THEORY OF COMPUTING, Volume 21 (7), 2025, pp. 1–13
www.theoryofcomputing.org

NOTE

Even Quantum Advice is Unlikely to Solve PP

Justin Yirka*

Received May 29, 2024; Revised September 4, 2025; Published October 2, 2025

Abstract. We give a corrected proof that if PP $\subseteq$ BQP/qpoly (probabilistic polynomial time can be efficiently simulated by quantum circuits with quantum advice), then the Counting Hierarchy collapses, as originally claimed by Aaronson (CCC'06). This recovers the related unconditional claim that PP does not have circuits of any fixed-polynomial size n^k even with quantum advice. Our result is based on proving that YQP*, an oblivious version of QMA $\cap$ coQMA, is contained in APP, a PP-low subclass of PP with an arbitrarily small but nonzero promise gap.

1 Introduction

Do reasonably-sized circuits solve hard problems, given they may be chosen non-uniformly for each input size? While directly answering this question for classes such as NP has proven difficult, progress has been made showing conditional results, such as the Karp–Lipton theorem that if NP $\subseteq$ P/poly, then the Polynomial Hierarchy collapses [14], or showing upper bounds

*Supported via Scott Aaronson by a Simons Investigator Award.

ACM Classification: F.1.3

AMS Classification: 68Q15, 68Q12, 68Q06

Key words and phrases: circuit complexity, quantum computing, counting hierarchy, quantum advice, oblivious proofs

 DOI: 10.4086/toc.2025.v021a007

against larger classes, such as PP or NEXP [21, 24]. Exploring further, we can consider quantum computation. In this model, circuits are typically uniformly generated but might accept non-uniform *advice* strings as part of their input. Moreover, quantum circuits can receive not only classical advice strings (BQP/poly), but quantum *advice states* (BQP/qpoly).

In [2], Aaronson proved new quantum circuit lower bounds, among other results. In particular, he gave several results characterizing quantum circuits' ability to solve problems in the class PP, the class of problems decidable by probabilistic polynomial-time algorithms with no promise gap, i. e., which accept with probability at least $1/2$ or strictly less than $1/2$. PP contains both BPP and NP and is contained in PSPACE. Aaronson proved that P^{PP} does not have circuits of size n^k for any fixed constant k even if the circuits use quantum advice states. Second, he claimed a quantum analogue of the Karp–Lipton theorem, showing that if $\mathsf{PP} \subseteq \mathsf{BQP/qpoly}$, then the Counting Hierarchy (CH) collapses to QMA, where the Counting Hierarchy is the infinite sequence of classes $\mathsf{C_1P} = \mathsf{PP}$ and $\mathsf{C_iP} = (\mathsf{C_{i-1}P})^{\mathsf{PP}}$, and where QMA is a quantum analogue of NP. Similarly, he showed that under the stronger assumption $\mathsf{PP} \subseteq \mathsf{BQP/poly}$, using classical advice instead of quantum advice, then $\mathsf{CH} = \mathsf{QCMA}$. Third, Aaronson combined these results to give the unconditional bound that PP does not have classical or quantum circuits of size n^k for any fixed constant k even with quantum advice.[1]

However, Aaronson later noted there was an error in one of the proofs [4]. The first of the above results was unaffected, but the proof of the second result only held under the stronger assumption that $\mathsf{PP} \subseteq \mathsf{BQP/poly}$. This also meant the third result only held for quantum circuits with classical, not quantum, advice. Fortunately, no other results in [2] were affected, but no fix for this bug was forthcoming.

Very briefly, the error was a claim that for oracle classes of the form $\mathsf{C}^{\mathsf{BQP/qpoly}}$, if a machine for the base class C is able to find the quantum advice state that will be used by the oracle machine, then the base machine can "hard-code" the advice state into its oracle queries so that the oracle no longer needs the power to find its own advice, thus reducing $\mathsf{C}^{\mathsf{BQP/qpoly}}$ to $\mathsf{C}^{\mathsf{BQP}}$. This approach works for classes with classical advice, like $\mathsf{C}^{\mathsf{BQP/poly}}$. But, because complexity classes and their associated oracles are defined in terms of (classical) strings as input, there is no way to hard-code a general quantum advice state into a query.

In this note, we give a corrected proof of Aaronson's full claims. We show that if $\mathsf{PP} \subseteq \mathsf{BQP/qpoly}$, then the Counting Hierarchy collapses to QMA and in fact to YQP^*, defined below. Given this correction, Aaronson's proof for the third claim, that PP does not have circuits of size n^k for any fixed constant k even with quantum advice, now goes through.

Our primary technical contribution is to show $\mathsf{YQP}^* \subseteq \mathsf{APP}$. Here, YQP^* (Definition 2.2) is an oblivious version of $\mathsf{QMA} \cap \mathsf{coQMA}$, where "oblivious" means there exists a useful proof state which depends only on the size of the input. Crucially, a YQP^* protocol includes a proof-verification circuit that tests if the given quantum state is a "good" proof, independent of whether a particular input is a YES- or NO-instance. The class APP (Definition 2.3) is a subclass of PP with an arbitrarily small but nonzero promise gap. It is known to have the nice property $\mathsf{PP}^{\mathsf{APP}} = \mathsf{PP}$ [16], i. e., it is PP-low. Thus, we find that YQP^* is also PP-low. Our corrected proof combines this result with the equality $\mathsf{BQP/qpoly} = \mathsf{YQP}^*\mathsf{/poly}$, serendipitously proven by

[1]Slightly earlier, Vinochandran [21] gave a proof that PP does not have *classical* circuits of fixed-polynomial size.

