$, the following holds:*

$$J_{w_1, \dots, w_k}^{E \cup \{w_{k+1}\}} = J_{w_1, \dots, w_k}^E - J_{w_1, \dots, w_k, w_{k+1}}^E.$$

Proof All properties of Q and J used in the proof of Lemma 5.3 in [Tor91] carry over to our gap versions of Q and J . Hence, Torán’s proof works also for our lemma. \blacksquare

Suppose that in stage s no set satisfying (\star) exists. Then, for every set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$,

$$\text{I-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N_s^{A \cup B}}(0^{n_s}) = \text{val}, \text{ and} \quad (3.iii)$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N_s^{A \cup B}}(0^{n_s}) = 0. \quad (3.iv)$$

Lemma 3.6 *For each $s \geq 1$ and every $t \in \{0, \dots, 2^{n_s-1}\}$, if for every set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$, it holds that*

$$\text{I-Promise}(B, n_s, t) \implies \text{gap}_{N_s^{A \cup B}}(0^{n_s}) = \text{val},$$

then the following hold:

1. *For every nonempty sequence of words $0w_1, \dots, 0w_k$ such that $\{0w_1, \dots, 0w_k\} \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$ and for every set R that satisfies $R \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$, $\text{I-Promise}(R, n_s, t)$, and $\{0w_1, \dots, 0w_k\} \cap R = \emptyset$, it holds that $J_{0w_1, \dots, 0w_k}^R = 0$.*
2. *For every set C_t such that $C_t \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$, $\|C_t\| = t$, and $\{0w_1, \dots, 0w_k\} \cap C_t = \emptyset$, it holds that $J_{0w_1, \dots, 0w_k}^{C_t} = \text{gap}_{N_s^{A \cup C_t}}(0^{n_s}) - \text{val}$.*

Proof By hypothesis and by using the definition of J , for every set R such that $R \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$ and $\text{I-Promise}(R, n_s, t)$ is true, the following holds: $J^R = Q^R = \text{val}$. By Lemma 3.5,

$$J_{0w_1, \dots, 0w_{k+1}}^R = J_{0w_1, \dots, 0w_k}^R - J_{0w_1, \dots, 0w_k}^{R \cup \{0w_{k+1}\}}.$$

We now prove (1) by induction on k .

For $k = 1$, $J_{0w_1}^R = J^R - J^{R \cup \{0w_1\}} = \text{val} - \text{val} = 0$.

For $k > 1$, $J_{0w_1, \dots, 0w_k}^R = J_{0w_1, \dots, 0w_{k-1}}^R - J_{0w_1, \dots, 0w_{k-1}}^{R \cup \{0w_k\}}$, where by induction hypothesis both terms are 0.

We prove (2) also by induction on k .

For $k = 1$,

$$J_{0w_1}^{C_t} = Q_{0w_1}^{C_t} - Q_{0w_1}^{C_t \cup \{0w_1\}} = Q^{C_t} - Q_{(0w_1)}^{C_t} - Q^{C_t \cup \{0w_1\}} + Q_{(0w_1)}^{C_t \cup \{0w_1\}} = Q^{C_t} - Q^{C_t \cup \{0w_1\}},$$

where the third equality follows from the fact that, by Definition 3.2, $Q_{(0w_1)}^{C_t} = Q_{(0w_1)}^{C_t \cup \{0w_1\}}$.

Thus,

$$J_{0w_1}^{C_t} = \text{gap}_{N_s^{\mathcal{A} \cup C_t}}(0^{n_s}) - \text{val},$$

using the definition of C_t and using the fact that I-Promise($C_t \cup \{0w_1\}, n_s, t$) is true. For $k > 1$, by Lemma 3.5,

$$J_{0w_1, \dots, 0w_k}^{C_t} = J_{0w_1, \dots, 0w_{k-1}}^{C_t} - J_{0w_1, \dots, 0w_{k-1}}^{C_t \cup \{0w_k\}}.$$

By (1),

$$J_{0w_1, \dots, 0w_{k-1}}^{C_t \cup \{0w_k\}} = 0.$$

Also, by the induction hypothesis,

$$J_{0w_1, \dots, 0w_{k-1}}^{C_t} = \text{gap}_{N_s^{\mathcal{A} \cup C_t}}(0^{n_s}) - \text{val}.$$

Thus,

$$J_{0w_1, \dots, 0w_k}^{C_t} = \text{gap}_{N_s^{\mathcal{A} \cup C_t}}(0^{n_s}) - \text{val}. \quad \blacksquare$$

Lemma 3.7 For each $s \geq 1$ and every $t \in \{0, \dots, 2^{n_s-1}\}$, if for every set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$, it holds that I-Promise(B, n_s, t) \implies $\text{gap}_{N_s^{\mathcal{A} \cup B}}(0^{n_s}) = \text{val}$,

then for every set C_t such that $C_t \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$ and $\|C_t\| = t$, it holds that $\text{gap}_{N_s^{\mathcal{A} \cup C_t}}(0^{n_s}) = \text{val}$.

Proof Let C_t be an arbitrary set such that $C_t \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$ and $\|C_t\| = t$. Suppose the hypothesis of the lemma holds but $\text{gap}_{N_s^{\mathcal{A} \cup C_t}}(0^{n_s}) \neq \text{val}$. Then, by Lemma 3.6(2), for any nonempty sequence of words $0w_1, \dots, 0w_k$, where $\{0w_1, \dots, 0w_k\} \subseteq \overline{C_t} \cap \overline{T_0} \cap 0\Sigma^{n_s}$, it holds that $J_{0w_1, \dots, 0w_k}^{C_t} \neq 0$. Let $0w_1, \dots, 0w_{p_s(n_s)+1}$ be a sequence of words such that $\{0w_1, \dots, 0w_{p_s(n_s)+1}\} \subseteq \overline{C_t} \cap \overline{T_0} \cap 0\Sigma^{n_s}$. (Such a sequence of words exists since $2^{n_s} > 4p_s(n_s)+2$.) Since the running time of $N_s^{(\cdot)}(0^{n_s})$ is bounded by $p_s(n_s)$, $N_s^{(\cdot)}(0^{n_s})$ can make at most $p_s(n_s)$ queries to the oracle on every computation path. Therefore, $J_{0w_1, \dots, 0w_{p_s(n_s)+1}}^{C_t} = 0$. This follows from the fact that $J_{0w_1, \dots, 0w_{p_s(n_s)+1}}^{C_t}$ is the sum of sign values associated with computation paths of $N_s^{(\cdot)}(0^{n_s})$ in which all the words $0w_1, \dots, 0w_{p_s(n_s)+1}$ are queried. Thus we obtain a contradiction. So, $\text{gap}_{N_s^{\mathcal{A} \cup C_t}}(0^{n_s}) = \text{val}$. \blacksquare

If in stage s no set satisfying (\star) exists, then (3.iii) is true. Thus, from Lemma 3.7, for $t = 2^{n_s-1}$ and for any set C_t such that $C_t \subseteq \overline{T_0} \cap 0\Sigma^{n_s}$ and $\|C_t\| = t$, it holds that

$gap_{N_s^{A \cup C_t}}(0^{n_s}) = val$. Thus, under the assumption that in stage s no set satisfying (\star) exists, the following holds.

For every set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$,

$$\text{I-Promise}(B, n_s, 2^{n_s-1} - 1) \implies gap_{N_s^{A \cup B}}(0^{n_s}) = val, \text{ and} \quad (3.v)$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1}) \implies gap_{N_s^{A \cup B}}(0^{n_s}) = 0. \quad (3.vi)$$

Note that (3.v) is a logically stronger statement than (3.iii). We now argue that the above chain of arguments can be pushed further to show that, if in stage s no set satisfying (\star) exists, then $gap_{N_s^A}(0^{n_s}) = val$.

Lemma 3.8 *For each $s \geq 1$, if in stage s no set satisfying (\star) exists, then $gap_{N_s^A}(0^{n_s}) = val$.*

Proof Assume that in stage s no set satisfying (\star) exists. For any $t' \in \mathbb{Z}$, let $P(t')$ be the following statement:

$$P(t') \equiv (\forall B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}) [\text{I-Promise}(B, n_s, t') \implies gap_{N_s^{A \cup B}}(0^{n_s}) = val].$$

We prove using induction that, for every $t' \in \{-1, \dots, 2^{n_s-1}\}$, $P(t')$ holds. The base cases (when $t' = 2^{n_s-1}$ or $t' = 2^{n_s-1} - 1$) have already been shown to be true. By induction hypothesis, for $t' = k$, where $k \geq 0$,

$$P(k) \equiv (\forall B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}) [\text{I-Promise}(B, n_s, k) \implies gap_{N_s^{A \cup B}}(0^{n_s}) = val]$$

is true. Now applying Lemma 3.7 for $t = k$, we have that

$$(\forall C \subseteq \overline{T_0} \cap 0\Sigma^{n_s}) [||C|| = k \implies gap_{N_s^{A \cup C}}(0^{n_s}) = val].$$

Thus, certainly $(\forall C \subseteq \overline{T_0} \cap \overline{T_1} \cap 0\Sigma^{n_s}) [||C|| = k \implies gap_{N_s^{A \cup C}}(0^{n_s}) = val]$. Combining this with the induction hypothesis, we have that

$$P(k-1) \equiv (\forall B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}) [\text{I-Promise}(B, n_s, k-1) \implies gap_{N_s^{A \cup B}}(0^{n_s}) = val]$$

is also true. Thus, by induction, for all $t' \in \{-1, \dots, 2^{n_s-1}\}$, $P(t')$ is true. Since $P(-1)$ and $\text{I-Promise}(\emptyset, n_s, -1)$ are true, it follows that $gap_{N_s^A}(0^{n_s}) = val$. \blacksquare

We now show that if in stage s no set satisfying (\star) exists, then, using (3.iv), a contradiction can be achieved from Lemma 3.8; this will imply that in stage s a set B satisfying (\star) always exists. We first prove a lemma analogous to Lemma 3.6.

Lemma 3.9 *For each $s \geq 1$ and every $t \in \{0, \dots, 2^{n_s-1}\}$, if for every set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$, it holds that*

$$\text{II-Promise}(B, n_s, t) \implies gap_{N_s^{A \cup B}}(0^{n_s}) = 0,$$

then the following hold:

1. *For every nonempty sequence of words $1w_1, 1w_2, \dots, 1w_k$ such that $\{1w_1, \dots, 1w_k\} \subseteq \overline{T_1} \cap 1\Sigma^{n_s}$ and for every set R that satisfies $R \subseteq \overline{T_1} \cap 1\Sigma^{n_s}$, $\text{II-Promise}(R, n_s, t)$, and $\{1w_1, \dots, 1w_k\} \cap R = \emptyset$, it holds that $J_{1w_1, \dots, 1w_k}^R = 0$.*
2. *For every set D_t such that $D_t \subseteq \overline{T_1} \cap 1\Sigma^{n_s}$, $||D_t|| = t$, and $\{1w_1, \dots, 1w_k\} \cap D_t = \emptyset$, it holds that $J_{1w_1, \dots, 1w_k}^{D_t} = gap_{N_s^{A \cup D_t}}(0^{n_s})$.*

Proof We prove (1) by induction on k . By hypothesis and by using the definition of J , for every set R such that $R \subseteq \overline{T_1} \cap 1\Sigma^{n_s}$ and $\text{II-Promise}(R, n_s, t)$ is true, the following holds: $J^R = Q^R = 0$. By Lemma 3.5,

$$J_{1w_1, \dots, 1w_{k+1}}^R = J_{1w_1, \dots, 1w_k}^R - J_{1w_1, \dots, 1w_k}^{R \cup \{1w_{k+1}\}}.$$

For $k = 1$, $J_{1w_1}^R = J^R - J^{R \cup \{1w_1\}} = Q^R - Q^{R \cup \{1w_1\}} = 0 - 0 = 0$.

For $k > 1$, $J_{1w_1, \dots, 1w_k}^R = J_{1w_1, \dots, 1w_{k-1}}^R - J_{1w_1, \dots, 1w_{k-1}}^{R \cup \{1w_k\}}$, where by induction hypothesis both terms are 0.

We prove (2) also by induction on k .

For $k = 1$,

$$J_{1w_1}^{D_t} = Q_{1w_1}^{D_t} - Q_{1w_1}^{D_t \cup \{1w_1\}} = Q^{D_t} - Q_{(1w_1)}^{D_t} - Q^{D_t \cup \{1w_1\}} + Q_{(1w_1)}^{D_t \cup \{1w_1\}} = Q^{D_t} - Q^{D_t \cup \{1w_1\}},$$

where the third equality follows from the fact that, by Definition 3.2, $Q_{(1w_1)}^{D_t} = Q_{(1w_1)}^{D_t \cup \{1w_1\}}$. Thus,

$$J_{1w_1}^{D_t} = \text{gap}_{N_s^{A \cup D_t}}(0^{n_s}) - 0 = \text{gap}_{N_s^{A \cup D_t}}(0^{n_s}),$$

using the definition of D_t and using the fact that $\text{II-Promise}(D_t \cup \{1w_1\}, n_s, t)$ is true. For $k > 1$, by Lemma 3.5,

$$J_{1w_1, \dots, 1w_k}^{D_t} = J_{1w_1, \dots, 1w_{k-1}}^{D_t} - J_{1w_1, \dots, 1w_{k-1}}^{D_t \cup \{1w_k\}}.$$

By (1),

$$J_{1w_1, \dots, 1w_{k-1}}^{D_t \cup \{1w_k\}} = 0.$$

Also, by the induction hypothesis,

$$J_{1w_1, \dots, 1w_{k-1}}^{D_t} = \text{gap}_{N_s^{A \cup D_t}}(0^{n_s}).$$

Thus,

$$J_{1w_1, \dots, 1w_k}^{D_t} = \text{gap}_{N_s^{A \cup D_t}}(0^{n_s}).$$

■

The next two lemmas, Lemma 3.10 and Lemma 3.11, are analogous to Lemma 3.7 and Lemma 3.8, respectively, with minor differences in their statement. So, we omit their proof in the paper.

