Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 58 (2018)

Amplification with One NP Oracle Query

Thomas Watson[*]

November 26, 2021

Abstract

We provide a complete picture of the extent to which amplification of success probability is possible for randomized algorithms having access to one NP oracle query, in the settings of two-sided, one-sided, and zero-sided error. We generalize this picture to amplifying one-query algorithms with q-query algorithms, and we show our inclusions are tight for relativizing techniques.

1 Introduction

Amplification of the success probability of randomized *algorithms* is a ubiquitous tool in complexity theory. We investigate amplification for randomized *reductions* to NP-complete problems, which can be modeled as randomized algorithms with the ability to make queries to an NP oracle. The usual amplification strategy involves running multiple independent trials, which would also increase the number of NP oracle queries, so this does not generally work if we restrict the number of queries. We study, and essentially completely answer, the following question:

> *If a language is solvable with success probability p by a randomized polynomial-time algorithm with access to one* NP *oracle query, what is the highest success probability achievable with one query (or $q > 1$ many queries) to an* NP *oracle?*

The question makes sense for two-sided error ($\mathsf{BPP}^{\mathsf{NP}[1]}$), one-sided error ($\mathsf{RP}^{\mathsf{NP}[1]}$), and zero-sided error ($\mathsf{ZPP}^{\mathsf{NP}[1]}$), and it was mentioned in [CC06] as "an interesting problem worthy of further investigation." Partial results for zero-sided error were shown in [CP08]. The question is also relevant to the extensive literature on bounded NP queries (the boolean hierarchy); e.g., $\mathsf{ZPP}^{\mathsf{NP}[1]}$ shows up frequently in the context of the "two queries problem" [Tri10], which was the main application area of the results from [CP08]. A complementary question (about lowering the success probability in exchange for fewer NP queries) was studied in [Roh95].

Our first contribution characterizes the best amplification achievable by relativizing techniques in the two-sided error setting. In general, the best strategy for amplifying plain randomized algorithms is to take the majority vote of q independent trials, which in our setting would naively involve q NP oracle queries. One may suspect this majority vote strategy is optimal for us. We show this intuition is a red herring; it is possible to do better by "combining" NP oracle queries across different trials. As an extreme example, consider the special case of randomized *mapping* reductions to NP problems. These are equivalent to Arthur–Merlin games (AM), for which amplification is possible by running independent trials and simply having Merlin's message consist of certificates for a majority of the trials. However, if we allow one NP oracle query, but do not

[*] Department of Computer Science, University of Memphis.

ISSN 1433-8092

necessarily output the same bit the oracle returns, then combining queries is less straightforward, and it turns out amplification is only possible to a limited extent.

Our main take-home message is that starting with success probability greater than $\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{k+1}$, where k is an integer, we can get arbitrarily close to $\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{k}$ success probability while still using one NP query; using q nonadaptive queries, roughly a factor q improvement over this is possible.

We give precise definitions in § 2, but we now clarify our notation before stating the theorem. For $\epsilon \in (0,1]$ (the *advantage*), $\mathsf{BPP}_{\epsilon}^{\mathsf{NP}[1]}$ is the set of all languages solvable by a randomized polynomial-time algorithm that may make one query to an NP oracle and produces the correct output with probability $\geq \frac{1}{2}+\frac{1}{2}\epsilon$ on each input. For convenience, we define $\mathsf{BPP}_{>\epsilon}^{\mathsf{NP}[1]}$ by requiring that for some constant c there exists such an algorithm with advantage $\geq \epsilon + n^{-c}$, and we define $\mathsf{BPP}_{\epsilon>}^{\mathsf{NP}[1]}$ by requiring that for every constant d there exists such an algorithm with advantage $\geq \epsilon - 2^{-n^d}$; the reason for these conventions is just that they naturally arise in the proofs (e.g., standard majority amplification implies $\mathsf{BPP}_{>0} = \mathsf{BPP}_{1>}$). We make similar definitions for $\mathsf{BPP}^{\mathsf{NP}\|[q]}$ but allowing q nonadaptive NP oracle queries. Allowing q adaptive NP queries is equivalent to allowing $2^q - 1$ nonadaptive NP queries [Bei91].