Aaronson with Drucker [5]. Now, instead of following Aaronson's original attempt to collapse $\mathsf{PP}^{\mathsf{PP}}$ to $\mathsf{PP}^{\mathsf{BQP/qpoly}}$ to $\mathsf{PP}^{\mathsf{BQP}}$ to PP, we can collapse $\mathsf{PP}^{\mathsf{PP}}$ to $\mathsf{PP}^{\mathsf{YQP^*/poly}}$ to $\mathsf{PP}^{\mathsf{YQP^*}}$ to PP.

Our results provide stronger implications and improved bounds for quantum circuits with quantum advice and establish new insights into PP-lowness and classes within APP. Known quantum Karp–Lipton style bounds include that if NP ⊆ BQP/qpoly, then $\Pi_2^{\mathsf{p}} \subseteq \mathsf{QMA}^{\mathsf{PromiseQMA}}$ [5], and that if QCMA ⊆ BQP/poly, then QCPH collapses to its second level [6], where QCPH is defined like PH but with quantum verifiers and classical proofs. Our result that if PP ⊆ BQP/qpoly, then CH = YQP* adds to this list. As for unconditional bounds, following Aaronson's unaffected result that P^{PP} does not have quantum circuits with quantum advice of any fixed-polynomial size, our corrected result that PP also does not have such circuits is the first improved bound on fixed-polynomial-size circuits with quantum advice. Regarding PP-lowness, our primary lemma establishes YQP* as the largest natural quantum complexity class known to be PP-low, improving on the fact that BQP is PP-low [10].[2] Finally, our result adds to the verification-related classes known to be contained in APP, including FewP [16] (but not NP), showing that APP contains the oblivious-witness classes YQP* ⊇ YMA* ⊇ YP*.

2 Preliminaries

In this section, we give definitions, state a fact relating quantum circuits to GapP, and recall the technique of in-place error reduction to make a few technical observations that will be useful for our main result. For a deeper introduction to this area, see the textbook by Arora and Barak [7] or the survey by Watrous [23]. For more motivation, see [2, 5].

Non-uniform circuits The classes of non-uniform circuits we discuss, such as P/poly, BQP/poly, and BQP/qpoly, share the following key characteristics. First, the circuits are defined with bounded fan-in and fan-out, in contrast to classes such as AC_0 or QAC_0. Second, the classes consider circuits of polynomial-size, where the size is the number of gates in a circuit. Third, they are defined in terms of circuits or advice that may depend on the size of the problem input (but not on the input itself), with no requirement that the circuit or advice is generated by a uniform algorithm.

Advice is generally considered "trusted", in contrast to the untrusted proofs or witnesses received in classes such as NP. In NP, in a NO-instance, a verifier should not accept given any proof. But in P/poly, a circuit is only guaranteed to be correct when given the correct advice.

The names BQP/poly and BQP/qpoly have been used to refer to similar but distinct classes (cf., BQP/mpoly, BQP/*Qpoly, and the Complexity Zoo [25]). The distinction is in the behavior of the circuit when the "wrong" advice is provided: is a probabilistic circuit required to accept with high or low probability (outside of the promise gap) only when the correct advice is provided, or for all advice? We follow the convention that because advice is considered "trusted", there is no need for a promised behavior when given the wrong advice. An explicit definition of BQP/qpoly,

[2] Morimae and Nishimura [20] gave definitions involving quantum postselection constructed to equal AWPP and APP, which are PP-low.

following this convention, is given below. We follow this same convention for BQP/poly and, later, YQP*/poly.

Definition 2.1 ([5])**.** A language L is in BQP/qpoly if there exists a polynomial-time quantum algorithm A and polynomial-time computable function $m \leq \text{poly}(n)$ such that for all n, there exists an m-qubit advice state ρ_n such that $A(x, \rho_n)$ outputs $L(x)$ with probability at least 2/3 for all $x \in \{0,1\}^n$.

YQP The class YQP was first described in [3], but the definition was later corrected by Aaronson and Drucker [5]. Informally, it is the oblivious version of QMA ∩ coQMA, where "oblivious" means that the witness sent by the prover, Merlin, depends only on the length of the input. Oblivious proofs can also be thought of as a restriction of non-uniform classes, like BQP/qpoly, to advice that is verifiable, as in NP and QMA [13]. This combination in YQP been described as "untrusted advice" [3].

Definition 2.2. A language L is in YQP if there exists a polynomial-time uniform family of quantum circuits $\{Y_n\}_{n\in\mathbb{N}}$ that satisfy the following. Circuit Y_n is of size poly(n) and takes as input $x \in \{0,1\}^n$, an m-qubit state ρ for some $m \in \text{poly}(n)$, and an ancilla register initialized to the all-zero state, and has two designated "advice-testing" and "output" qubits. $Y_n(x, \rho)$ acts as follows:

1. First, Y_n applies a subcircuit A_n to all registers, after which the advice-testing qubit is measured, producing a value $b_{\text{adv}} \in \{0,1\}$.
2. Next, Y_n applies a second subcircuit B_n to all registers, then measures the output qubit, producing a value $b_{\text{out}} \in \{0,1\}$.