Lemma 3.10 *For each $s \geq 1$ and every $t \in \{0, \dots, 2^{n_s-1}\}$, if for every set B such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$, it holds that*

$$\text{II-Promise}(B, n_s, t) \implies \text{gap}_{N_s^{A \cup B}}(0^{n_s}) = 0,$$

then for every set D_t such that $D_t \subseteq \overline{T_1} \cap 1\Sigma^{n_s}$ and $\|D_t\| = t$, it holds that $\text{gap}_{N_s^{A \cup D_t}}(0^{n_s}) = 0$.

Proof Omitted, since the proof is similar to that of Lemma 3.7. ■

If in stage s no set satisfying (\star) exists, then (3.iv) is true. Thus, from Lemma 3.10, for $t = 2^{n_s-1}$ and for any set D_t such that $D_t \subseteq \overline{T_1} \cap 1\Sigma^{n_s}$ and $\|D_t\| = t$, it holds that $gap_{N_s^{\mathcal{A} \cup D_t}}(0^{n_s}) = 0$. Thus, under the assumption that in stage s no set satisfying (\star) exists, the following holds.

$$\text{For every set } B \text{ such that } B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}, \\ \text{I-Promise}(B, n_s, 2^{n_s-1}) \implies gap_{N_s^{\mathcal{A} \cup B}}(0^{n_s}) = val, \text{ and} \quad (3.vii)$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1} - 1) \implies gap_{N_s^{\mathcal{A} \cup B}}(0^{n_s}) = 0. \quad (3.viii)$$

The above chain of arguments can be pushed further to show that, if in stage s no set satisfying (\star) exists, then $gap_{N_s^{\mathcal{A}}}(0^{n_s}) = 0$.

Lemma 3.11 *For each $s \geq 1$, if in stage s no set satisfying (\star) exists, then $gap_{N_s^{\mathcal{A}}}(0^{n_s}) = 0$.*

Proof Omitted, since the proof is similar to that of Lemma 3.8. ■

Since $val \neq 0$, Lemma 3.8 and Lemma 3.11 imply a contradiction. Thus, for each $s \geq 1$, a set B satisfying (\star) can always be found in stage s . This completes the proofs of Claim 1 and Theorem 3.1. ■

Corollary 3.12 *There exists an oracle $\mathcal{A} \subseteq \Sigma^*$ such that, for each $\mathcal{C} \in \{\text{ZPP}, \text{RP}, \text{BPP}, \text{NP}, \text{BQP}, \text{C=P} \cap \text{coC=P}, \text{AWPP}, \text{APP}\}$ and for each $\mathcal{D} \in \{\text{UP}, \text{FewP}, \text{SPP}, \text{EQP}, \text{LWPP}, \text{WPP}\}$, $\mathcal{C}^{\mathcal{A}} \not\subseteq \mathcal{D}^{\mathcal{A}}$.*

Proof Let \mathcal{A} be the oracle constructed in Theorem 3.1. Then, $\mathcal{C}^{\mathcal{A}} \not\subseteq \mathcal{D}^{\mathcal{A}}$ follows from the oracle separation of ZPP from WPP in Theorem 3.1, and from $\text{ZPP}^{\mathcal{A}} \subseteq \mathcal{C}^{\mathcal{A}}$ and $\mathcal{D}^{\mathcal{A}} \subseteq \text{WPP}^{\mathcal{A}}$ (see Proposition 2.12). ■

Corollary 3.12 shows that nonrelativizable proof techniques will be required to prove that error-free quantum polynomial-time (EQP) algorithms exist for all languages in ZPP. Corollary 3.12 also shows that, using relativizable techniques, we cannot lower the best known classical upper bound for BQP from AWPP to even WPP, the largest known natural gap-definable subclass of AWPP. Note that WPP can be considered as a weak subclass of AWPP in the following sense: There is an oracle relative to which $\text{UP} \cap \text{coUP}$ is not low for WPP [ST04b,ST04a], whereas SPP, a superclass of UP, is low for AWPP in every relativized world. Thus, Corollary 3.12 shows that BQP might not be as weak a subclass of AWPP as WPP in some relativized world.

Theorem 3.1 provides an intuition on the relativized complexity of EQP and BQP. But, what about the relativized complexity of NQP? It is equally worth investigating the relativized complexity of C=P for this question, since NQP equals coC=P in every relativized world [FGHP98,YY99]. Theorem 3.13 makes a step in this direction and shows that there is a relativized world where coRP is not contained in NQP.

Tarui [Tar91] used a lower bound technique in decision trees for a certain AC^0 function to show that BPP is not contained in $\text{P}^{\text{C=P}}$ in some relativized world. Green [Gre93] used circuit lower bound techniques to obtain the same result. In contrast with BPP, RP is contained in $\text{P}^{\text{C=P}}$ in every relativized world. In Theorem 3.13, we use the ‘‘gap analog’’

of the proof technique by Torán [Tor91] to construct an oracle relative to which RP is not contained in C=P. This result is optimal in the sense that the largest known natural subclass of RP, namely ZPP, is contained in C=P in every relativized world. This oracle separation of RP from C=P is also a strengthening of the oracle separation of NP from C=P by Torán [Tor91].

Theorem 3.13 $(\exists \mathcal{A}) [\text{RP}^{\mathcal{A}} \not\subseteq \text{C=P}^{\mathcal{A}}]$.

Proof The oracle construction is similar to that in Theorem 3.1. For every set B , let $L_B =_{df} \{0^n \mid (\exists w \in \Sigma^n)[w \in B]\}$. Define predicates “I-Promise” and “II-Promise” as follows.

$$\begin{aligned} \text{I-Promise}(B, n, k) &\equiv \quad ||B \cap \Sigma^n|| > k, \text{ and} \\ \text{II-Promise}(B, n, k) &\equiv \quad ||B \cap \Sigma^n|| = 0. \end{aligned}$$

We say that a set B satisfies the promise at length n with threshold k if $[\text{I-Promise}(B, n, k) \vee \text{II-Promise}(B, n, k)]$ is true. We show that there is an oracle \mathcal{A} such that \mathcal{A} satisfies the promise at each length $n \geq 1$ with threshold 2^{n-1} and $L_{\mathcal{A}} \notin \text{C=P}^{\mathcal{A}}$.

Let $(N_s, p_s)_{s \geq 1}$ be an enumeration of all pairs such that the first component is a nondeterministic polynomial-time oracle Turing machine, the second component is a polynomial, and the running time of N_s is bounded by the polynomial p_s regardless of the oracle. We now describe the stages in the construction of the oracle.

Stage 0: Let $\mathcal{A}_0 := \emptyset$ and let $n_0 := 0$.

Stage s , where $s > 0$: Let n_s be the smallest integer such that the following hold: $n_s > n_{s-1}$, $2^{n_s} > 2p_s(n_s) + 2$, and n_s is large enough so that the constructions in previous stages are not affected. In stage s , we diagonalize against the pair (N_s, p_s) .

($\star\star$) Choose a set $B \subseteq \Sigma^{n_s}$ such that one of the following conditions are satisfied:

$$\text{I-Promise}(B, n_s, 2^{n_s-1}) \quad \text{and} \quad \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) \neq 0, \text{ or} \quad (3.\text{ix})$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1}) \quad \text{and} \quad \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) = 0. \quad (3.\text{x})$$

Let $\mathcal{A}_s := \mathcal{A}_{s-1} \cup B$ and move to stage $s + 1$.

End of Stage

Let $\mathcal{A} := \lim_{s \rightarrow \infty} \mathcal{A}_s$. Obviously, the construction guarantees that $L_{\mathcal{A}} \notin \text{C=P}^{\mathcal{A}}$. The feasibility of the construction follows from the following claim.

Claim 2 *For each $s \geq 1$, there exists an oracle extension B satisfying ($\star\star$).*

Proof of Claim 2. Our combinatorial tool is the same as the one we used in proving Claim 1. The Q and J notations used here are as in Definitions 3.2 and 3.4, respectively. (The only minor changes are that $E \subseteq \Sigma^{n_s+1}$ is replaced by $E \subseteq \Sigma^{n_s}$ and \mathcal{A} is replaced by \mathcal{A}_{s-1} in the new definition.) Suppose that in stage s no set satisfying ($\star\star$) exists. Then, for every set $B \subseteq \Sigma^{n_s}$, it follows that

$$\text{I-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) = 0, \text{ and} \quad (3.\text{xi})$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) \neq 0. \quad (3.\text{xii})$$

Lemmas 3.14, 3.15, and 3.16 can be proved in a way similar to Lemmas 3.6, 3.7, and 3.8, respectively, are proved. So, we omit their proofs in the paper.

Lemma 3.14 *For each $s \geq 1$ and every $t \in \{0, \dots, 2^{n_s-1}\}$, if for every set $B \subseteq \Sigma^{n_s}$, it holds that*

$$\text{I-Promise}(B, n_s, t) \implies \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) = 0,$$

then the following hold:

1. *For every nonempty sequence of words w_1, \dots, w_k , where $\{w_1, \dots, w_k\} \subseteq \Sigma^{n_s}$ and for every set R such that $R \subseteq \Sigma^{n_s}$, $\text{I-Promise}(R, n_s, t)$ is true and $\{w_1, \dots, w_k\} \cap R = \emptyset$, it holds that $J_{w_1, \dots, w_k}^R = 0$.*
2. *For every set C_t such that $C_t \subseteq \Sigma^{n_s}$, $\|C_t\| = t$, and $\{w_1, \dots, w_k\} \cap C_t = \emptyset$, it holds that $J_{w_1, \dots, w_k}^{C_t} = \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup C_t}}(0^{n_s})$.*

Lemma 3.15 *For each $s \geq 1$ and every $t \in \{0, \dots, 2^{n_s-1}\}$, if for every set $B \subseteq \Sigma^{n_s}$, it holds that*

$$\text{I-Promise}(B, n_s, t) \implies \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) = 0,$$

then for every set C_t such that $C_t \subseteq \Sigma^{n_s}$ and $\|C_t\| = t$, it holds that $\text{gap}_{N_s^{\mathcal{A}_{s-1} \cup C_t}}(0^{n_s}) = 0$.

If in stage s no set satisfying $(\star\star)$ exists, then (3.xi) is true. Thus, from Lemma 3.15, for $t = 2^{n_s-1}$ and for any set C_t such that $C_t \subseteq \Sigma^{n_s}$ and $\|C_t\| = t$, it holds that $\text{gap}_{N_s^{\mathcal{A}_{s-1} \cup C_t}}(0^{n_s}) = 0$. Thus, under the assumption that in stage s no set satisfying $(\star\star)$ exists, the following holds. For every set $B \subseteq \Sigma^{n_s}$,

$$\text{I-Promise}(B, n_s, 2^{n_s-1} - 1) \implies \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) = 0, \text{ and} \quad (3.xiii)$$

$$\text{II-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N_s^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) \neq 0. \quad (3.xiv)$$

Note that (3.xiii) is a logically stronger statement than (3.xi). As in the proof of Claim 1, the above chain of arguments can be pushed further to show that, if in stage s no set satisfying $(\star\star)$ exists, then $\text{gap}_{N_s^{\mathcal{A}_{s-1}}}(0^{n_s}) = 0$.

Lemma 3.16 *For each $s \geq 1$, if in stage s no set satisfying $(\star\star)$ exists, then $\text{gap}_{N_s^{\mathcal{A}_{s-1}}}(0^{n_s}) = 0$.*

Lemma 3.16 and (3.xii) imply a contradiction. Thus, for each $s \geq 1$, partial oracle \mathcal{A}_{s-1} can always be extended in stage s . This completes the proofs of Claim 2 and Theorem 3.13. \blacksquare

The following corollary follows immediately from Theorem 3.13 and the inclusions stated in Proposition 2.12.

Corollary 3.17 *There exists an oracle $\mathcal{A} \subseteq \Sigma^*$ such that, for each $\mathcal{C} \in \{\text{coRP}, \text{BPP}, \text{coNP}, \text{BQP}, \text{AWPP}, \text{APP}\}$, $\mathcal{C}^{\mathcal{A}} \not\subseteq \text{NQP}^{\mathcal{A}}$.*

Later in Section 4, we will prove that Theorem 3.13 can be strengthened to a relativized immunity separation of RP from C=P (see Theorem 4.9).