Theorem 1 (Two-sided error). *For integers $1 \leq q \leq k$:*

- *If q is odd:* $\mathsf{BPP}_{>1/(k+1)}^{\mathsf{NP}[1]} \subseteq \mathsf{BPP}_{q/k>}^{\mathsf{NP}\|[q]}$ *and* $\mathsf{BPP}_{1/k}^{\mathsf{NP}[1]} \not\subseteq \mathsf{BPP}_{>q/k}^{\mathsf{NP}\|[q]}$ *relative to an oracle.*
- *If q, k are even:* $\mathsf{BPP}_{>1/(k+1)}^{\mathsf{NP}[1]} \subseteq \mathsf{BPP}_{q/k>}^{\mathsf{NP}\|[q]}$ *and* $\mathsf{BPP}_{1/(k-1)}^{\mathsf{NP}[1]} \not\subseteq \mathsf{BPP}_{>q/k}^{\mathsf{NP}\|[q]}$ *relative to an oracle.*

The word "oracle" has two meanings here. Besides the bounded NP oracle queries of central interest, "relative to an oracle" means there exists a language such that the separation holds when all computations (the randomized algorithm and the NP verifier) can make polynomially many adaptive queries to an oracle for that language. In particular, in the context of our relativized separations, randomized algorithms have access to two oracles. The separations in Theorem 1 are tight since the inclusions relativize. This implies that using "black-box simulation" techniques, it is not possible to significantly improve any of our inclusions.

If we start with advantage $> \frac{1}{k+1}$ where k is an integer, Theorem 1 tells us the best advantage achievable with q nonadaptive NP queries using relativizing techniques: if k is even we can amplify to essentially $\frac{q}{k}$; if k is odd we can amplify to essentially $\frac{q}{k}$ if q is odd, and $\frac{q}{k+1}$ if q is even. (Theorem 1 does not explicitly mention the case where q is even and k is odd, but in this case the best inclusion and separation are obtained by applying the theorem to the even integer $k+1$.)

A subtle issue is whether "$q/k>$" in the inclusion subscripts can be improved to "q/k"; e.g., it remains open to show that $\mathsf{BPP}_{>1/3}^{\mathsf{NP}[1]} \subseteq \mathsf{BPP}_{1/2}^{\mathsf{NP}[1]}$ or that $\mathsf{BPP}_{>1/3}^{\mathsf{NP}[1]} \not\subseteq \mathsf{BPP}_{1/2}^{\mathsf{NP}[1]}$ relative to an oracle.

The proof of Theorem 1 appears in § 3. No such nontrivial inclusion was known before; for relativized separations, the case $q = 1$, $k = 2$ was shown in [Wat20].

One-sided error algorithms must always output 0 if the answer is 0, and must output 1 with probability at least some $\epsilon \in (0,1]$ if the answer is 1. We define the advantage (the subscript of $\mathsf{RP}^{\mathsf{NP}\|[q]}$) to be this ϵ. In contrast to $\mathsf{BPP}_{\epsilon}^{\mathsf{NP}\|[q]}$ (where the advantage ϵ measures how much better than $\frac{1}{2}$ the success probability is), for $\mathsf{RP}_{\epsilon}^{\mathsf{NP}\|[q]}$ the advantage ϵ measures how much better than 0 the success probability is.

Theorem 2 (One-sided error).