These output bits satisfy the following:

- For all n, there exists a ρ_n such that for all $x \in \{0,1\}^n$, the advice bit satisfies $\mathrm{E}[b_{\text{adv}}] \geq 9/10$.
- For any x, ρ such that $\mathrm{E}[b_{\text{adv}}] \geq 1/10$, on input x, ρ we have

$$\Pr[b_{\text{out}} = L(x) \mid b_{\text{adv}} = 1] \geq 9/10\,.$$

L is in the subclass YQP* if the family can be chosen such that b_{adv} is independent of x.

Note that in the above definition, the subcircuit B_n acts on the output of subcircuit A_n. It does not necessarily receive a clean copy of the input.

The classes YQP/poly and YQP*/poly are simply defined by removing the requirement that the circuit family $\{Y_n\}_{n\in\mathbb{N}}$ be uniform [5].

Just as Oblivious-NP is unlikely to contain NP [11], it also seems unlikely that QMA is contained in YQP. We have the trivial bounds BQP ⊆ YQP* ⊆ YQP ⊆ QMA and YQP ⊆ BQP/qpoly. Studying YQP may be motivated by the use of oblivious complexity classes in constructing circuit lower bounds [11, 8, 12], by the fact that BQP/qpoly = YQP*/poly = YQP/poly shown in [5], or by the results shown in this article.

APP The class APP was introduced by Lide Li [16] in pursuit of a large class of PP-low languages. We use the equivalent definition given by Fenner [9, Corollary 3.7].

Definition 2.3. $L \in \mathsf{APP}$ if and only if there exist functions $f, g \in \mathsf{GapP}$ and constants $0 \le \lambda < \upsilon \le 1$ such that for all n and $x \in \{0,1\}^n$, we have $g(1^n) > 0$ and

- If $x \in L$ then $\upsilon g(1^n) \le f(x) \le g(1^n)$;
- If $x \notin L$ then $0 \le f(x) \le \lambda g(1^n)$.

In the above definition, recall that GapP is the closure of #P under subtraction. In other words, while every function $f \in$ #P corresponds to a nondeterministic polynomial-time Turing Machine N such that $f(x)$ equals the number of accepting paths of $N(x)$, a GapP function equals the number of accepting paths minus the number of rejecting paths.

The class PP can be thought of as comparing a #P function to a threshold exactly, with no promise gap. The class in fact remains unchanged if it is defined as comparing a GapP function to a threshold, and the threshold may be as simple as one-half of the possible paths or as complex as a GapP function. In these terms, APP can be thought of as comparing a GapP function (here $f(x)$) to some threshold (here $g(1^n)$), where the complexity of the threshold is limited to a GapP function which may depend on the input size but not the input, and where there is some arbitrarily small but nonzero promise gap (from $\lambda g(1^n)$ to $\upsilon g(1^n)$). Comparing the two classes, APP is a subclass of PP and is PP-low, meaning $\mathsf{PP}^{\mathsf{APP}} = \mathsf{PP}$.

Like APP, the best upper bound on the well-known class $\mathsf{A_0PP} = \mathsf{SBQP}$ [15] is PP, so we make a brief comparison. $\mathsf{A_0PP}$ contains QMA [22] and so also contains YQP^*, and $\mathsf{A_0PP}$ is *not* known to be PP-low. In contrast, APP is not known to contain even NP and is PP-low. Both APP and $\mathsf{A_0PP}$ contain the class AWPP [9, 22]. However, neither APP or $\mathsf{A_0PP}$ is known to contain the other.

A useful fact We use the following fact shown for uniform circuit families by Watrous [23, Section IV.5], and shown earlier for QTMs by Fortnow and Rogers [10].

Lemma 2.4. *For any polynomial-time uniformly generated family of quantum circuits $\{Q_n\}_{n\in\mathbb{N}}$ each of size bounded by a polynomial $t(n)$, there is a GapP function f such that for all n-bit x,*

$$\Pr[Q_n(x) \text{ accepts}] = \frac{f(x)}{5^{t(n)}}.$$

Error reduction In our proof that $\mathsf{YQP}^* \subseteq \mathsf{APP}$, we perform error reduction on quantum circuits. Error reduction for complexity classes involving quantum inputs, such as QMA or BQP/qpoly, is often performed using many copies of the input in parallel, but we wish to use a particular state ρ as input, not $\rho^{\otimes k}$. Therefore, we require the "in-place" error reduction technique of Marriott and Watrous [18].

Theorem 2.5 (Theorem 3.3 of [18]). *Let $a, b : \mathbb{N} \to [0,1]$ and $q \in \mathrm{poly}(n)$ satisfy $a(n) - b(n) \ge 1/q(n)$. Consider any quantum circuit C of size s acting on an m-qubit input and an all-zero ancilla* *register such that C accepts with probability at least a or at most b for $s, m \leq \text{poly}(n)$. Then there exists a polynomial-time procedure that, for any $r \in \text{poly}(n)$, produces a circuit C' of size $\text{poly}(s)$ that also acts on an m-qubit input and accepts with probability at least $1 - 2^{-r}$ or at most 2^{-r}.*

Our proof requires analyzing not just the output probabilities of the amplified circuit, but also the state of the system at the end of the circuit. Fortunately, studying the proof of [18, Theorem 3.3] provides some straightforward observations. Here we briefly sketch part of the construction and then state the properties necessary for our proof.