4 Immunity Separations of Quantum Classes from Counting Classes

A (simple) separation of a class \mathcal{C} from another class \mathcal{D} shows the existence of sets A in \mathcal{C} that do not belong to \mathcal{D} . However, it does not preclude the possibility that every such set A has a close approximation (in some sense) by some set in \mathcal{D} . For instance, it is possible that \mathcal{C} separates from \mathcal{D} , yet every infinite set in \mathcal{C} has an infinite subset (approximation) in \mathcal{D} . An immunity separation of a class \mathcal{C} from a class \mathcal{D} is stronger than a simple separation, since the immunity separation also shows the existence of sets S in \mathcal{C} that cannot even be approximated (in the sense mentioned above) by any infinite set in \mathcal{D} . The sets S are said to be \mathcal{D} -immune.

A relativized separation of one class from another does not necessarily imply a relativized immunity separation of the classes. We give an example regarding the boolean hierarchy. Cai et al. [CGH⁺88] showed that there is an oracle relative to which the levels of the boolean hierarchy separate completely. Since every set in the boolean hierarchy can be expressed, using one of the normal form representations of the set, as a finite union of sets in DP, the second level of the boolean hierarchy, there is no relativized world where some set in the boolean hierarchy is DP-immune. Another example is as follows: Though P separates from NP in some relativized world [BGS75], it is known that relative to a generic oracle, NP has no infinite P-immune set [BI87].

Immunity (strong) separations have been used to study the relativized complexity of classes in different settings. Ko [Ko90] showed that there is a relativized world, where for every $k \geq 1$, Σ_k^p contains a Σ_{k-1}^p -immune set. Bruschi [Bru92] extended this result of Ko and showed that there exist oracles that witness the following: For every $k \geq 1$, the existence of a Δ_k^p -immune set in Σ_k^p , the existence of a set L in Σ_k^p such that \bar{L} is Σ_k^p -immune, the existence of a Δ_k^p -immune set in a relativized polynomial hierarchy for which $\Sigma_k^p = \Pi_k^p \neq \Delta_k^p$, and for every $k > 1$, the existence of a Σ_{k-1}^p -immune set in a relativized polynomial hierarchy for which $\Sigma_k^p = \Delta_k^p$. Bruschi, Joseph, and Young [BJY90] strongly separated the boolean hierarchy over RP, and Rothe [Rot99] used immunity separations to study the relativized complexity of counting classes. Rothe [Rot99], in particular, showed that there are oracles A_1 , A_2 , and A_3 such that $C=P^{A_1}$ contains a $BPP^{\oplus P^{A_1}}$ -immune set, NP^{A_2} contains a $C=P^{A_2}$ -immune set, and $\oplus P^{A_3}$ contains a $PP^{PH^{A_3}}$ -immune set.

Before we state and prove our results, we formally define the notions of immunity and strong separations.

Definition 4.1 *Let \mathcal{C} be a class of languages. An infinite language L is called \mathcal{C} -immune if $(\forall L' \in \mathcal{C}) [|L'| = \infty \Rightarrow L' \cap \bar{L} \neq \emptyset]$. A class \mathcal{D} is \mathcal{C} -immune (or, is immune to \mathcal{C}) if there is a set L in \mathcal{D} that is \mathcal{C} -immune.*

Definition 4.2 *Given relativizable classes \mathcal{C}_1 and \mathcal{C}_2 , an oracle E strongly separates \mathcal{C}_2^E from \mathcal{C}_1^E if \mathcal{C}_2^E is \mathcal{C}_1^E -immune.*

M. de Graaf and P. Valiant [dGV02] proved that, for any prime p and integer $k \geq 1$, there exists an oracle A' such that $EQP^{A'} \not\subseteq \text{Mod}_{p^k} P^{A'}$. In Theorem 4.4, we strengthen this result

by proving that there is a relativized world where every infinite set in EQP cannot even be approximated by some set in Mod_{p^k}P , i.e., EQP *strongly separates* from Mod_{p^k}P . To prove that the test language we use is in (relativized) EQP, we make use of the observation by Boyer et al. [BBHT98] that quantum database searching can be done in polynomial time with *certainty* if the number of solutions is exactly one fourth of the total search-space. We also need the following theorem by Lucas [Luc78] to determine $\binom{n}{k} \bmod p$ for any prime p and integers $n, k \geq 1$.

Theorem 4.3 [Luc78] *For any prime p and integers $n, k \geq 1$, $\binom{n}{k} \bmod p = \prod_{i=0}^r \binom{\binom{n}{p^i}}{\binom{k}{p^i}} \bmod p$, where for any $m \in \mathbb{N}$, $\binom{m}{p^i}$ is the coefficient of p^i in the base- p expansion of m and r is the largest integer such that, either $\binom{n}{p^r} \neq 0$ or $\binom{k}{p^r} \neq 0$. We take here the standard convention that for any $n, k \in \mathbb{N}$, $\binom{n}{k} = 0$ if $n < k$.*

Theorem 4.4 *For every prime p and integer $k \geq 1$, there exist an oracle \mathcal{A} and an infinite set $L_{\mathcal{A}}$ such that $L_{\mathcal{A}} \in \text{EQP}^{\mathcal{A}}$ and $L_{\mathcal{A}}$ is $\text{Mod}_{p^k}\text{P}^{\mathcal{A}}$ -immune.*

Proof Since for each prime p and integer $k \geq 1$, $\text{Mod}_{p^k}\text{P} = \text{Mod}_p\text{P}$ [BG92] relative to all oracles, we may henceforth assume that $k = 1$. For each $n \in \mathbb{N}$, let $f_n = 2(1 + 3p)p^{n+1}$, $g_n = (p^2 + 5p - 2)p^n/2$, and $\text{ext}(n) = \lceil \log_2 f_n \rceil$. For each $D \subseteq \Sigma^*$ and $n \in \mathbb{N}$, let $D^{\text{ext}(n)}$ be defined as: $D^{\text{ext}(n)} = D \cap \{x \in \Sigma^{\text{ext}(n)} \mid \text{pos}(x) \leq f_n\}$, where $\text{pos}(x)$ is the number of strings of length $|x|$ that are lexicographically less than or equal to x . For each $D \subseteq \Sigma^*$, define L_D as follows:

$$L_D = \{0^n \mid \|D^{\text{ext}(n)}\| = (1/4) * f_n\}.$$

We say that an oracle $D \subseteq \Sigma^*$ is *relevant*, if for each $n \in \mathbb{N}$, $\|D^{\text{ext}(n)}\| \in \{f_n/4, 3f_n/4\}$. Boyer et al. [BBHT98] observed that Grover's [Gro96] algorithm can be used to design a quantum polynomial-time algorithm that decides with certainty whether the number of solutions in a database is exactly one fourth or three fourths of the total search space. Using their result, it is easy to see that for any *relevant* oracle D , $L_D \in \text{EQP}^D$.

We construct a *relevant* oracle \mathcal{A} such that $L_{\mathcal{A}}$ is $\text{Mod}_p\text{P}^{\mathcal{A}}$ -immune. The oracle \mathcal{A} we construct will have the following form: $\mathcal{A} = \bigcup_{n \in \mathbb{N}} (Z_n \cup W_n)$, where for every $n \in \mathbb{N}$, $Z_n = \{x \in \Sigma^{\text{ext}(n)} \mid g_n < \text{pos}(x) \leq g_n + f_n/4\}$ and $W_n \subseteq \{x \in \Sigma^{\text{ext}(n)} \mid g_n + f_n/4 < \text{rank}(x) \leq f_n\}$. Note that since \mathcal{A} is *relevant*, this means that, for any n , the cardinality of W_n is either exactly 0 or exactly $f_n/2$.

The set \mathcal{A} is constructed in stages. Let $\mathcal{A}_0 = \bigcup_{n \in \mathbb{N}} Z_n$. In stage s , $s > 0$, we construct a set B consisting of strings of length $\text{ext}(n_s)$, for sufficiently large n_s , from the set $\{x \in \Sigma^{\text{ext}(n_s)} \mid g_n + f_n/4 < \text{pos}(x) \leq f_n\}$. At the end of each stage, we set $\mathcal{A}_s := \mathcal{A}_{s-1} \cup B$. Let $\mathcal{A} =_{df} \lim_{s \rightarrow \infty} \mathcal{A}_s$.

Let N_1, N_2, \dots be an enumeration of all oracle NPTMs such that (a) for every $i \geq 1$, the running time of N_i is bounded by nondecreasing polynomial p_i independent of the oracle, and (b) there are an infinite number of indices j such that, for each $X \subseteq \Sigma^*$ and each $x \in \Sigma^*$, $\#\text{acc}_{N_j^X}(x) \equiv 0 \pmod{p}$. In order to construct a set $L_{\mathcal{A}}$ that is $\text{Mod}_p\text{P}^{\mathcal{A}}$ -immune, we will use a set T of conditions (requirements) that are to be met during the construction

of the oracle. We will update T during the construction in the following way. At each stage s , we add to T the condition corresponding to machine N_s , and delete from T a condition if during stage s the condition is met. Thus, at the beginning of each stage of the construction, T will contain all the conditions that have yet to be satisfied. For each $i \in \mathbb{N}$, define condition T_i as “ $L(N_i^A) \cap \overline{L_A} \neq \emptyset$.” Note that if condition T_i is satisfied, then $L(N_i^A)$ is not a subset of L_A ; thus $L(N_i^A)$ is definitely not an infinite subset of L_A .

Claim 3 For each $T = \{T_{t_1}, T_{t_2}, \dots, T_{t_k}\} \subseteq \mathbb{N}$, where t_1, t_2, \dots, t_k are indices of oracle NPTMs $N_{t_1}, N_{t_2}, \dots, N_{t_k}$, respectively, and for each $W \subseteq \Sigma^*$, there exists an oracle NPTM \hat{N}_T such that, $\#\text{acc}_{\hat{N}_T^W}(x) \not\equiv 0 \pmod{p}$ if and only if there exists $T_{t_j} \in T$ such that $\#\text{acc}_{N_{t_j}^W}(x) \not\equiv 0 \pmod{p}$. Moreover, the running time of \hat{N}_T^W on every x is bounded by $p^2 \sum_{i=1}^k p_{t_i}(|x|)$.

Proof The claim follows from the fact that, for any prime p , Mod_pP is closed under union [BG92]. ■

Notation: For any set $T = \{T_{t_1}, T_{t_2}, \dots, T_{t_k}\}$ of conditions, let $\text{Index}(T)$ denote the set $\{t_1, t_2, \dots, t_k\}$ of indices of machines.

Stage 0: Set $\mathcal{A}_0 := \bigcup_{n \in \mathbb{N}} Z_n$, $T = \emptyset$.

Stage s , where $s \geq 1$: Add condition T_s to T . Let $T = \{T_{t_1}, T_{t_2}, \dots, T_{t_k}\}$. Choose n_s large enough such that previous stages are not affected and $p^{n_s} > p^2 \sum_{i=1}^k p_{t_i}(n_s)$. We call B allowable if it contains exactly 0 or exactly $f_{n_s}/2$ strings from the set $\{x \in \Sigma^{\text{ext}(n_s)} \mid g_{n_s} + f_{n_s}/4 < \text{pos}(x) \leq f_{n_s}\}$. We will choose an allowable set B such that

$$B = \emptyset \iff \#\text{acc}_{\hat{N}_T^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) \equiv 0 \pmod{p}$$

is satisfied. (We prove in Claim 6 that such an allowable set always exists.) Let B_s be the allowable set chosen. If B_s is nonempty, then there exists some $t_j \in \text{Index}(T)$ such that $\#\text{acc}_{N_{t_j}^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) \not\equiv 0 \pmod{p}$. Let t_j be the smallest such element. Remove T_{t_j} from T .

Set $\mathcal{A}_s = \mathcal{A}_{s-1} \cup B_s$. Go to Stage $s + 1$.

End of Stage.

Note that, as stated earlier, $\mathcal{A} = \lim_{s \rightarrow \infty} \mathcal{A}_s$.

Claim 4 L_A is an infinite set.

Proof Since for each s , $0^{n_s} \in L_A$ if and only if B_s is empty, it suffices to prove that there are an infinite number of stages s such that B_s is empty. Let us assume that the claim is false. Then there exists a stage m such that, for each stage $s > m$, B_s is nonempty. Thus, for each $s > m$, $\#\text{acc}_{\hat{N}_T^{\mathcal{A}_{s-1} \cup B_s}}(0^{n_s}) \not\equiv 0 \pmod{p}$, and by construction, some element is removed from T . Also, note that the enumeration of NPTMs has an infinite number of indices j such that N_j robustly (i.e., for all oracles) and for all strings has a number of accepting paths divisible by p . Let such machines N_j be called *robustly rejecting*. Since a condition T_t can be removed from T only if N_t with some oracle on some string has a number of accepting paths not divisible by p , a condition corresponding to a robustly

rejecting machine N_j can never be removed from T . Recall that in every stage s , we add a condition T_s corresponding to machine N_s in T . By assumption, for every stage $s > m$, one of the conditions corresponding to some machine N_k , where N_k is not robustly rejecting, is removed from T . Thus in every stage $s > m$, if N_s is robustly rejecting, then the number of conditions corresponding to machines that are not robustly rejecting decreases by one and if N_s is not robustly rejecting, then the number of conditions corresponding to machines that are not robustly rejecting remains the same at the end of stage s . Since there are infinitely many robustly rejecting machines in our enumeration of oracle NPTMs, there will exist an integer $q > m$ such that after stage q , $\text{Index}(T)$ will consist of indices of robustly rejecting machines. Thus, at stage $q + 1$, for all allowable B , $\#\text{acc}_{\hat{N}_T^{\mathcal{A}_q \cup B}}(0^{n_{q+1}}) \equiv 0 \pmod{p}$. Thus, B_{q+1} will be empty, contradicting our assumption. \blacksquare

Claim 5 For each i , $N_i^{\mathcal{A}}$ does not accept (in the Mod_pP sense) an infinite subset of $L_{\mathcal{A}}$.