- $\mathsf{RP}_{>1/2}^{\mathsf{NP}[1]} \subseteq \mathsf{RP}_{1>}^{\mathsf{NP}[1]}$.
- $\mathsf{RP}_{>0}^{\mathsf{NP}[1]} \subseteq \mathsf{RP}_{1/2}^{\mathsf{NP}[1]} \cap \mathsf{RP}_{1>}^{\mathsf{NP}\|[2]}$ *and* $\mathsf{RP}_{1/2}^{\mathsf{NP}[1]} \not\subseteq \mathsf{RP}_{>1/2}^{\mathsf{NP}[1]}$ *relative to an oracle.*

The proof of Theorem 2 appears in § 4 and is relatively straightforward (and could serve as a warm-up for Theorem 1 if the reader would like that). The inclusion $\mathsf{RP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{1/2}$ (which is stronger than $\mathsf{RP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{1/2>}$) uses a trick described in [CP08] for getting a tiny boost in the advantage.

Zero-sided error algorithms must output the correct bit with probability at least some $\epsilon \in (0, 1]$ and output $\perp$ (plead ignorance) with the remaining probability. We define the advantage (the subscript of $\mathsf{ZPP}^{\mathsf{NP}\|[q]}$) to be this ϵ.

[CP08] proved that $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{1/4}$ and $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/2} \subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{1>}$,[1] and left it unresolved what happens between advantages $\frac{1}{4}$ and $\frac{1}{2}$. We settle this decade-old open problem: amplification is possible between $\frac{1}{4}$ and $\frac{1}{3}$ and between $\frac{1}{3}$ and $\frac{1}{2}$.

Theorem 3 (Zero-sided error). *For integers $1 \leq q \leq k \leq 4$:*

- *If $k = 4$:* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$.
- *If $k \leq 3$:* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$.
- *If $q = 1$:* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/k} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{>q/k}$ *relative to an oracle.*

Moreover, the "$q/k>$" in the inclusion subscripts can be improved to "q/k" if $q < k$ and $k \geq 3$.

The proof of Theorem 3 appears in § 5. The "moreover" part uses the trick from [CP08] for a tiny boost in the advantage. Like the situation with $\mathsf{BPP}^{\mathsf{NP}[1]}$, it remains open to show that $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/3} \subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{1/2}$ or that $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/3} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{1/2}$ relative to an oracle. There is no reason to consider $k > 4$ in Theorem 3, since then $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/4>}$.

We conjecture that the third bullet in Theorem 3 also holds for $q > 1$ (i.e., the relativized separations $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/4} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[2]}_{>2/4}$ and $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/4} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[3]}_{>3/4}$ and $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/3} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[2]}_{>2/3}$). This remains open, though we are aware of how to prove that $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/4} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[2]}_{>3/4}$. Anyway, $q = 1$ is the most natural case, and we provide a complete proof for it.

Finally, we point out that none of the inclusions in this paper can be strengthened to yield advantage exactly 1 via relativizing techniques, since $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/2}$ relativizes [CC06] but $\mathsf{BPP} \not\subseteq \mathsf{P}^{\mathsf{NP}}$ relative to an oracle [Sto85].

2 Definitions

We formally define the relevant complexity classes in § 2.1 and their decision tree analogues (which are used for relativized separations) in § 2.2.

2.1 Time complexity

We think of a randomized algorithm M as taking a uniformly random string $s \in \{0, 1\}^r$ (for some number of coins r that depends on the input length); we let $M_s(x)$ denote M running on input x with outcome s.

[1] [Wat20] gave an alternative proof of the latter but with only $1 - \frac{1}{\text{poly}}$, rather than $1 - \frac{1}{\text{exp}}$, success probability.

For $\epsilon \in (0,1]$ (the *advantage*) and integer $q \geq 1$, language L is in $\mathsf{BPP}_\epsilon^{\mathsf{NP}\|[q]}$ iff there is a polynomial-time randomized algorithm M (taking input x and coin tosses $s \in \{0,1\}^r$) and a language $L' \in \mathsf{NP}$ such that the following hold.