Consider some quantum circuit C that takes an all-zero ancilla register $|0\dots0\rangle$ and some quantum state as input and that accepts or rejects with some probability. The error reduction algorithm of [18] involves applying the circuit C, measuring the output qubit and recording whether it is $|0\rangle$ or $|1\rangle$ in a variable y_{2i-1}, applying $C^\dagger$, measuring the circuit's ancilla register and recording whether it is in the all-zero state or not in a variable y_{2i}, and repeating these steps for some number of iterations M. Call the full, amplified circuit C'. At the end of C', the recorded bits $\{y_1, \dots, y_{2M}\}$ can be used to estimate the probability C accepts with high precision. Specifically, the more pairs such that $y_i = y_{i+1}$, the more likely that C' accepts.

The recorded bits also tell us about the state of the system after applying C'. If after applying $C^\dagger$, a bit $y_{2i} = 1$, then the state of the ancilla register was projected into the all-zero state. Now, suppose the circuit C' is applied to an m-qubit proof state, so there are 2^m eigenstates $\{|\lambda_i\rangle\}_{j\in[2^m]}$ of C'. Studying the proof of [18], if the initial state given to C' was an eigenstate $|\lambda_j\rangle$, and after a round of applying C and $C^\dagger$ the recorded bits were $y_{2i-1} = y_{2i} = 1$, then not only is the ancilla register known to be in the all-zero state, but the final state of the proof register is the same as its initial state, $|\lambda_j\rangle$.

A brief analysis allows us to characterize the probability of this outcome. Note C and C' have the same m-qubit eigenstates, and suppose an eigenstate $|\lambda_j\rangle$ is accepted by the original circuit C with probability p, Intuitively, consecutive bits are transitions which depend on whether we expect the circuit beginning with a particular proof and properly initialized all-zero ancilla register to produce an output qubit close to $|1\rangle$ or to $|0\rangle$, and vice-versa. In other words, when C' is run on $|\lambda_j\rangle$, we have $\Pr[y_i = 1 \mid y_{i-1} = 1] = p$ for all i. This is a two-state Markov chain with probability p of changing states. Raising the appropriate transition matrix to the i-th power and applying it to the initial state $y_0 = 1$ (C' begins with ancilla in the all-zero state) yields $\Pr[y_i = 1] = \left((2p-1)^i + 1\right)/2$ for all $i > 0$. If we make the additional assumption that $p > 1/2$, then this probability is greater than $1/2$. Then we can also conclude that $\Pr[y_{i+1} = y_i = 1] > p/2$ for all i. Moreover, as noted above, any pair $y_i = y_{i+1}$ only increases the probability the amplified circuit accepts, allowing us to calculate a lower bound on the joint probability.

Lemma 2.6 (Extension of Theorem 3.3 of [18])**.** *In addition to the statement of Theorem 2.5, the amplified circuit C' records two final variables $y, z \in \{0, 1\}$. Suppose $|\lambda\rangle$ is an eigenstate of C and that C' is run on $|\lambda\rangle$. If $y = z = 1$, then the system is left in the state $|\lambda\rangle|0\dots0\rangle$. If C accepts $|\lambda\rangle$ with probability at least a and $a > 1/2$, then the probability that C' accepts $|\lambda\rangle$ and the variables $y = z = 1$ is at least $(1 - 2^{-r})\, a/2$.*

3 Results

We first prove our main technical result, that $\mathsf{YQP}^* \subseteq \mathsf{APP}$. Our approach is as follows. APP evaluates the ratio of two GapP functions, where one of the functions is only allowed to depend on the input length. By Lemma 2.4, functions in GapP can encode the output probabilities of quantum circuits. So, for a YQP^* computation with circuit Y_n and subcircuit A_n, we run them both on a random proof using the maximally mixed state and ask APP to determine the ratio of their acceptance probabilities. This is possible because the acceptance probability of A_n over a random proof depends only on the input length. We perform error reduction on the subcircuit A_n so that A_n mistakenly accepting "bad" proofs has a negligible effect on the acceptance probability of Y_n. Thus, the ratio of probabilities approximates how often Y_n accepts given a good proof. Because we wish to use the maximally mixed state as input and parallel copies of a uniform mixture is not the same as a uniform mixture of parallel copies, we require the "in-place" error reduction technique reviewed above.

Lemma 3.1. $\mathsf{YQP}^* \subseteq \mathsf{APP}$.

Proof. Consider any language $L \in \mathsf{YQP}^*$. Let $\{Y_n, A_n, B_n\}_{n\in\mathbb{N}}$ be the associated family of circuits and subcircuits, in which Y_n takes string x and a supposed proof or advice state as input, in which subcircuit A_n validates the proof and produces output bit b_{adv}, and in which, given A_n accepted, B_n uses the proof to verify whether the particular input x is in L, producing the output bit b_{out}. Note that because we consider YQP^*, the circuit A_n only takes the proof state, not x, as input. Let m be a polynomial in n denoting the size of the proof register.

Apply the in-place error reduction of [18] stated in Theorem 2.5 on the circuits A_n with a polynomial q in n of our choosing to produce a new circuit family $\{A'_n\}_{n\in\mathbb{N}}$ such that for any proof ρ,

- $\Pr[A_n(\rho)] \geq \frac{9}{10} \Rightarrow \Pr[A'_n(\rho)] \geq 1 - 2^{-q}$;
- $\Pr[A_n(\rho)] \leq \frac{1}{10} \Rightarrow \Pr[A'_n(\rho)] \leq 2^{-q}$.