Proof For each i , either condition T_i is satisfied at some stage s or condition T_i is never satisfied. Consider the case that T_i is satisfied at stage s . Then, $L(N_i^{\mathcal{A}}) \cap \overline{L_{\mathcal{A}}} \neq \emptyset$. Thus, $N_i^{\mathcal{A}}$ does not accept a subset of $L_{\mathcal{A}}$. Now consider the case that T_i is never satisfied. We will now prove that $L(N_i^{\mathcal{A}}) \cap L_{\mathcal{A}}$ is finite. Since T_i is added to T at stage i , it suffices to prove that, for each $k \geq i$, if $0^{n_k} \in L_{\mathcal{A}}$ then $0^{n_k} \notin L(N_i^{\mathcal{A}})$. Assume that $0^{n_k} \in L_{\mathcal{A}}$. Then, by construction, $\#\text{acc}_{\hat{N}_T^{\mathcal{A}}}(0^{n_k}) \equiv 0 \pmod{p}$. Thus, by Claim 3, $N_i^{\mathcal{A}}(0^{n_k})$ has a number of accepting paths divisible by p and $0^{n_k} \notin L(N_i^{\mathcal{A}})$. \blacksquare

Claims 4 and 5 together imply that if the construction is feasible, then $L_{\mathcal{A}}$ is $\text{Mod}_p\text{P}^{\mathcal{A}}$ -immune. It remains to show that the construction is feasible. That is, we need to show that at each stage $s \geq 1$, B_s can be chosen.

Claim 6 For each $s \geq 1$, there exists an allowable set B such that the following holds:

$$B = \emptyset \iff \#\text{acc}_{\hat{N}_T^{\mathcal{A}_{s-1} \cup B}}(0^{n_s}) \equiv 0 \pmod{p}.$$

Proof The application of the combinatorial “double counting” technique of “reversing the order of summation” in the construction of oracles was introduced by Beigel [Bei91]. This technique can be described as follows. Let N be an oracle NPTM. Let \mathcal{B} be a finite set of oracles and let $x \in \Sigma^*$. Then the sum of the number of accepting paths in $N^B(x)$ for each $B \in \mathcal{B}$ can be computed in two ways. One way is to sum the total number of accepting paths for each $B \in \mathcal{B}$. The other way is to sum, for each path ρ , the number of oracles $B \in \mathcal{B}$ that make $N^B(x)$ accept along ρ . Formally, in our setting, we will use the following equality to prove the claim:

$$\sum_B \#\text{acc}_{\hat{N}_T^{\mathcal{A}_{i-1} \cup B}}(0^{n_i}) = \sum_{\rho} \|\{B \mid \rho \in \text{ACCEPT}(\hat{N}_T^{\mathcal{A}_{i-1} \cup B}, 0^{n_i})\}\|. \quad (4.i)$$

If Claim 6 is false, then the sum of the number of accepting paths of $\hat{N}_T^{\mathcal{A}_{s-1} \cup B}(0^{n_s})$ over all allowable B , i.e., the left hand side of the equality, is not divisible by p . Let ρ be an arbitrary augmented path in $\hat{N}_T(0^{n_s})$. In what follows, we say that a set B supports an augmented computation path ρ of $\hat{N}_T(0^{n_s})$ if $\rho \in \text{ACCEPT}(\hat{N}_T^{\mathcal{A}_{s-1} \cup B}, 0^{n_s})$. We will show

that the number of allowable sets B supporting path ρ of $\hat{N}_T(0^{n_s})$ is a multiple of p in each case. According to Eq. (4.i), this is a contradiction.

Let $\{q_1, \dots, q_\ell\}$ be the set of distinct queries from $\{x \in \Sigma^{ext(n_s)} \mid g_{n_s} + f_{n_s}/4 < \text{rank}(x) \leq f_{n_s}\}$ asked on path ρ of $\hat{N}_T(0^{n_s})$. Let $a_i = 0$ if and only if query q_i is negatively answered, and $a_i = 1$ if and only if query q_i is positively answered on path ρ . For each $x_1, x_2, \dots, x_m \in \Sigma^*$ such that x_1, x_2, \dots, x_m are pairwise distinct and for each $b_1, b_2, \dots, b_m \in \{0, 1\}$, denote by $S(x_1, b_1, x_2, b_2, \dots, x_m, b_m)$ the set of allowable sets B such that $x_i \in B \iff b_i = 1$. If path ρ queries any string from $\{x \in \Sigma^{ext(n_s)} \mid \text{pos}(x) \leq g_{n_s}\}$ positively or any string from Z_{n_s} negatively, then there is no allowable set B supporting path ρ of $\hat{N}_T(0^{n_s})$. Otherwise, $S(q_1, a_1, q_2, a_2, \dots, q_\ell, a_\ell)$ is the set of allowable sets B supporting path ρ of $\hat{N}_T(0^{n_s})$. Let $q_{\ell+1}, q_{\ell+2}, \dots, q_{p^{n_s}} \in \{x \in \Sigma^{ext(n_s)} \mid g_{n_s} + f_{n_s}/4 < \text{rank}(x) \leq f_{n_s}\}$ be arbitrary strings such that $q_1, \dots, q_{p^{n_s}}$ are pairwise distinct. Then

$$\|S(q_1, a_1, q_2, a_2, \dots, q_\ell, a_\ell)\| = \sum_{a_{\ell+1}=0}^1 \sum_{a_{\ell+2}=0}^1 \cdots \sum_{a_{p^{n_s}}=0}^1 \|S(q_1, a_1, q_2, a_2, \dots, q_{p^{n_s}}, a_{p^{n_s}})\|.$$

To prove that $\|S(q_1, a_1, q_2, a_2, \dots, q_\ell, a_\ell)\| \equiv 0 \pmod{p}$, we show that for each $a_{\ell+1}, a_{\ell+2}, \dots, a_{p^{n_s}}$, $\|S(q_1, a_1, q_2, a_2, \dots, q_{p^{n_s}}, a_{p^{n_s}})\| \equiv 0 \pmod{p}$. Consider arbitrary $a_{\ell+1}, a_{\ell+2}, \dots, a_{p^{n_s}}$. Let C be an arbitrary allowable set from $S(q_1, a_1, q_2, a_2, \dots, q_{p^{n_s}}, a_{p^{n_s}})$.

Case 1: $a_1 = 0, \dots, a_{p^{n_s}} = 0$.

Then $q_1, \dots, q_{p^{n_s}}$ cannot be in C . There are exactly $(3/4) * f_{n_s} - g_{n_s} - p^{n_s} = 3p^{n_s+2} + (p-1)p^{n_s+1}$ strings remaining that potentially may be included in C . To be allowable, C must either be empty or must contain exactly $f_{n_s}/2 = 3p^{n_s+2} + p^{n_s+1}$ strings. Therefore, by Lucas's theorem, the number of choices modulo p is, $\left[1 + \binom{3p^{n_s+2} + (p-1)p^{n_s+1}}{3p^{n_s+2} + p^{n_s+1}}\right] \pmod{p} = 0$.

Case 2: Exactly j of the a_i 's are equal to 1 ($1 \leq j \leq p^{n_s}$).

In this case, C may not be the empty set. The membership in C of p^{n_s} strings, namely $q_1, \dots, q_{p^{n_s}}$, is fixed. From the remaining $(3/4) * f_{n_s} - g_{n_s} - p^{n_s}$ possible strings that may be included in some allowable set, we may take an arbitrary subset of strings such that we have altogether exactly $f_{n_s}/2$ strings in C . (Note that exactly j strings are already fixed to be in C .) Hence we have $\binom{3p^{n_s+2} + (p-1)p^{n_s+1}}{f_{n_s}/2 - j} = \binom{3p^{n_s+2} + (p-1)p^{n_s+1}}{(p-2)p^{n_s+1} + j}$ different choices for C . (Here, we use $\binom{\alpha}{\beta} = \binom{\alpha}{\alpha-\beta}$.) Therefore by Lucas's theorem, the number of choices modulo p is $\binom{3}{0} \cdot \binom{p-1}{p-2} \cdot \binom{0}{j} \pmod{p} = 0$, since $j \geq 1$.

This completes the proofs of Claim 6 and Theorem 4.4. ■

Corollary 4.5 *There exists an oracle \mathcal{A} such that, for each $\mathcal{C} \in \{\text{EQP}, \text{BQP}, \text{LWPP}, \text{WPP}, \text{AWPP}, \text{APP}, \text{C=P}, \text{PP}\}$ and for each $\mathcal{D} \in \{\text{UP}, \text{FewP}, \text{SPP}\}$, $\mathcal{C}^{\mathcal{A}}$ is immune to $\mathcal{D}^{\mathcal{A}}$.*

Proof This follows from Theorem 4.4 and the inclusions stated in Proposition 2.12. ■

Theorem 3.13 separates RP from C=P and as a corollary we get a separation of coRP from NQP. In Theorem 4.9, we prove that, relative to an oracle, RP strongly separates from C=P, which in turn implies that coRP strongly separates from NQP. In the proof of Theorem 4.9, we will use a sufficient condition, stated in Theorem 4.8, due to Bovet, Crescenzi, and Silvestri [BCS92], for lifting simple separations between complexity classes to immunity separations.

Definition 4.6 [BCS92] *A function $\sigma : \Sigma^* \rightarrow \Sigma^*$ is polynomially bit-computable if there exist two polynomial-time transducers $R : \Sigma^* \times (\mathbb{N} - \{0\}) \rightarrow \Sigma$ and $\ell : \Sigma^* \rightarrow (\mathbb{N} - \{0\})$ such that, for any $x \in \Sigma^*$,*

$$\sigma(x) = R(x, 1)R(x, 2) \dots R(x, \ell(x)).$$

Definition 4.7 [BCS92] *Let (A, B) be a pair of languages. The C-class $\mathcal{C}(A, B)$ is the set of all languages L for which there exists a polynomially bit computable function σ such that*

$$\sigma(L) \subseteq A \text{ and } \sigma(\bar{L}) \subseteq B.$$

For any class \mathcal{D} , \mathcal{D} is C-class representable if there exist $A, B \subseteq \Sigma^*$ such that $\mathcal{D} = \mathcal{C}(A, B)$.

Theorem 4.8 [BCS92] *There exists an oracle E which strongly separates $\mathcal{C}^E(A_2, B_2)$ from $\mathcal{C}^E(A_1, B_1)$ if the following conditions hold:*

1. *There exists an oracle H such that $\mathcal{C}^H(A_2, B_2) \not\subseteq \mathcal{C}^H(A_1, B_1)$,*
2. *For all K , $\mathcal{C}^K(A_1, B_1)$ admits a $\leq_m^{p,K}$ -complete language.*
3. *For all K , $\mathcal{C}^K(A_1, B_1)$ is closed with respect to the union of languages.*

Using Theorem 4.8, Theorem 3.13 can be strengthened as follows.

Theorem 4.9 *There exists an oracle A such that RP^A contains a $\text{C}=\text{P}^A$ -immune set.*

Proof Both the classes RP and C=P are C-class representable because $\text{RP} = \mathcal{C}(A_1, B_1)$ where $A_1 = \{z \mid z \in \Sigma^* \wedge |z|_1 > \frac{1}{2}|z|\}$ and $B_1 = \{0\}^*$, and $\text{C}=\text{P} = \mathcal{C}(A_2, B_2)$ where $A_2 = \{z \mid z \in \Sigma^* \wedge |z|_0 = |z|_1\}$ and $B_2 = \{z \mid z \in \Sigma^* \wedge |z|_0 \neq |z|_1\}$. From Theorem 3.13, there exists an oracle H such that $\text{RP}^H \not\subseteq \text{C}=\text{P}^H$. Also, in every relativized world, C=P admits a many-one complete language and is closed with respect to the union of languages. Thus, the relativized strong separation follows from Theorem 4.8. \blacksquare

Tarui [Tar91] and Green [Gre93] independently showed that BPP separates from $\text{P}^{\text{C}=\text{P}}$ in some relativized world. In Theorem 4.12 we extend the oracle separation of BPP from $\text{P}^{\text{C}=\text{P}}$ to a strong separation result. From this it follows that, relative to an oracle, BQP strongly separates even from $\text{P}^{\text{C}=\text{P}}$. To prove this result, we use Theorem 4.11, which states that, if a class \mathcal{D} is C-class representable, then the Turing closure of \mathcal{D} is also C-class representable.