Syntax: The computation of $M_s(x)$ produces a tuple of query strings $(z_1, \ldots, z_q)$ and a truth table *out*: $\{0,1\}^q \to \{0,1\}$; the output is then *out*$(L'(z_1), \ldots, L'(z_q))$.
Correctness: The output is $L(x)$ with probability $\geq \frac{1}{2} + \frac{1}{2}\epsilon$.

$\mathsf{RP}_\epsilon^{\mathsf{NP}\|[q]}$ is defined similarly except for correctness, we require the output is always 0 if $L(x) = 0$, and is 1 with probability $\geq \epsilon$ if $L(x) = 1$. $\mathsf{ZPP}_\epsilon^{\mathsf{NP}\|[q]}$ is defined similarly except *out*: $\{0,1\}^q \to \{0,1,\perp\}$ and for correctness, we require the output is always $L(x)$ or $\perp$, and is $L(x)$ with probability $\geq \epsilon$.

For $\mathcal{C} \in \{\mathsf{BPP}^{\mathsf{NP}\|[q]}, \mathsf{RP}^{\mathsf{NP}\|[q]}, \mathsf{ZPP}^{\mathsf{NP}\|[q]}\}$, we define

$$\mathcal{C}_{>\epsilon} = \bigcup_{\text{constants } c} \mathcal{C}_{\epsilon + n^{-c}} \quad \text{and} \quad \mathcal{C}_{\epsilon>} = \bigcap_{\text{constants } d} \mathcal{C}_{\epsilon - 2^{-n^d}}.$$

When $q = 1$ we may drop the $\|$ from the superscripts.

2.2 Decision tree complexity

We think of a randomized decision tree T as the uniform distribution over a multiset of corresponding deterministic decision trees T_s indexed by $s \in \{0,1\}^r$; we denote this as $T \sim \{T_s : s \in \{0,1\}^r\}$. In this setting, "query" actually has two meanings for us: a decision tree makes queries to individual input bits, then it forms an NP-type (DNF) oracle query.

We define a $\mathsf{BPP}_\epsilon^{\mathsf{NP}\|[q]}$-type decision tree T for $f\colon \{0,1\}^n \to \{0,1\}$ on input x as follows.

Syntax: $T \sim \{T_s : s \in \{0,1\}^r\}$ where each T_s makes queries to the bits of x until it reaches a leaf, which is labeled with a tuple of DNFs $(\varphi_1, \ldots, \varphi_q)$ and a function *out*: $\{0,1\}^q \to \{0,1\}$; the output is then *out*$(\varphi_1(x), \ldots, \varphi_q(x))$.
Correctness: The output is $f(x)$ with probability $\geq \frac{1}{2} + \frac{1}{2}\epsilon$.
Cost: The maximum height of any T_s, plus the maximum width (maximum number of literals in any term) of any DNF appearing at a leaf.

An $\mathsf{RP}_\epsilon^{\mathsf{NP}\|[q]}$-type decision tree is defined similarly except for correctness we require the output is always 0 if $f(x) = 0$, and is 1 with probability $\geq \epsilon$ if $f(x) = 1$. A $\mathsf{ZPP}_\epsilon^{\mathsf{NP}\|[q]}$-type decision tree is defined similarly except *out*: $\{0,1\}^q \to \{0,1,\perp\}$ and for correctness, we require the output is always $f(x)$ or $\perp$, and is $f(x)$ with probability $\geq \epsilon$.