For later use, we choose $q \geq \max\{2m, 10\}$. Note that A'_n also produces the two variables y and z described by Lemma 2.6.

We define $\{A''_n\}_{n\in\mathbb{N}}$ to be the amplified circuits $\{A'_n\}_{n\in\mathbb{N}}$ with the additional rule that the circuit accepts iff both $b_{\text{adv}} = 1$ and the two recorded bits $y = z = 1$. Further, define $\{A'''_n\}_{n\in\mathbb{N}}$ so that $A'''_n = A''_n(\frac{\mathbb{I}}{2^m})$, with the maximally mixed state hard-wired into the proof register. Similarly, we define $\{Y'_n\}_{n\in\mathbb{N}}$ to apply the amplified subcircuit A'_n and B_n, we define $\{Y''_n\}_{n\in\mathbb{N}}$ to apply A''_n and B_n and thus accept iff $b_{\text{adv}}, b_{\text{out}}, y, z$ all equal 1, and we define $\{Y'''_n\}_{n\in\mathbb{N}}$ so that $Y'''_n(x) = Y''_n(x, \frac{\mathbb{I}}{2^m})$ with the maximally mixed state hard-wired into the proof register, meaning that it uses A'''_n as a subcircuit.

Analysis Applying Lemma 2.4, there exist GapP functions f, g and polynomials r, t such that for all n-bit x,

$$\Pr\left[A'''_n \text{ accepts}\right] = \frac{f(1^n)}{5^{r(n)}} \quad \text{and} \quad \Pr\left[Y'''_n(x) \text{ accepts}\right] = \frac{g(x)}{5^{t(n)}}.$$

The function f depends only on the input length n, not x, because the circuit A_n''' is independent of x. Next, we define $F(1^n) = f(1^n)5^{t(n)-r(n)}$, which is a GapP function since $5^{t(n)-r(n)} \in \mathsf{FP} \subseteq \mathsf{GapP}$ and GapP is closed under multiplication. Given the definition of YQP* guarantees there exists a "good" proof for circuit A_n, we have $f(1^n), F(1^n) > 0$. Combining these definitions,

$$\frac{g(x)}{F(1^n)} = \frac{\Pr\left[Y_n'''(x) \text{ accepts}\right]}{\Pr\left[A_n''' \text{ accepts}\right]} .$$

We will show bounds on the ratio $g(x)/F(1^n)$ based on whether x is in L or not in L in order to prove L is in APP. First, note that the ratio is at most 1 since Y_n''' only accepts if the subcircuit A_n''' accepts, and it is at least 0 since probabilities are non-negative. Next, let $\{|\lambda_i\rangle\}_{i\in[2^m]}$ be the set of eigenvectors of the circuit A_n. By writing the maximally mixed state, which is hard-wired into the proof register of Y_n''', in terms of this eigenbasis, we find

$$\frac{\Pr\left[Y_n'''(x) \text{ accepts}\right]}{\Pr\left[A_n''' \text{ accepts}\right]} = \frac{\Pr\left[Y_n''(x, \frac{\mathbb{I}}{2^m}) \text{ accepts}\right]}{\Pr\left[A_n''(\frac{\mathbb{I}}{2^m}) \text{ accepts}\right]} = \frac{\sum_{i=1}^{2^m} \Pr\left[Y_n''(x, |\lambda_i\rangle) \text{ accepts}\right]}{\sum_{i=1}^{2^m} \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right]} .$$

Next, use the fact that Y_n'' accepting requires that A_n'' accepts to find the above equals

$$\frac{\sum_{i=1}^{2^m} \Pr\left[Y_n''(x, |\lambda_i\rangle) \text{ accepts} \mid A_n''(|\lambda_i\rangle) \text{ accepts}\right] \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right]}{\sum_{i=1}^{2^m} \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right]} .$$

The observation from Lemma 2.6 implies that if the amplified circuit A_n'' accepts, then the state sent on to the subcircuit B_n within Y_n'' is the initial eigenstate $|\lambda_i\rangle$. Then, the above equals

$$\frac{\sum_{i=1}^{2^m} \Pr\left[B_n(x, |\lambda_i\rangle) \text{ accepts}\right] \cdot \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right]}{\sum_{i=1}^{2^m} \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right]} .$$

Define

$$\mathcal{B} = \{i \in [2^m] \mid \Pr\left[A_n(|\lambda_i\rangle)\right] \leq 0.1\} ,$$

which are intuitively the "bad" proofs, such that states in $\mathcal{B}$ will be rejected by A_n' with high probability while the "not bad" states in $\overline{\mathcal{B}}$ cause B_n to output the correct answer with high probability. We can now rewrite the numerator in the above ratio as

$$\sum_{i\in\mathcal{B}} \Pr\left[B_n(x, |\lambda_i\rangle) \text{ accepts}\right] \cdot \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right] + \sum_{i\in\overline{\mathcal{B}}} \Pr\left[B_n(x, |\lambda_i\rangle) \text{ accepts}\right] \cdot \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right]$$

and rewrite the denominator as

$$\sum_{i\in\mathcal{B}} \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right] + \sum_{i\in\overline{\mathcal{B}}} \Pr\left[A_n''(|\lambda_i\rangle) \text{ accepts}\right] .$$

We will use this expression for $g(x)/F(1^n)$ as the starting point in our analysis of the YES and NO cases. Additionally, let $|\lambda^*\rangle$ denote a proof in $\overline{\mathcal{B}}$, which is guaranteed to be nonempty by the definition of YQP^*. By the observation from Lemma 2.6, $\Pr\left[A''_n(|\lambda^*\rangle)\text{ accepts}\right] \geq (1-2^{-q})\,(0.9)(0.5)$.