Theorem 4.10 [Tar91, Gre93] *There exists an oracle A such that $\text{BPP}^A \not\subseteq \text{P}^{\text{C}=\text{P}^A}$.*

Theorem 4.11 [BCS95] *For any pair of languages (A, B) , there exists a pair of languages (A', B') such that, for all sets $X \subseteq \Sigma^*$*

$$\text{P}^{\text{C}=\text{P}}(A, B)^X = \mathcal{C}^X(A', B')$$

Theorem 4.12 *There exists an oracle A such that for every complexity class $\mathcal{C} \in \{\text{BPP}, \text{BQP}, \Sigma_2^p \cap \Pi_2^p, \text{AWPP}, \text{APP}, \text{PP}\}$, \mathcal{C}^A contains a $\text{P}^{\text{C}=\text{P}^A}$ -immune set.*

Proof We show the existence of an oracle A such that BPP^A is immune to $\text{P}^{\text{C}=\text{P}^A}$. The statement holds for the other classes because they contain BPP in every relativized world. BPP is \mathcal{C} -class representable because $\text{BPP} = \mathcal{C}(A_1, B_1)$, where $A_1 = \{z \mid z \in \Sigma^* \wedge |z|_1 \geq \frac{3}{4}|z|\}$ and $B_1 = \{z \mid z \in \Sigma^* \wedge |z|_1 \leq \frac{1}{4}|z|\}$. Since $\text{C}=\text{P}$ is \mathcal{C} -class representable, it follows from Theorem 4.11 that $\text{P}^{\text{C}=\text{P}}$ is also \mathcal{C} -class representable. It is known that in every relativized world, $\text{P}^{\text{C}=\text{P}}$ admits a complete language and is closed under union. Thus from Theorem 4.10 and Theorem 4.8, it follows that BPP contains a $\text{P}^{\text{C}=\text{P}}$ -immune set in some relativized world. \blacksquare

Since it is not clear whether EQP is \mathcal{C} -class representable, Theorem 4.8 cannot be applied to strengthen the oracle separation of EQP from Mod_{p^k}P [dGV02] to a relativized immunity separation. However, Theorem 4.4 shows that EQP separates from Mod_{p^k}P with immunity in a certain relativized world. Theorem 4.8 is also not useful for strengthening the oracle separation of Theorem 3.1, since there are oracles relative to which WPP has no complete sets [ST04b,ST04a].

5 Closure and Collapse Results

In this section, we further study properties of counting classes and use them to prove consequences of the following hypotheses about quantum classes: $\text{NQP} \subseteq \text{BQP}$ and $\text{NQP} \subseteq \text{EQP}$. Note that these hypotheses are the quantum analogs of the “ $\text{NP} \subseteq \text{BPP}$ ” and “ $\text{NP} \subseteq \text{P}$ ” hypotheses. We show that these hypotheses involving quantum classes have interesting consequences for the polynomial hierarchy.

Zachos [Zac88] proved that $\text{NP} \not\subseteq \text{BPP}$ unless the polynomial hierarchy is contained in BPP, and thus it is unlikely that $\text{NP} \subseteq \text{BPP}$. In this section, we prove in Corollary 5.5 a similar consequence for $\text{NQP} \subseteq \text{BQP}$: $\text{NQP} \subseteq \text{BQP} \implies \text{PH} \subseteq \text{AWPP}$. For the proof of that implication, we show a new reduction closure property of AWPP and make use of an observation—PH is contained in $\text{UP}^{\text{C}=\text{P}}$ in every relativized world—that follows from Toda’s theorem [Tod91]. This observation is implicit in a paper by Vyalıy [Vya03], who showed that if QMA, the quantum analog of the Merlin-Arthur class MA, equals PP, then the polynomial hierarchy is contained in PP. For completeness, we sketch a proof of this observation by Vyalıy [Vya03].

Theorem 5.1 [Vya03] $\text{PH} \subseteq \text{UP}^{\text{C}=\text{P}}$.

Proof Let L be an arbitrary language in the polynomial hierarchy. From Toda’s [Tod91] theorem, $\text{PH} \subseteq \text{P}^{\#\text{P}[1]}$, there exists an oracle DPTM M and a $\#\text{P}$ function f such that $L = L(M^f)$ and for all $x \in \Sigma^*$, $M(x)$ asks only a single query to oracle f . For every $x \in \Sigma^*$, let q_x be the query asked by $M(x)$ to f . Since M is a polynomial-time machine and f is a $\#\text{P}$ oracle, we can assume that there is a polynomial $p(\cdot)$ such that, for all $x \in \Sigma^*$, $0 \leq f(q_x) \leq 2^{p(|x|)}$. Define $A = \{\langle y, k \rangle \mid f(y) = k\}$. It is easy to see that $A \in \text{C}=\text{P}$. (The

function $g(\langle y, k \rangle) = f(y) - k$ is in GapP since $f \in \#P \subseteq \text{GapP}$, and we know that GapP is closed under subtraction. Apply the definition of $C=P$.) Consider an oracle NPTM N such that $N^A(x)$ is defined as follows.

1. Imitate the behavior of $M(x)$ until the query q_x to the $\#P$ -oracle f is generated.
2. Guess an integer j between 0 and $2^{p(|x|)}$ and query to A whether $\langle q_x, j \rangle \in A$.
3. If the oracle answer is “yes,” then go on with the simulation of $M(x)$ with j taken as oracle answer. Accept x if and only if $M(x)$ accepts.
4. Otherwise, i.e., if the oracle answer is “no,” then reject.

It is easy to verify that $L(N^A)$ is in UP^A and $L(N^A) = L$. It follows that $\text{PH} \subseteq \text{UP}^{C=P}$. ■
 Li [Li93] studied closure properties of AWPP and APP, and showed that these classes are closed under intersection, union, complementation, and join. Li [Li93] also showed that APP is closed under polynomial-time Turing reductions and used this result to prove that APP is low for PP. In Theorem 5.4 we show that both AWPP and APP are closed under stronger \leq_T^{UP} (*unambiguous nondeterministic polynomial-time Turing*) reductions. From this closure property of AWPP and Theorem 5.1, we conclude that if $\text{NQP} \subseteq \text{BQP}$ then $\text{PH} \subseteq \text{AWPP}$.

Definition 5.2 We say that $A \leq_T^{\text{UP}} B$ if there exists an oracle NPTM N such that the following hold:

1. $L(N^B) = A$, and
2. $(\forall x \in \Sigma^*)[\#\text{acc}_{N^B}(x) \leq 1]$.

Definition 5.3 For any \mathcal{C} , $\text{UP}^{\mathcal{C}} = \{L \mid (\exists A \in \mathcal{C})[L \leq_T^{\text{UP}} A]\}$.

Theorem 5.4 (a) $\text{UP}^{\text{AWPP}} \subseteq \text{AWPP}$. (b) $\text{UP}^{\text{APP}} \subseteq \text{APP}$.

Proof We will prove (a) only, since the proof for (b) is similar. The proof of (a) is also similar to the proof of the result that APP is closed under polynomial-time Turing reductions by Li [Li93]. Let $L \in \text{UP}^{\text{AWPP}}$ via an oracle NPTM N and an oracle $B \in \text{AWPP}$. Let $r(\cdot)$ be a polynomial (which we will fix later) and let the corresponding GapP function g_B and the polynomial $p(\cdot)$ define B as in Definition 2.3. Without loss of generality, we assume that there exist polynomials $t(\cdot)$, $q(\cdot)$, and $s(\cdot)$ such that for each $A \subseteq \Sigma^*$, $x \in \Sigma^*$, and augmented path $\rho = \langle \rho', \sigma \rangle$ of $N^A(x)$, $|\rho| \leq t(|x|)$, $|\sigma| = q(|x|)$, and the length of any query along ρ is at most $s(|x|)$. We define an NPTM N' as follows. On input $x \in \Sigma^*$, $N'(x)$ guesses an augmented computation path $\rho_N = \langle \rho'_N, \sigma \rangle$ of $N(x)$ and simulates $N(x)$ on ρ'_N with query answers from σ . If $N(x)$ on ρ'_N with answers $\sigma_1, \dots, \sigma_{q(|x|)}$ to the queries rejects, then $N'(x)$ generates a gap of 0 by branching into two paths, rejecting on the one and accepting on the other, and if $N(x)$ on ρ'_N with answers $\sigma_1, \dots, \sigma_{q(|x|)}$ to the queries accepts, then $N'(x)$ generates a gap of $h(\langle x, \rho_N \rangle)$, where h is defined for each $x, \rho_N \in \Sigma^*$ as follows. (Note that in the definition of h below, $z_1, \dots, z_{q(|x|)}$ are the queries that $N(x)$

asks of its oracle along augmented path ρ_N .) Let $h(\langle x, \rho_N \rangle) =_{df} \prod_{i=1}^{q(|x|)} h_i$, where, for each $1 \leq i \leq q(|x|)$, h_i is defined as follows:

$$h_i =_{df} \begin{cases} g_B(z_i) & \text{if } \sigma_i = 1, \\ 2^{p(|z_i|)} - g_B(z_i) & \text{if } \sigma_i = 0. \end{cases}$$

Clearly, $h \in \text{GapP}$. Let $p'(\cdot), r'(\cdot)$ be polynomials such that, for all $n \in \mathbb{N}$, $p'(n) = p(s(n))$ and $r'(n) = r(s(n))$. It is easy to see that, for each $1 \leq i \leq q(|x|)$, $\sigma_i = \chi_B(z_i) \implies (1 - 2^{-r'(|x|)})2^{p'(|x|)} \leq h_i \leq 2^{p'(|x|)}$, and $\sigma_i \neq \chi_B(z_i) \implies 0 \leq h_i \leq 2^{-r'(|x|)}2^{p'(|x|)}$.

If $x \in L$, then there is a unique augmented accepting path $\rho_{acc} = \langle \rho'_{acc}, \sigma_{acc} \rangle$ with query answers σ_{acc} consistent with B in $N(x)$. Clearly, $(1 - 2^{-r'(|x|)})2^{p'(|x|)q(|x|)} \leq h(\langle x, \rho_{acc} \rangle) \leq 2^{p'(|x|)q(|x|)}$. However, if $x \notin L$, then for each augmented path ρ of $N(x)$, $0 \leq h(\langle x, \rho \rangle) \leq 2^{-r'(|x|)}2^{p'(|x|)q(|x|)}$. Thus,

$$x \in L \implies (1 - 2^{-r'(|x|)})2^{p'(|x|)q(|x|)} \leq \frac{gap_{N'}(x)}{2^{p'(|x|)q(|x|)}} \leq 1 + (2^{t(|x|)} - 1)2^{-r'(|x|)}, \quad (5.i)$$

$$x \notin L \implies 0 \leq \frac{gap_{N'}(x)}{2^{p'(|x|)q(|x|)}} \leq (2^{t(|x|)} - 1)2^{-r'(|x|)}. \quad (5.ii)$$

Since for all $x \in \Sigma^*$, $t(|x|) \geq q(|x|)$, we get $(1 - 2^{-r'(|x|)})2^{p'(|x|)q(|x|)} \geq 1 - 2^{-r'(|x|)q(|x|)} \geq 1 - 2^{-(r'(|x|)-q(|x|))} \geq 1 - 2^{-(r'(|x|)-t(|x|))}$.

Thus, (5.i) and (5.ii) can be expressed as

$$x \in L \implies 1 - 2^{-(r'(|x|)-t(|x|))} \leq \frac{gap_{N'}(x)}{2^{p'(|x|)q(|x|)}} \leq 1 + 2^{-(r'(|x|)-t(|x|))}, \quad (5.iii)$$

$$x \notin L \implies 0 \leq \frac{gap_{N'}(x)}{2^{p'(|x|)q(|x|)}} \leq 2^{-(r'(|x|)-t(|x|))}. \quad (5.iv)$$

We choose the polynomial $r(\cdot)$ large enough so that, for all $x \in \Sigma^*$, $r'(|x|) - t(|x|) = r''(|x|) > 1$ and r'' is nondecreasing. Thus,

$$x \in L \implies 1 - 2^{-r''(|x|)} \leq \frac{gap_{N'}(x)}{2^{p'(|x|)q(|x|)}} \leq 1 + 2^{-r''(|x|)}, \quad (5.v)$$

$$x \notin L \implies 0 \leq \frac{gap_{N'}(x)}{2^{p'(|x|)q(|x|)}} \leq 2^{-r''(|x|)}. \quad (5.vi)$$

Consider a GapP function G such that, for all $x \in \Sigma^*$, $G(x) = (2^{r''(|x|)}gap_{N'}(x) + 2^{p'(|x|)q(|x|)}(2^{r''(|x|)-1} - 1))$. For all $x \in \Sigma^*$, let $r'''(|x|) = r''(|x|) - 1$ and $p''(|x|) = p'(|x|)q(|x|) + 2r''(|x|) - 1$. Then,

$$x \in L \implies 1 - 2^{-r'''(|x|)} \leq \frac{G(x)}{2^{p''(|x|)}} \leq (1 + 2^{-r'''(|x|)})(1 - 2^{-r'''(|x|)}) \leq 1,$$

$$x \notin L \implies 0 \leq \frac{G(x)}{2^{p''(|x|)}} \leq 2^{-r'''(|x|)}.$$

By Theorem 2.5(1), $L \in \text{AWPP}$. ■

Corollary 5.5 *If $\text{NQP} \subseteq \text{BQP}$, then $\text{PH} \subseteq \text{AWPP}$.*