We follow the convention of overloading complexity class names as decision tree complexity measures: for $\mathcal{C} \in \{\mathsf{BPP}^{\mathsf{NP}\|[q]}, \mathsf{RP}^{\mathsf{NP}\|[q]}, \mathsf{ZPP}^{\mathsf{NP}\|[q]}\}$, $\mathcal{C}_\epsilon^{\mathsf{dt}}(f)$ denotes the minimum cost of any $\mathcal{C}_\epsilon$-type decision tree for a partial function f, and $\mathcal{C}_\epsilon^{\mathsf{dt}}$ also denotes the class of all families of f's with $\mathcal{C}_\epsilon^{\mathsf{dt}}(f) \leq \operatorname{polylog}(n)$, and we define

$$\mathcal{C}_{>\epsilon}^{\mathsf{dt}} = \bigcup_{\text{constants } c} \mathcal{C}_{\epsilon + \log^{-c} n}^{\mathsf{dt}} \quad \text{and} \quad \mathcal{C}_{\epsilon>}^{\mathsf{dt}} = \bigcap_{\text{constants } d} \mathcal{C}_{\epsilon - n^{-d}}^{\mathsf{dt}}.$$

3 Two-sided error

To prove Theorem 1, we first restate it in a more convenient form.

Theorem 1 (Two-sided error, restated). *For integers* $1 \leq q \leq k$*:*

(i) If k, q *are odd:* $\mathsf{BPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{BPP}^{\mathsf{NP}\|[q]}_{q/k>}$.

(ii) If k *is even:* $\mathsf{BPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{BPP}^{\mathsf{NP}\|[q]}_{q/k>}$.

(iii) If q, k *are even:* $\mathsf{BPP}^{\mathsf{NP}[1]}_{1/(k-1)} \not\subseteq \mathsf{BPP}^{\mathsf{NP}\|[q]}_{>q/k}$ *relative to an oracle.*

(iv) If q *is odd:* $\mathsf{BPP}^{\mathsf{NP}[1]}_{1/k} \not\subseteq \mathsf{BPP}^{\mathsf{NP}\|[q]}_{>q/k}$ *relative to an oracle.*

We prove the inclusions *(i)* and *(ii)* in § 3.1 and the separations *(iii)* and *(iv)* in § 3.2.

3.1 Inclusions

We prove the $q = 1$ case of *(i)* in § 3.1.1 and the $q = 1$ case of *(ii)* in § 3.1.2 (together these show that $\mathsf{BPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{BPP}^{\mathsf{NP}[1]}_{1/k>}$ for all integers $k \geq 1$), then we generalize to the $q > 1$ case of *(i)* in § 3.1.3 and the $q > 1$ case of *(ii)* in § 3.1.4. The techniques from [CP08] for the zero-sided error setting are not particularly helpful for the two-sided error setting, so we develop the ideas from scratch.

We now describe the common setup. For some constant c we have $L \in \mathsf{BPP}^{\mathsf{NP}[1]}_{1/(k+1)+n^{-c}}$, witnessed by a polynomial-time randomized algorithm M (taking input x and coin tosses $s \in \{0,1\}^r$) and a language $L' \in \mathsf{NP}$. For an arbitrary constant d, we wish to show $L \in \mathsf{BPP}^{\mathsf{NP}\|[q]}_{q/k-2^{-n^d}}$.

Fix an input x. The first step is to sample a sequence of $m = O(n^{2c+d})$ many independent strings $s^1, \ldots, s^m \in \{0,1\}^r$, so with probability $\geq 1 - 2^{-n^d-1}$, the sequence is *good* in the sense that on input x, M still has advantage strictly greater than $\frac{1}{k+1}$ when its coin tosses are chosen uniformly from the multiset $\{s^1, \ldots, s^m\}$. Then we design a polynomial-time randomized algorithm which, given a good sequence, outputs $L(x)$ with advantage $\geq \frac{q}{k}$ after making q nonadaptive NP oracle queries. Hence, over the random $s^1, \ldots, s^m$ and the other randomness of our algorithm,

$$\begin{aligned}\mathbb{P}[\text{output is } L(x)] \;&\geq\; \mathbb{P}\big[\text{output is } L(x) \,\big|\, s^1, \ldots, s^m \text{ is good}\big] - \mathbb{P}[s^1, \ldots, s^m \text{ is bad}]\\ &\geq\; \big(\tfrac{1}{2} + \tfrac{1}{2} \cdot \tfrac{q}{k}\big) - 2^{-n^d-1} \;=\; \tfrac{1}{2} + \tfrac{1}{2}\big(\tfrac{q}{k} - 2^{-n^d}\big).\end{aligned}$$