Suppose we have a YES instance with $x \in L$. Then, we may calculate that $g(x)/F(1^n)$ is at least

$$\begin{aligned}\frac{\sum_{i\in\mathcal{B}} 0 + \sum_{i\in\overline{\mathcal{B}}} \frac{9}{10}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]}{\sum_{i\in\mathcal{B}} 2^{-q} + \sum_{i\in\overline{\mathcal{B}}}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]} &= \frac{\sum_{i\in\overline{\mathcal{B}}} \frac{9}{10}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]}{|\mathcal{B}|\,2^{-q} + \sum_{i\in\overline{\mathcal{B}}}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]}\\ &\geq \frac{\frac{9}{10}\Pr\left[A''_n(|\lambda^*\rangle)\text{ accepts}\right]}{|\mathcal{B}|\,2^{-q} + \Pr\left[A''_n(|\lambda^*\rangle)\text{ accepts}\right]}\,,\end{aligned}$$

where the second line follows by the fact that $x/(c+x)$ decreases as x decreases. Applying this fact again along with our choice $q \geq \max\{2m, 10\}$, we find the above is at least

$$\begin{aligned}\frac{\frac{9}{10}(1-2^{-q})(0.9)(0.5)}{|\mathcal{B}|\,2^{-q} + (1-2^{-q})(0.9)(0.5)} &\geq \frac{0.405(1-2^{-q})}{2^{m-q} + 0.45(1-2^{-q})}\\ &\geq \frac{0.405(1-2^{-q})}{2^{q/2-q} + 0.45(1-2^{-q})} > 0.84\,.\end{aligned}$$

On the other hand, consider a NO-instance. By similar steps as in the YES-case, we have that $g(x)/F(1^n)$ is at most

$$\begin{aligned}\frac{|\mathcal{B}|\,2^{-q} + \sum_{i\in\overline{\mathcal{B}}} \frac{1}{10}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]}{\sum_{i\in\mathcal{B}} 0 + \sum_{i\in\overline{\mathcal{B}}}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]} &\leq \frac{2^{m-q}}{\sum_{i\in\overline{\mathcal{B}}}\Pr\left[A''_n(|\lambda_i\rangle)\text{ accepts}\right]} + \frac{1}{10}\\ &\leq \frac{2^{-q/2}}{\Pr\left[A''_n(|\lambda^*\rangle)\text{ accepts}\right]} + \frac{1}{10}\\ &\leq \frac{2^{-q/2}}{(1-2^{-q})\,(0.9)(0.5)} + \frac{1}{10} < 0.2\,.\end{aligned}$$

We have shown a constant separation of $g(x)/F(1^n)$ in YES- and NO-instances. This satisfies the criteria of APP in Definition 2.3, so we conclude $\mathsf{YQP}^* \subseteq \mathsf{APP}$. □

Next, the fact APP is known to be PP-low [16, Theorem 6.4.14] gives us the following corollary.

Corollary 3.2. YQP^* *is* PP*-low, i. e.,* $\mathsf{PP}^{\mathsf{YQP}^*} = \mathsf{PP}$.

For intuition, an alternative proof of Corollary 3.2 without Lemma 3.1 might have relied on the equality $\mathsf{PP} = \mathsf{postBQP}$ [1], where $\mathsf{postBQP}$ has the ability to post-select, i. e., it is guaranteed to output the correct answer with high probability *conditioned on* some other event which may occur with very small probability. So, instead of $\mathsf{PP}^{\mathsf{YQP}^*}$, we might have considered

$\mathsf{postBQP}^{\mathsf{YQP}^*}$. Whenever the postBQP machine would make a query, it instead could run the YQP^* proof-validation circuit on the maximally mixed state, post-select on it accepting, then simulate the rest of the YQP^* computation.

We are now able to give a corrected proof of the result originally claimed for BQP/qpoly but only proved for BQP/poly by Aaronson [2]. We mostly repeat Aaronson's proof, but substitute YQP^* where he relied on QMA.

Theorem 3.3. *If* $\mathsf{PP} \subseteq \mathsf{BQP/qpoly}$, *then the Counting Hierarchy collapses to* $\mathsf{CH} = \mathsf{QMA} = \mathsf{YQP}^*$.

Proof. Suppose $\mathsf{PP} \subseteq \mathsf{BQP/qpoly}$. Clearly then P^{PP} is also contained in BQP/qpoly. So, the #P-complete problem PERMANENT is contained in BQP/qpoly, since $\#\mathsf{P} \subseteq \mathsf{P}^{\#\mathsf{P}} = \mathsf{P}^{\mathsf{PP}}$. From [5], we know that $\mathsf{BQP/qpoly} = \mathsf{YQP}^*\mathsf{/poly}$. A $\mathsf{YQP}^*\mathsf{/poly}$ protocol involves a circuit, a trusted advice string, and an untrusted quantum proof. Let C and a be the circuit and advice string for solving PERMANENT in $\mathsf{YQP}^*\mathsf{/poly}$.