Proof Suppose that $\text{NQP} \subseteq \text{BQP}$. Since $\text{NQP} = \text{coC=P}$ [FGHP98,YY99], $\text{BQP} \subseteq \text{AWPP}$ [FR99], and BQP is closed under complementation, we get that $\text{C=P} \subseteq \text{BQP} \subseteq \text{AWPP}$. By Theorem 5.1 and 5.4(a), it follows that $\text{PH} \subseteq \text{UP}^{\text{C=P}} \subseteq \text{UP}^{\text{AWPP}} \subseteq \text{AWPP}$. ■ Since AWPP is low for PP , we also conclude that if $\text{NQP} \subseteq \text{BQP}$, then PH is low for PP . However, the mere PP -lowness of PH can also be derived easily from known results as follows. By Toda and Ogihara [TO92] and Tarui [Tar93], we have that $\text{PH} \subseteq \text{BPP}^{\text{C=P}}$. Since $\text{coC=P} = \text{NQP}$ and BPP is low for PP relative to all oracles, we get that $\text{PP}^{\text{PH}} \subseteq \text{PP}^{\text{BPP}^{\text{C=P}}} = \text{PP}^{\text{BPP}^{\text{NQP}}} \subseteq \text{PP}^{\text{NQP}}$. Hence, under the assumption that $\text{NQP} \subseteq \text{BQP}$, it follows that $\text{PP}^{\text{PH}} \subseteq \text{PP}^{\text{BQP}}$, and so by the PP -lowness of BQP , we get that $\text{PP}^{\text{PH}} \subseteq \text{PP}^{\text{BQP}} \subseteq \text{PP}$. We emphasize that the interest of Theorem 5.4 lies in deriving the stronger result $\text{NQP} \subseteq \text{BQP} \implies \text{PH} \subseteq \text{AWPP}$ instead of the mere lowness of PH for PP . Assuming that $\text{NQP} \subseteq \text{BQP}$, Corollary 5.5 provides an intuition on the position of the polynomial hierarchy relative to the classes that are low for PP , whereas it is not possible to get the same intuition with the mere lowness result of the polynomial hierarchy.

In the rest of this section, we turn our attention to closure properties of WPP . In Theorem 5.6, we prove that WPP is closed under the polynomial-time truth-table reduction, while its closure under the stronger \leq_T^{UP} reduction is contained in coC=P (in every relativized world). By Corollary 5.5, $\text{NQP} \subseteq \text{BQP}$ implies $\text{PH} \subseteq \text{AWPP}$. On the other hand, Theorem 5.6(b) along with Theorem 5.1 allows us to show that the stronger hypothesis, namely $\text{NQP} \subseteq \text{EQP}$, implies a stronger conclusion about the polynomial hierarchy, namely $\text{PH} \subseteq \text{EQP}$.

Theorem 5.6 (a) *WPP is closed under polynomial-time truth-table reductions.* (b) $\text{UP}^{\text{WPP}} \subseteq \text{coC=P}$.

Proof We will prove (a). The proof for (b) is similar. The proof of (a) uses ideas reminiscent of the proof of $\text{GapP}^{\text{SPP}} = \text{GapP}$ by Fenner, Fortnow, and Kurtz [FFK94]. Let L be an arbitrary language accepted by an oracle DPTM M with nonadaptive access to an oracle $B \in \text{WPP}$. Then, there is an NPTM N_B and an FP function f_B with $0 \notin \text{range}(f_B)$ such that, for all $x \in \Sigma^*$, if $x \in B$, then $\text{gap}_{N_B}(x) = f_B(x)$, and if $x \notin B$, then $\text{gap}_{N_B}(x) = 0$. Without loss of generality, we may assume that, on any input x , $M^A(x)$ makes exactly $p(|x|)$ queries (in parallel, i.e., non-adaptively) where p is some polynomial. We define an NPTM M' as follows. M' , on an arbitrary input x , simulates $M(x)$ to get a list of queries that $M(x)$ asks of its oracle. M' then nondeterministically guesses the answers to these queries. Let $q_1, q_2, \dots, q_{p(|x|)}$ be the queries that $M(x)$ asks of its oracle, and let $a_1, a_2, \dots, a_{p(|x|)}$ be the corresponding guessed answers (for each i , $a_i \in \{0, 1\}$) along an arbitrary computation path ρ . On path ρ , M' does the following. If $M(x)$, with answers $a_1, a_2, \dots, a_{p(|x|)}$ to its queries rejects, then $M'(x)$ along path ρ generates a gap of 0 by guessing a nondeterministic bit and accepting on the path if and only if the guessed bit is 1. On the other hand, if $M(x)$, with answers $a_1, a_2, \dots, a_{p(|x|)}$ to its queries accepts, then $M'(x)$ generates a gap of $g(\langle x, \rho \rangle)$ extending ρ , where g is defined, for each x and ρ , as follows. (Note that in the definition of g below, $q_1, q_2, \dots, q_{p(|x|)}$ are the queries that $M(x)$ asks of its oracle and $a_1, a_2, \dots, a_{p(|x|)}$ are the answers to these queries guessed on the path ρ .) $g(\langle x, \rho \rangle) =_{df} \prod_{i=1}^{p(|x|)} g_i$, where, for

each $1 \leq i \leq p(|x|)$, g_i is defined as follows:

$$g_i =_{df} \begin{cases} gap_{N_B}(q_i) & \text{if } a_i = 1, \\ f_B(q_i) - gap_{N_B}(q_i) & \text{if } a_i = 0. \end{cases}$$

Since GapP is closed under subtraction and polynomial times iterated multiplication [FFK94], g is indeed a GapP function. It is easy to see, that for each $1 \leq i \leq p(|x|)$,

$$g_i = \begin{cases} f_B(q_i) & \text{if } a_i = \chi_B(q_i), \\ 0 & \text{otherwise.} \end{cases}$$

Since there is exactly one path ρ in $M'(x)$ that guesses the answers of all the queries correctly, it follows that $gap_{M'}(x) = \prod_{i=1}^{p(|x|)} f_B(q_i)$ if $M(x)$ accepts, and $gap_{M'}(x) = 0$ otherwise. Define f' as follows. For each $x \in \Sigma^*$, $f'(x) = \prod_{i=1}^{p(|x|)} f_B(q_i)$, where $q_1, q_2, \dots, q_{p(|x|)}$ are the queries asked by $M(x)$ of its oracle. Since $f_B \in \text{FP}$, $f' \in \text{FP}$. Thus, for all $x \in \Sigma^*$, if $x \in L$, then $gap_{M'}(x) = f'(x)$, otherwise (i.e., if $x \notin L$) $gap_{M'}(x) = 0$. Thus, M' and f' witness that L is in WPP. Therefore, WPP is closed under polynomial-time truth-table reductions. \blacksquare

Corollary 5.7 *If $\text{NQP} \subseteq \text{EQP}$, then $\text{PH} \subseteq \text{EQP}$.*

Proof Suppose that $\text{NQP} \subseteq \text{EQP}$. Since $\text{NQP} = \text{coC=P}$ [FGHP98,YY99], $\text{EQP} \subseteq \text{LWPP}$ [FR99], and EQP is closed under complementation, we get that $\text{C=P} \subseteq \text{EQP} \subseteq \text{LWPP} \subseteq \text{WPP}$. By Theorems 5.1 and 5.6(b), and our assumption that $\text{NQP} \subseteq \text{EQP}$, it follows that $\text{PH} \subseteq \text{UP}^{\text{C=P}} \subseteq \text{UP}^{\text{WPP}} \subseteq \text{coC=P} = \text{NQP} \subseteq \text{EQP}$. \blacksquare

Theorem 5.6 states that WPP is closed under truth-table reductions. In Theorem 5.9, we will prove that relativizable techniques cannot improve this to polynomial-time Turing reductions. This gives a partial answer to an open question raised by Fenner, Fortnow, and Kurtz [FFK94]. They showed that SPP and LWPP are closed under polynomial-time Turing reductions, but left open the corresponding problem for WPP: Is WPP closed under polynomial-time Turing reductions? Since LWPP is closed under polynomial-time Turing reductions in every relativized world [FFK94], Theorem 5.9 also yields an oracle separating the seemingly similar classes LWPP and WPP.

Recently, Spakowski and Tripathi [ST04b,ST04a] used ideas in the proof of Theorem 5.9 to construct a relativized world where $\text{UP} \cap \text{coUP}$ is not low for LWPP as well as for WPP. Using this result, they were able to settle an open question of Fenner, Fortnow, and Kurtz [FFK94]: Is WPP *uniformly* gap-definable? They showed that both LWPP and WPP are not *uniformly* gap-definable. (See [FFK94] for the definition of *uniform* and *non-uniform* gap-definability.) Their result makes both LWPP and WPP exceptional compared to most other uniformly gap-definable counting classes (such as PP, C=P, Mod_kP , and SPP), since LWPP and WPP are natural counting classes that are nonuniformly gap-definable but are not uniformly gap-definable. The ideas used in the proof of Theorem 5.9 might have more applications in constructing oracles involving counting classes and, therefore, might be helpful in understanding the relativized complexity of quantum classes such as EQP, BQP, and NQP.

We now prove that, relative to an oracle, WPP is not closed under polynomial-time Turing reductions. Before we state and prove Theorem 5.9, we state a lemma that will be needed in the proof of this result.

Lemma 5.8 [RS62] *For every $n \geq 17$, the number of primes less than or equal to n , $\pi(n)$, satisfies*

$$n/\ln n < \pi(n) < 1.25506 n/\ln n.$$

Theorem 5.9 $(\exists A)[P^{\text{WPP}^A} \not\subseteq \text{WPP}^A]$.

Proof For any $\alpha \in \Sigma^*$, let $\text{pos}(\alpha) = \text{num}(\alpha) - 2^{|\alpha|} + 1$ represent the lexicographic position of α among strings of length $|\alpha|$. For every set $A \subseteq \Sigma^*$, $\alpha \in \Sigma^*$, and $n \in \mathbb{N}$, we define “Witcount,” “Promise,” and “Boundary” as follows.

$$\text{Witcount}(A, \alpha) = |\{x \in \Sigma^* \mid |x| = |\alpha| \wedge \alpha x \in A\}|,$$

$$\begin{aligned} \text{Promise}(A, n) \equiv & (\forall \alpha \in \Sigma^n)[\text{Witcount}(A, \alpha) = 0 \vee \text{Witcount}(A, \alpha) = \text{pos}(\alpha)] \wedge \\ & (\forall \alpha_1, \alpha_2 \in \Sigma^n)[\text{pos}(\alpha_1) \leq \text{pos}(\alpha_2) \wedge \text{Witcount}(A, \alpha_2) \neq 0 \Rightarrow \\ & \text{Witcount}(A, \alpha_1) \neq 0], \text{ and} \end{aligned}$$

$$\text{Boundary}(A, n) = \max\{\text{pos}(\alpha) \mid |\alpha| = n \wedge \text{Witcount}(A, \alpha) \neq 0\}.$$

For every set $A \subseteq \Sigma^*$, define L_A as follows:

$$L_A = \{0^n \mid \text{Boundary}(A, n) \equiv 1 \pmod{2}\}.$$

Clearly, if A satisfies $\text{Promise}(A, n)$ at each length n , then L_A is in P^{WPP^A} (using binary search along the strings α with $|\alpha| = n$). We construct an oracle A such that, for each n , $\text{Promise}(A, n)$ is true and $L_A \notin \text{WPP}^A$. Let $(N_s, M_s, p_s)_{s \geq 1}$ be an enumeration of all triples such that N_s is a nondeterministic polynomial-time oracle Turing machine, M_s is a deterministic polynomial-time oracle transducer, p_s is a polynomial, and the running time of both N_s and M_s is bounded by p_s regardless of the oracle. We need the following technical lemmas, the proofs of which are given later at the end of this proof.

Lemma 5.10 *Let $N, p \in \mathbb{N}$, where $1 < p \leq N/2$. Let $s(y_1, y_2, \dots, y_N)$ be a multilinear polynomial with rational coefficients, where each monomial has exactly $p - 1$ different variables. Suppose that for some $\text{val} \in \mathbb{Q}$, it holds that $s(y_1, y_2, \dots, y_N) = \text{val}$ for every $y_1, y_2, \dots, y_N \in \{0, 1\}$ with $\sum_{i=1}^N y_i = p$. Then each monomial in $s(y_1, y_2, \dots, y_N)$ has the same rational coefficient, i.e.,*

$$s(y_1, y_2, \dots, y_N) = \sum_{1 \leq i_1 < i_2 < \dots < i_{p-1} \leq N} (\text{val}/p) \cdot y_{i_1} y_{i_2} \cdots y_{i_{p-1}}.$$

Lemma 5.11 *Let $N \in \mathbb{N}$ and p be a prime with $p \leq N/2$. Let $s(y_1, y_2, \dots, y_N)$ be a multilinear polynomial of total degree $< p$ with integer coefficients. If for some $\text{val} \in \mathbb{Z}$, it holds that*

1. $s(0, 0, \dots, 0) = 0$, and

2. $s(y_1, y_2, \dots, y_N) = \text{val}$, for every $y_1, y_2, \dots, y_N \in \{0, 1\}$ with $\sum_{i=1}^N y_i = p$,

then $p \mid \text{val}$.