Henceforth fix a good sequence $s^1, \ldots, s^m$, and let z^h and $out^h \colon \{0,1\} \to \{0,1\}$ be the query string and truth table produced by $M_{s^h}(x)$ (so the output is $out^h(L'(z^h))$). We assume w.l.o.g. that each out^h is nonconstant, and is hence either identity or negation. Henceforth assume that identity is at least as common as negation among $out^1, \ldots, out^m$; the proof is completely analogous if negation is more common.

Taking probabilities over a uniformly random $h \in [m]$, we make the following definitions.

$$\begin{aligned} \alpha &= \tfrac{1}{2}\mathbb{P}[out^h = \text{id}] & \beta &= \tfrac{1}{2}\mathbb{P}[out^h = \text{neg}]\\ a &= \mathbb{P}\big[out^h = \text{id},\ L'(z^h) = 1\big] - \alpha & b &= \mathbb{P}\big[out^h = \text{neg},\ L'(z^h) = 1\big] - \beta \end{aligned}$$

The key observation is now

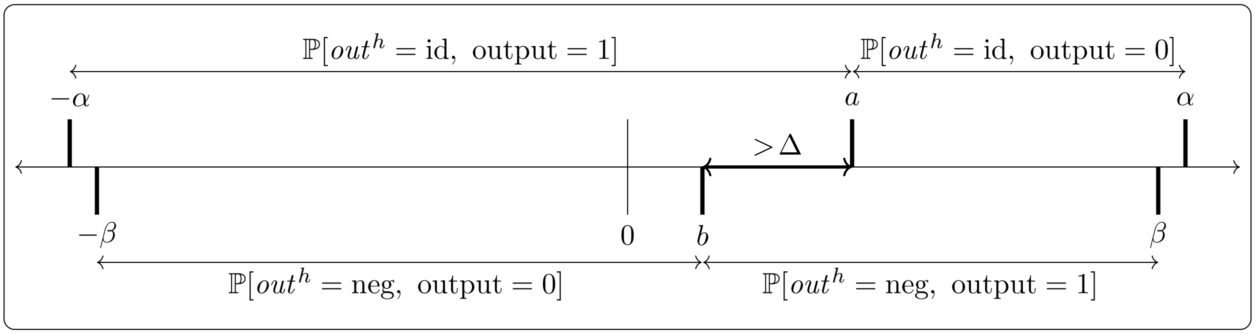
$$\begin{aligned}(a + \alpha) + (\beta - b) \;&=\; \mathbb{P}\big[out^h = \text{id},\ \text{output} = 1\big] + \big(\mathbb{P}[out^h = \text{neg}] - \mathbb{P}\big[out^h = \text{neg},\ \text{output} = 0\big]\big)\\ &=\; \mathbb{P}\big[out^h = \text{id},\ \text{output} = 1\big] + \mathbb{P}\big[out^h = \text{neg},\ \text{output} = 1\big] \;=\; \mathbb{P}[\text{output} = 1]\end{aligned}$$

and thus, defining $\Delta = \frac{1}{2} \cdot \frac{1}{k+1}$, we have

$$a - b \;=\; (a + \alpha) + (\beta - b) - \tfrac{1}{2} \;=\; \mathbb{P}[\text{output} = 1] - \tfrac{1}{2} \begin{cases} > \Delta & \text{if } L(x) = 1\\ < -\Delta & \text{if } L(x) = 0 \end{cases}$$

because of M's advantage w.r.t. a good sequence $s^1, \ldots, s^m$.