Next in YQP^*, without trusted advice, Arthur can request that an untrusted Merlin send many copies of the resources from the above $\mathsf{YQP}^*\mathsf{/poly}$ protocol, including the quantum proof and a supposed copy of the circuit C and advice a. To check these *untrusted* copies of the proof, circuit, and advice, Arthur generates a random set of inputs and simulates the interactive protocol for PERMANENT due to [17] using the copies in place of the prover. If the protocol accepts (meaning the "prover" worked), then with high probability, Merlin must have sent resources that work on a large fraction of inputs. This ends the proof-verification phase of YQP^*.

Given a circuit and advice that work on most inputs, Arthur can use the random self-reducibility of PERMANENT to generate a circuit C' that is correct on *all* inputs with high probability (see e. g., [7, Sec. 8.6.2]). Thus, after the verification phase, the YQP^* protocol can simulate #P. By the same argument as above, this also means it can simulate $\mathsf{P}^{\#\mathsf{P}} \supseteq \mathsf{PP}$, and we have $\mathsf{PP} = \mathsf{YQP}^*$.

In this way, any level of the Counting Hierarchy $\mathsf{C_iP} = (\mathsf{C_{i-1}P})^{\mathsf{PP}}$ with $i > 1$ is reducible to $(\mathsf{C_{i-1}P})^{\mathsf{YQP}^*}$ which by Corollary 3.2 equals $\mathsf{C_{i-1}P}$. This works recursively for all levels, collapsing $\mathsf{C_iP}$ to $\mathsf{C_1P} = \mathsf{PP}$, so that all of $\mathsf{CH} = \mathsf{PP} = \mathsf{YQP}^*$. ☐

An alert reader may notice the proof never directly asks the base PP machine to guess advice and hard-code it into a query, as referenced in the introduction. That step was explicit in the original proof in [2], to reduce $\mathsf{PP}^{\mathsf{BQP/poly}}$ to $\mathsf{PP}^{\mathsf{BQP}} = \mathsf{PP}$. Here, unlike BQP, the class YQP^* is itself able to guess a proof, so we reduce $\mathsf{PP}^{\mathsf{YQP}^*\mathsf{/poly}}$ to $\mathsf{PP}^{\mathsf{YQP}^*}$ without "asking" anything of the base computation until collapsing $\mathsf{PP}^{\mathsf{YQP}^*}$ to PP.

Given the above result, we can also fully recover the following result originally claimed by Aaronson [2], giving an improved unconditional upper bound on fixed-polynomial-size quantum circuits with quantum advice. For completeness, we repeat the proof.

Theorem 3.4. PP *does not have quantum circuits of size n^k for any fixed k. Furthermore, this holds even if the circuits can use quantum advice.*

Proof. Suppose PP does have circuits of size n^k. This implies $\mathsf{PP} \subseteq \mathsf{BQP/qpoly}$, which by Theorem 3.3 implies $\mathsf{CH} = \mathsf{YQP}^*$, which includes $\mathsf{P}^{\mathsf{PP}} = \mathsf{PP} = \mathsf{YQP}^*$. Together, there are circuits of size n^k for P^{PP}, which contradicts the result [2, Theorem 4] (unaffected by the bug) that P^{PP} does not have such circuits even with quantum advice. □

In fact, [2] noted that the proof showing P^{PP} does not have circuits of size n^k for fixed k even with quantum advice can be strengthened. Substituting this stronger result into the above proof, we have that Theorem 3.4 can be strengthened to show for all functions $f(n) \leq 2^n$, the class PTIME(f(f(n))), which is like PP but for machines of running time $f(f(n))$, requires quantum circuits using quantum advice of size at least $f(n)/n^2$. In particular, this implies PEXP, the exponential-time version of PP, requires quantum circuits with quantum advice of "half-exponential" size (meaning a function that becomes exponential when composed with itself [19]).

References

[1] SCOTT AARONSON: Quantum computing, postselection, and probabilistic polynomial-time. *Proc. Royal Soc. A*, 461(2063):3473–3482, 2005. [doi:10.1098/rspa.2005.1546] 9

[2] SCOTT AARONSON: Oracles are subtle but not malicious. In *Proc. 21st IEEE Conf. on Comput. Complexity (CCC'06)*, pp. 340–354. IEEE Comp. Soc., 2006. [doi:10.1109/CCC.2006.32] 2, 3, 10, 11

[3] SCOTT AARONSON: The learnability of quantum states. *Proc. Royal Soc. A*, 463(2088):3089–3114, 2007. [doi:10.1098/rspa.2007.0113] 4

[4] SCOTT AARONSON: Yet more errors in papers, May 2017. Available at `https://scottaaronson.blog/?p=3256`. 2

[5] SCOTT AARONSON AND ANDREW DRUCKER: A full characterization of quantum advice. *SIAM J. Comput.*, 43(3):1131–1183, 2014. [doi:10.1137/110856939] 3, 4, 10

[6] AVANTIKA AGARWAL, SEVAG GHARIBIAN, VENKATA KOPPULA, AND DORIAN RUDOLPH: Quantum polynomial hierarchies: Karp–Lipton, error reduction, and lower bounds. In *Proc. Internat. Symp. Math. Foundations of Comp. Sci. (MFCS'24)*, pp. 7:1–17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2024. [doi:10.4230/LIPIcs.MFCS.2024.7] 3