The oracle A is constructed in stages. In stage s , the membership in A of strings of length $2n_s$ is decided and the initial segment A_{s-1} is extended to A_s . Our choice of n_s guarantees that the oracle extension in stage s does not affect the computation in earlier stages. Set $A_0 := \emptyset$ and $n_0 := 17$.

Stage s , where $s \geq 1$: Let n_s be large enough so that the previous stages are not affected and $2^{n_s} > 4n_s^2 p_s(n_s)$. We diagonalize against nondeterministic polynomial-time oracle Turing machine N_s and deterministic polynomial-time oracle transducer M_s . Let val be the value computed by $M_s^{A_{s-1}}(0^{n_s})$. Without loss of generality, we assume that $\text{val} \neq 0$. Let $T = \{\alpha \in \Sigma^{2n_s} \mid M_s^{A_{s-1}}(0^{n_s}) \text{ queries } \alpha\}$.

($\star\star\star$) Choose a set B , $B \subseteq \overline{T} \cap \Sigma^{2n_s}$, satisfying $\text{Promise}(B, n_s)$ such that the following holds:

$$\text{Boundary}(B, n_s) \equiv 1 \pmod{2} \quad \text{and} \quad \text{gap}_{N_s^{A_{s-1} \cup B}}(0^{n_s}) \neq \text{val}, \text{ or}$$

$$\text{Boundary}(B, n_s) \equiv 0 \pmod{2} \quad \text{and} \quad \text{gap}_{N_s^{A_{s-1} \cup B}}(0^{n_s}) \neq 0.$$

Let $A_s := A_{s-1} \cup B$. Clearly, the construction guarantees that $L_A \notin \text{WPP}^A$. The feasibility of the construction follows from the following claim.

Claim 7 For each $s \geq 1$, there exists an oracle extension B satisfying ($\star\star\star$).

Proof Suppose that in stage s no set B satisfying ($\star\star\star$) exists. Then, for every $B \subseteq \overline{T} \cap \Sigma^{2n_s}$ satisfying $\text{Promise}(B, n_s)$, the following hold.

$$\text{Boundary}(B, n_s) \equiv 1 \pmod{2} \implies \text{gap}_{N_s^{A_{s-1} \cup B}}(0^{n_s}) = \text{val}, \text{ and} \quad (5.vii)$$

$$\text{Boundary}(B, n_s) \equiv 0 \pmod{2} \implies \text{gap}_{N_s^{A_{s-1} \cup B}}(0^{n_s}) = 0. \quad (5.viii)$$

Let $U = \{\alpha \in \Sigma^{n_s} \mid \text{pos}(\alpha) \text{ is prime, and } 2^{n_s-2} \leq \text{pos}(\alpha) \leq \frac{3}{2} \cdot 2^{n_s-2}\}$. Fix an arbitrary $\alpha \in U$. Choose a set $C_\alpha \subseteq \overline{T} \cap \Sigma^{2n_s}$ satisfying (a) $\text{Promise}(C_\alpha, n_s)$, and (b) $\text{Boundary}(C_\alpha, n_s) = \text{pos}(\alpha) - 1$. Such a set C_α always exists because $2^{n_s} - p_s(n_s) > \frac{3}{2} \cdot 2^{n_s-2}$. Statements (5.vii) and (5.viii) in particular imply that, for all $D_\alpha \subseteq \overline{T} \cap \alpha \Sigma^{n_s}$, it holds that

$$\text{Witcount}(D_\alpha, \alpha) = 0 \implies \text{gap}_{N_s^{A_{s-1} \cup C_\alpha \cup D_\alpha}}(0^{n_s}) = 0, \text{ and} \quad (5.ix)$$

$$\text{Witcount}(D_\alpha, \alpha) = \text{pos}(\alpha) \implies \text{gap}_{N_s^{A_{s-1} \cup C_\alpha \cup D_\alpha}}(0^{n_s}) = \text{val}. \quad (5.x)$$

Henceforth, we use N to denote $|\overline{T} \cap \alpha \Sigma^{n_s}|$. Let x_1, x_2, \dots, x_N be the lexicographic enumeration of the strings in $\overline{T} \cap \alpha \Sigma^{n_s}$. We define s_α to be the function $\{0, 1\}^N \rightarrow \mathbb{Z}$ that has the following property. For all $D_\alpha \subseteq \overline{T} \cap \alpha \Sigma^{n_s}$,

$$s_\alpha(\chi_{D_\alpha}(x_1), \chi_{D_\alpha}(x_2), \dots, \chi_{D_\alpha}(x_N)) = \text{gap}_{N_s^{A_{s-1} \cup C_\alpha \cup D_\alpha}}(0^{n_s}). \quad (5.xi)$$

We will show that s_α can be represented by a multilinear polynomial having low total degree. For arbitrary $z_1, z_2, \dots, z_N \in \{0, 1\}$, we call a computation path ρ of $N_s^{(\cdot)}(0^{n_s})$ “ (z_1, z_2, \dots, z_N) -allowable” if, along ρ , all queries $q \in A_{s-1} \cup C_\alpha$ have a “yes” answer, all

queries $q \notin A_{s-1} \cup C_\alpha \cup (\bar{T} \cap \alpha \Sigma^{n_s})$ have a “no” answer, all queries x_i with $z_i = 1$ are answered “yes,” and all queries x_i with $z_i = 0$ are answered “no.” Let $z_1, z_2, \dots, z_N \in \{0, 1\}$ and let ρ be a (z_1, z_2, \dots, z_N) -allowable path of $N_s^{(\cdot)}(0^{n_s})$. Let $x_{i_1}, x_{i_2}, \dots, x_{i_\ell}$, where $\ell \leq p_s(n_s) < 2^{n_s-2}/n_s^2 < 2^{n_s-2}$, be the distinct queries to strings in $\bar{T} \cap \alpha \Sigma^{n_s}$ along ρ . Create a monomial $\text{mono}(\rho)$ that is the product of terms γ_k , $k = 1, 2, \dots, \ell$, where $\gamma_k = y_{i_k}$ if $z_{i_k} = 1$, and $\gamma_k = (1 - y_{i_k})$ otherwise. Let

$$s'_\alpha(y_1, y_2, \dots, y_N) = \sum_{z_1, z_2, \dots, z_N \in \{0, 1\}} \sum_{\rho: \rho \text{ is } (z_1, z_2, \dots, z_N)\text{-allowable}} \text{sign}(N_s, 0^{n_s}, \rho) \cdot \text{mono}(\rho).$$

It is easy to see that the thus constructed multilinear polynomial $s'_\alpha(y_1, y_2, \dots, y_N)$ coincides with s_α on $\{0, 1\}^N$, and has total degree $\leq p_s(n_s) < 2^{n_s-2}/n_s^2 < \text{pos}(\alpha) < N/2$. Statements (5.ix) and (5.x) imply that for all $z_1, z_2, \dots, z_N \in \{0, 1\}$ such that $\sum_{i=1}^N z_i = \text{pos}(\alpha)$,

$$s_\alpha(z_1, z_2, \dots, z_N) = \text{val} \text{ and } s_\alpha(0, 0, \dots, 0) = 0.$$

It follows from Lemma 5.11 that $\text{pos}(\alpha) \mid \text{val}$. Therefore, for each $\alpha \in U$, $\text{pos}(\alpha) \mid \text{val}$. Hence,

$$\text{val} \geq \prod_{\alpha \in U} \text{pos}(\alpha) \geq 2^{|U|} \geq 2^{\pi(\frac{3}{2} \cdot 2^{n_s-2}) - \pi(2^{n_s-2})} \geq 2^{2^{n_s-2}/n_s^2} > 2^{p_s(n_s)},$$

where the fourth inequality follows from Lemma 5.8 and the fifth inequality follows because, $2^{n_s} > 4n_s^2 p_s(n_s)$. However, $\text{val} \leq 2^{p_s(n_s)}$, because the running time of $M_s^{(\cdot)}(0^{n_s})$ is bounded by $p_s(n_s)$ regardless of the oracle. Thus, for each $s \geq 1$, A_{s-1} can always be extended in stage s . This completes the proofs of Claim 7 and Theorem 5.9. \blacksquare

Proof of Lemma 5.10 Assume that the hypothesis of the lemma is true. For each $1 \leq i \leq N$, we identify variable y_i by its index i and identify a monomial $\prod_{j=1}^k y_{i_j}$ by the set of indices $\{i_1, i_2, \dots, i_k\}$. Let \mathcal{A} denote the collection of all subsets of $\{1, 2, \dots, N\}$ of size p and let \mathcal{B} denote the collection of all subsets of $\{1, 2, \dots, N\}$ of size $p-1$. W.l.o.g, we assume that the elements of \mathcal{A} and \mathcal{B} are ordered in an arbitrary but fixed manner. We use $m = \binom{N}{p}$ to denote the size of \mathcal{A} and $n = \binom{N}{p-1}$ to denote the size of \mathcal{B} . Let A_i , where $1 \leq i \leq m$, denote the i th element of \mathcal{A} , and B_j , where $1 \leq j \leq n$, denote the j th element of \mathcal{B} . For a 2-dimensional matrix $M_{m \times n}$, let $\text{Row}(M, i)$, where $1 \leq i \leq m$, denote the i th row of M .

The condition $s(y_1, y_2, \dots, y_N) = \text{val}$ for every $y_1, y_2, \dots, y_N \in \{0, 1\}$ with $\sum_{i=1}^N y_i = p$, as given in the hypothesis, can be expressed in terms of a matrix equation $M_{m \times n} X_{n \times 1} = b_{m \times 1}$. Here $M_{m \times n}$ is a 0-1 matrix whose (i, j) entry, $M[i, j]$, is one if $A_i \supseteq B_j$ and is zero otherwise, $X_{n \times 1}$ is a column vector with the j th entry, $X[j]$, is a variable that denotes the coefficient of monomial B_j , and $b_{m \times 1}$ is a column vector with each entry b_i , $1 \leq i \leq m$, equals val . By assigning all coefficients $X[j]$ the value val/p , we obtain clearly a solution for the system of equations. Hence it is sufficient to prove that the solution is unique, i.e., $\text{rank}(M) = n$. We show that it is possible to express each canonical vector $e_i = [0, \dots, 0, 1, 0, \dots, 0]$, $1 \leq i \leq n$, as a linear combination of row vectors in M . W.l.o.g, we show that for vector $e_1 = [1, 0, \dots, 0]$.

Form a matrix $\widehat{M}_{p \times n}$ in the following way. Row k , where $1 \leq k \leq p$, of \widehat{M} is the sum of all rows i in M with $\|A_i \cap B_1\| = p - k$. Note that there is at least one row i with $\|A_i \cap B_1\| = p - k$. This follows from $\|\{1, 2, \dots, N\} - B_1\| \geq p$, which is true because of the condition $p \leq N/2$.

Claim 8 *The matrix \widehat{M} has the following properties. For every row k ($1 \leq k \leq p$),*

1. $\widehat{M}[k, j_1] = \widehat{M}[k, j_2]$ whenever $\|B_{j_1} \cap B_1\| = \|B_{j_2} \cap B_1\|$,
2. $\widehat{M}[k, j] \neq 0$ for all j with $\|B_j \cap B_1\| = p - k$,
3. $\widehat{M}[k, j] = 0$ for all j with $\|B_j \cap B_1\| > p - k$.

Proof To see (1), note that for fixed k , the cardinality of the set $\{A_i \in \mathcal{A} \mid \|A_i \cap B_1\| = p - k \wedge A_i \supseteq B_j\}$ depends only on the number of elements of $B_j \in \mathcal{B}$ that are also in B_1 . Hence for every j_1 and j_2 , with $\|B_{j_1} \cap B_1\| = \|B_{j_2} \cap B_1\|$, it holds that

$$\|\{A_i \in \mathcal{A} \mid \|A_i \cap B_1\| = p - k \wedge A_i \supseteq B_{j_1}\}\| = \|\{A_i \in \mathcal{A} \mid \|A_i \cap B_1\| = p - k \wedge A_i \supseteq B_{j_2}\}\|.$$

Statement (1) follows immediately. For the proof of (2), we have to verify that for every k and j ,

$$\mathcal{S} =_{df} \{A_i \in \mathcal{A} \mid \|A_i \cap B_1\| = p - k \wedge A_i \supseteq B_j\} \neq \emptyset \text{ if } \|B_j \cap B_1\| = p - k.$$

It is easy to see that, if $\|B_j \cap B_1\| = p - k$ then \mathcal{S} has as element any set $B_j \cup \{g\}$, where $g \notin B_1$. Finally, to show (3), note that for every k and j ,

$$\|\{A_i \in \mathcal{A} \mid \|A_i \cap B_1\| = p - k \wedge A_i \supseteq B_j\}\| = 0 \text{ if } \|B_j \cap B_1\| \geq p - k + 1.$$

since $\|B_j \cap B_1\| \geq p - k + 1$ and $A_i \supseteq B_j$ implies that $\|A_i \cap B_1\| \geq p - k + 1$. ■

To complete the proof of the lemma, we show that the structure of the matrix \widehat{M} stated in Claim 8 implies that e_1 can be expressed as a linear combination of row vectors of \widehat{M} , and hence also as linear combination of row vectors of M . We construct a matrix $M'_{p \times p}$ from $\widehat{M}_{p \times n}$, which will turn out to have full rank. From Claim 8(1), we know that column j_1 and column j_2 of \widehat{M} are equal whenever $\|B_{j_1} \cap B_1\| = \|B_{j_2} \cap B_1\|$. Thus it makes sense to define a matrix M' eliminating all these duplicate columns from \widehat{M} . We define column ℓ , where $1 \leq \ell \leq p$, of M' to equal column j of \widehat{M} for some j with $\|B_j \cap B_1\| = p - \ell$. Note that column 1 of M' corresponds uniquely to column 1 of \widehat{M} . Claim 8(3) implies that the matrix M' is an upper triangular matrix, and from Claim 8(2), it follows that all diagonal elements in M' are $\neq 0$. Hence M' has full rank. In particular, row vector $[1, 0, \dots, 0]_{1 \times p}$ can be written as a linear combination of rows in M' . Suppose

$$[1, 0, \dots, 0]_{1 \times p} = \sum_{k=1}^p c_k \cdot \text{Row}(M', k),$$

for $c_1, \dots, c_k \in \mathbb{Q}$. Then

$$e_1 = [1, 0, \dots, 0]_{1 \times n} = \sum_{k=1}^p c_k \cdot \text{Row}(\widehat{M}, k),$$

because column 1 of \widehat{M} equals column 1 of M' , and all other columns of \widehat{M} equal a column j in M' with $p \geq j > 1$. Thus, Lemma 5.10 is proved. \blacksquare

Proof of Lemma 5.11 We transform $s(y_1, y_2, \dots, y_N)$ to a multilinear polynomial $s'(y_1, y_2, \dots, y_N)$ with the following properties:

- (1) All monomials in s' have exactly $p - 1$ different variables.
- (2) $s'(y_1, y_2, \dots, y_N) = s(y_1, y_2, \dots, y_N) = val \in \mathbb{Z}$, for all $y_1, y_2, \dots, y_N \in \{0, 1\}$ with $\sum_{i=1}^N y_i = p$.
- (3) The coefficients of the monomials in s' have the form a/b , where $a, b \in \mathbb{Z}$, and $p \nmid b$.

Since $s(0, 0, \dots, 0) = 0$, the polynomial s has no monomial with degree 0. Let $t(y_1, y_2, \dots, y_N) = a \prod_{i \in A} y_i$ be a monomial, where $A \subseteq \{1, 2, \dots, N\}$ and $1 \leq \|A\| = \ell < p - 1$. Define the multilinear polynomial u_t by

$$u_t(y_1, y_2, \dots, y_N) = \sum_{\substack{B \subseteq \{1, \dots, N\} - A, \\ \|A \cup B\| = p - 1}} \left(\frac{a}{p - \ell} \prod_{i \in A \cup B} y_i \right). \quad (5.xii)$$

Claim 9 For all $y_1, y_2, \dots, y_N \in \{0, 1\}$ with $\sum_{i=1}^N y_i = p$, $u_t(y_1, y_2, \dots, y_N) = t(y_1, y_2, \dots, y_N)$.

Proof Let $y_1, y_2, \dots, y_N \in \{0, 1\}$ be such that $\sum_{i=1}^N y_i = p$. Depending on the choice in the selection of y_1, y_2, \dots, y_N , we have two cases.

Case 1: $t(y_1, y_2, \dots, y_N) = 0$.

Then, clearly $u_t(y_1, y_2, \dots, y_N) = t(y_1, y_2, \dots, y_N) = 0$.

Case 2: $t(y_1, y_2, \dots, y_N) = a$.

Then $y_i = 1$ for all $i \in A$. Let $D = \{i \mid i \in \{1, 2, \dots, N\} - A \wedge y_i = 1\}$. Clearly, $\|D\| = p - \ell$. In the sum on the right hand side of Eq. (5.xii), only B 's with $B \subseteq D$ contribute a value $\neq 0$. The sets B have always cardinality $p - 1 - \ell$. Hence, there are exactly $\binom{p - \ell}{p - 1 - \ell} = p - \ell$ sets B contributing the value $a/(p - \ell)$ to the sum.

Thus, Claim 9 is proved. \blacksquare

Transform polynomial $s(y_1, y_2, \dots, y_N)$ to polynomial $s'(y_1, y_2, \dots, y_N)$ by substituting each monomial $t(y_1, y_2, \dots, y_N)$ in $s(y_1, y_2, \dots, y_N)$ of degree $< p - 1$ by the corresponding polynomial $u_t(y_1, y_2, \dots, y_N)$. Since $\{a/b \mid a, b \in \mathbb{Z} \wedge b \neq 0 \wedge p \nmid b\}$ is closed under addition, it follows that the polynomial $s'(y_1, y_2, \dots, y_N)$ satisfies properties (1), (2), and (3) stated at the beginning of the proof. Lemma 5.10 implies that all the coefficients of monomials in $s'(y_1, y_2, \dots, y_N)$ are equal to val/p . Thus to match with property (3) of polynomial $s'(y_1, y_2, \dots, y_N)$, val/p must be an integer. It follows that $p \mid val$. This completes the proof of Lemma 5.11. \blacksquare

As an immediate corollary of Theorem 5.9 and the fact that LWPP is closed under polynomial-time Turing reductions in all relativized worlds [FFK94], we get the following result.

Corollary 5.12 $(\exists A)[WPP^A \not\subseteq LWPP^A]$.

6 Conclusion and Open Problems

In this paper, we used classical complexity classes to study the complexity of quantum complexity classes EQP, BQP, and NQP. In particular, we used counting classes to prove relativized separations, strong separations, and conditional collapses. We mention several open problems.

Theorem 3.1 shows that, in some relativized world, ZPP is not contained in WPP. Theorem 4.9 shows that, relative to an oracle, RP contains a set that is immune to $C=P$. It would be interesting to know if the oracle separation of ZPP from WPP can be strengthened to an immunity separation result. This would also imply, as a corollary, an immunity separation of ZPP from EQP.

Another interesting open issue is whether or not the conditional collapse results in this paper can be improved. We show in Corollary 5.7 that $NQP \subseteq EQP \implies PH \subseteq EQP$. This is an analog of the collapse known in the classical case, namely $NP \subseteq P \implies PH \subseteq P$. Can we similarly prove that $NQP \subseteq BQP \implies PH \subseteq BQP$? This would be another analog of the collapse known in the classical case [Zac88].

Acknowledgments

We thank Lane Hemaspaandra for helpful advice and guidance throughout the project. We are grateful to the anonymous referees for helpful comments.

References

- [ADH97] L. Adleman, J. DeMarrais, and M. Huang. Quantum computability. *SIAM Journal on Computing*, 26(5):1524–1540, 1997.
- [All86] E. Allender. The complexity of sparse sets in P. In *Proceedings of the 1st Structure in Complexity Theory Conference*, pages 1–11. Springer-Verlag *Lecture Notes in Computer Science #223*, June 1986.
- [AR88] E. Allender and R. Rubinfeld. P-printable sets. *SIAM Journal on Computing*, 17(6):1193–1202, 1988.
- [BBBV97] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, October 1997.

- [BBHT98] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–506, 1998.
- [BCS92] D. Bovet, P. Crescenzi, and R. Silvestri. A uniform approach to define complexity classes. *Theoretical Computer Science*, 104(2):263–283, 1992.
- [BCS95] D. Bovet, P. Crescenzi, and R. Silvestri. Complexity classes and sparse oracles. *Journal of Computer and System Sciences*, 50(3):382–390, 1995.
- [Bei91] R. Beigel. Relativized counting classes: Relations among thresholds, parity, and mods. *Journal of Computer and System Sciences*, 42(1):76–96, 1991.
- [BG92] R. Beigel and J. Gill. Counting classes: Thresholds, parity, mods, and fewness. *Theoretical Computer Science*, 103(1):3–23, 1992.
- [BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.
- [BI87] M. Blum and R. Impagliazzo. Generic oracles and oracle classes. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 118–126, October 1987.
- [BJY90] D. Bruschi, D. Joseph, and P. Young. Strong separations for the boolean hierarchy over RP. *International Journal of Foundations of Computer Science*, 1(3):201–218, 1990.
- [Bru92] D. Bruschi. Strong separations of the polynomial hierarchy with oracles: Constructive separations by immune and simple sets. *Theoretical Computer Science*, 102(2):215–252, 1992.
- [BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.
- [CGH⁺88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.
- [CH90] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory*, 23(2):95–106, 1990.
- [dGV02] M. de Graaf and P. Valiant. Comparing EQP and MOD_{p^k}P using polynomial degree lower bounds. Technical Report quant-ph/0211179, Quantum Physics, 2002.
- [Fen03] S. Fenner. PP-lowness and a simple definition of AWPP. *Theory of Computing Systems*, 36(2):199–212, 2003.
- [FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

- [FFKL03] S. Fenner, L. Fortnow, S. Kurtz, and L. Li. An oracle builder’s toolkit. *Information and Computation*, 182(2):95–136, 2003.
- [FGHP98] S. Fenner, F. Green, S. Homer, and R. Pruim. Quantum NP is hard for PH. In *Proceedings of the 6th Italian Conference for Theoretical Computer Science*, pages 241–252. World Scientific, 1998.
- [For99] L. Fortnow. Relativized worlds with an infinite hierarchy. *Information Processing Letters*, 69(6):309–313, 1999.
- [FR99] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999.
- [Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.
- [GP86] L. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of boolean functions. *Theoretical Computer Science*, 43(1):43–58, 1986.
- [GP01] F. Green and R. Pruim. Relativized separation of EQP from P^{NP} . *Information Processing Letters*, 80:257–260, 2001.
- [Gre93] F. Green. On the power of deterministic reductions to $C=P$. *Mathematical Systems Theory*, 26(2):215–233, 1993.
- [Gro96] L. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 212–219, May 1996.
- [Gru99] J. Gruska. *Quantum Computing*. McGraw-Hill, New York, 1999.
- [Hal02] S. Hallgren. Polynomial-time quantum algorithms for Pell’s equation and the principal ideal problem. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 653–658, New York, 2002. ACM Press.
- [Her90] U. Hertrampf. Relations among MOD-classes. *Theoretical Computer Science*, 74(3):325–328, 1990.
- [HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Springer, 2002.
- [Ko90] K. Ko. A note on separating the relativized polynomial-time hierarchy by immune sets. *RAIRO Theoretical Informatics and Applications*, 24(3):229–240, 1990.
- [KSTT92] J. Köbler, U. Schöning, S. Toda, and J. Torán. Turing machines with few accepting computations and low sets for PP. *Journal of Computer and System Sciences*, 44(2):272–286, 1992.

- [Li93] L. Li. On PP-low classes. Technical Report TR-93-03, Department of Computer Science, University of Chicago, May 14 1993.
- [Luc78] E. Lucas. Sur les congruences des nombres eulériens et les coefficients différentiels des fonctions trigonométriques, suivant un module premier. *Bulletin de la S.M.F.*, 6:49–54, 1878.
- [NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. CUP, 2000.
- [OH93] M. Ogiwara and L. Hemachandra. A complexity theory for feasible closure properties. *Journal of Computer and System Sciences*, 46(3):295–325, 1993.
- [PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.
- [Rot99] J. Rothe. Immunity and simplicity for exact counting and other counting classes. *RAIRO Theoretical Informatics and Applications*, 33(2):159–176, 1999.
- [RS62] J. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6:64–94, 1962.
- [Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
- [Sim75] J. Simon. *On Some Central Problems in Computational Complexity*. PhD thesis, Cornell University, Ithaca, N.Y., January 1975. Available as Cornell Department of Computer Science Technical Report TR75-224.
- [ST04a] H. Spakowski and R. Tripathi. Degree bounds on polynomials and relativization theory. Technical Report TR820, Department of Computer Science, University of Rochester, December 2003 (revised October 2004).
- [ST04b] H. Spakowski and R. Tripathi. Degree bounds on polynomials and relativization theory. In *Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science*, pages 105–118. Kluwer Academic Publishers, August 2004.
- [Tar91] J. Tarui. Degree complexity of boolean functions and its applications to relativized separations. In *Proceedings of the 6th Annual Conference on Structure in Complexity Theory (SCTC '91)*, pages 285–285, Chicago, IL, USA, June 1991. IEEE Computer Society Press.
- [Tar93] J. Tarui. Probabilistic polynomials, AC^0 functions and the polynomial-time hierarchy. *Theoretical Computer Science*, 113(1):167–183, 1993.
- [TO92] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 21(2):316–328, 1992.

- [Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.
- [Tor91] J. Torán. Complexity classes defined by counting quantifiers. *Journal of the ACM*, 38(3):753–774, 1991.
- [Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5(1):20–23, 1976.
- [Vya03] M. Vyalıı. QMA = PP implies that PP contains PH. In *ECCCTR: Electronic Colloquium on Computational Complexity, technical reports*, 2003.
- [Wag86] K. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Informatica*, 23:325–356, 1986.
- [YY99] T. Yamakami and A. Yao. $\text{NQP}_C = \text{co-C=P}$. *Information Processing Letters*, 71(2):63–69, July 1999.
- [Zac88] S. Zachos. Probabilistic quantifiers and games. *Journal of Computer and System Sciences*, 36(3):433–451, 1988.