[7] SANJEEV ARORA AND BOAZ BARAK: *Computational Complexity: A Modern Approach*. Cambridge Univ. Press, 2009. [doi:10.1017/CBO9780511804090] 3, 10

[8] VENKATESAN T. CHAKARAVARTHY AND SAMBUDDHA ROY: Oblivious symmetric alternation. In *Proc. 23rd Symp. Theoret. Aspects of Comp. Sci. (STACS'06)*, pp. 230–241. Springer, 2006. [doi:10.1007/11672142_18] 4

[9] STEPHEN A. FENNER: PP-lowness and a simple definition of AWPP. *Theory Computing Sys.*, 36(2):199–212, 2003. [doi:10.1007/s00224-002-1089-8] 5

[10] LANCE FORTNOW AND JOHN ROGERS: Complexity limitations on quantum computation. *J. Comput. System Sci.*, 59(2):240–252, 1999. [doi:10.1006/jcss.1999.1651] 3, 5

[11] LANCE FORTNOW, RAHUL SANTHANAM, AND RYAN WILLIAMS: Fixed-polynomial size circuit bounds. In *Proc. 24th IEEE Conf. on Comput. Complexity (CCC'09)*, pp. 19–26. IEEE Comp. Soc., 2009. [doi:10.1109/CCC.2009.21] 4

[12] KARTHIK GAJULAPALLI, ZEYONG LI, AND ILYA VOLKOVICH: Oblivious complexity classes revisited: Lower bounds and hierarchies. In *Proc. 44th Found. Softw. Techn. Theoret. Comp. Sci. Conf. (FSTTCS'24)*, pp. 23:1–19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2024. [doi:10.4230/LIPIcs.FSTTCS.2024.23] 4

[13] ODED GOLDREICH AND OR MEIR: Input-oblivious proof systems and a uniform complexity perspective on P/poly. *ACM Trans. Comput. Theory*, 7(4):1–13, 2015. [doi:10.1145/2799645] 4

[14] RICHARD M. KARP AND RICHARD J. LIPTON: Some connections between nonuniform and uniform complexity classes. In *Proc. 12th STOC*, pp. 302–309. ACM Press, 1980. [doi:10.1145/800141.804678] 1

[15] GREG KUPERBERG: How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015. [doi:10.4086/toc.2015.v011a006] 5

[16] LIDE LI: *On the counting functions*. Ph. D. thesis, The University of Chicago, 1993. Available at https://www.proquest.com/dissertations-theses/on-counting-functions/docview/304080357/se-2. 2, 3, 5, 9

[17] CARSTEN LUND, LANCE FORTNOW, HOWARD KARLOFF, AND NOAM NISAN: Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. [doi:10.1145/146585.146605] 10

[18] CHRIS MARRIOTT AND JOHN WATROUS: Quantum Arthur–Merlin games. *Comput. Complexity*, 14(2):122–152, 2005. [doi:10.1007/s00037-005-0194-x] 5, 6, 7

[19] PETER BRO MILTERSEN, N. V. VINODCHANDRAN, AND OSAMU WATANABE: Super-polynomial versus half-exponential circuit size in the Exponential Hierarchy. In *Proc. 5th Internat. Combinatorics and Computing Conf. (COCOON'99)*, pp. 210–220. Springer, 1999. [doi:10.1007/3-540-48686-0_21] 11

[20] TOMOYUKI MORIMAE AND HARUMICHI NISHIMURA: Quantum interpretations of AWPP and APP. *Quantum Info. Comput.*, 16(5–6):498–514, 2016. [doi:10.26421/QIC16.5-6-6] 3

[21] N. V. VINODCHANDRAN: A note on the circuit complexity of PP. *Theoret. Comput. Sci.*, 347(1–2):415–418, 2005. [doi:10.1016/j.tcs.2005.07.032] 2

[22] MIKHAIL N. VYALI: QMA = PP implies that PP contains PH. *Electron. Colloq. Comput. Complexity*, TR03-021, 2003. [ECCC] 5

[23] JOHN WATROUS: Quantum computational complexity, 2008. See also the Encyclopedia of Complexity and Systems Science. [arXiv:0804.3401] 3, 5

[24] RYAN WILLIAMS: Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1, art. 2):1–32, 2014. [doi:10.1145/2559903] 2

[25] Complexity Zoo: BQP/poly, BQP/mpoly, BQP/qpoly, BQP. https://complexityzoo.net/Complexity_Zoo:B. Accessed 13 Mar. 2024. 3

AUTHOR

Justin Yirka
Quantum Computing Researcher
Blanqet
Austin, TX, USA
justinyirka@gmail.com
https://justinyirka.com/

ABOUT THE AUTHOR

JUSTIN YIRKA graduated from The University of Texas at Austin in May 2025, advised by Scott Aaronson. The subject of his dissertation was quantum computational complexity, including the results in this article. He is now a quantum computing researcher for Blanqet, a startup founded by professors from the University of Chicago and where his Ph. D. adviser, Scott Aaronson, is a partner. Blanqet's office is in Chicago and Justin works remotely from Austin. Justin earned his B. S. from Virginia Commonwealth University, where he was introduced to research by working with Sevag Gharibian. Justin has also worked at Sandia National Laboratories, at Los Alamos National Laboratory, as an Uber driver, and as a lifeguard.