This figure shows an example of how these values may fall on the number line if $L(x) = 1$:



$\mathbb{P}[out^h = \text{id},\ \text{output} = 1]$
$\mathbb{P}[out^h = \text{id},\ \text{output} = 0]$
$-\alpha$
a
α
$> \Delta$
$-\beta$
0
b
β
$\mathbb{P}[out^h = \text{neg},\ \text{output} = 0]$
$\mathbb{P}[out^h = \text{neg},\ \text{output} = 1]$

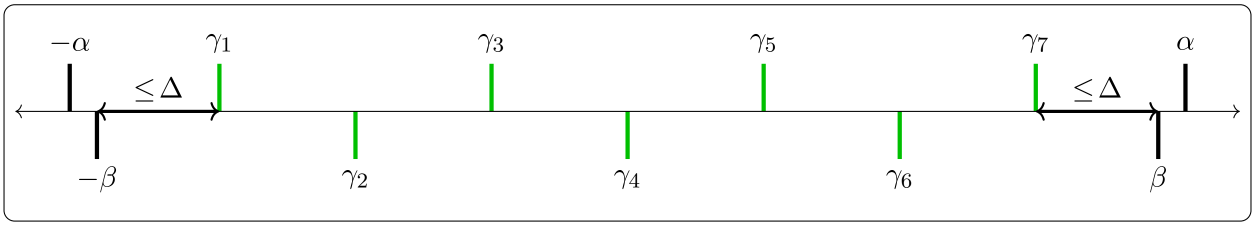
The following summarizes the key properties so far.

$$\begin{array}{rcl} \alpha \ge \beta & a \in [-\alpha, \alpha] & a - b > \Delta \ \text{ if } L(x) = 1 \\ \alpha + \beta = \frac{1}{2} & b \in [-\beta, \beta] & b - a > \Delta \ \text{ if } L(x) = 0 \end{array}$$

Also, for any rational p, testing whether $a \ge p$ can be expressed as an NP oracle query: a witness consists of a list of witnesses for $L'(z^h) = 1$ for at least $(p+\alpha)m$ many h's with $out^h = \text{id}$. Similarly, testing whether $b \ge p$ can be expressed as an NP oracle query.

3.1.1 Proof of *(i)*: $q = 1$

For $i \in [k]$ define $\gamma_i = (i - \frac{k+1}{2})\Delta$. We have $\beta - \gamma_k \le \Delta$ and $\gamma_1 - (-\beta) \le \Delta$ since $\beta \le \frac{1}{4} = \big((k+1) - \frac{k+1}{2}\big)\Delta$. This figure shows an example with $k = 7$:



Our algorithm now picks one of these k possibilities uniformly at random:[2]

- for some odd $i \in [k]$: output 1 iff $a \ge \gamma_i$,
- for some even $i \in [k]$: output 0 iff $b \ge \gamma_i$.

First suppose $L(x) = 1$. We have $a > \gamma_1$ since $a - b > \Delta$ and $b \ge -\beta$ and $\gamma_1 - (-\beta) \le \Delta$. Consider the greatest odd $j \in [k]$ such that $a \ge \gamma_j$; thus $a \ge \gamma_i$ for $\frac{j+1}{2}$ many odd i's $(1, 3, \ldots, j)$. If $j < k$ then $b < \gamma_{j+1}$ since $a - b > \Delta$ and $a < \gamma_{j+2}$; thus $b < \gamma_i$ for at least $\frac{k-j}{2}$ many even i's $(j+1, j+3, \ldots, k-1)$. Hence the probability of outputting 1 is at least $\frac{1}{k}\big(\frac{j+1}{2} + \frac{k-j}{2}\big) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{k}$.

Now suppose $L(x) = 0$. We have $a < \gamma_k$ since $b - a > \Delta$ and $b \le \beta$ and $\beta - \gamma_k \le \Delta$. Consider the least odd $j \in [k]$ such that $a < \gamma_j$; thus $a < \gamma_i$ for $\frac{k-j+2}{2}$ many odd i's $(j, j+2, \ldots, k)$. If

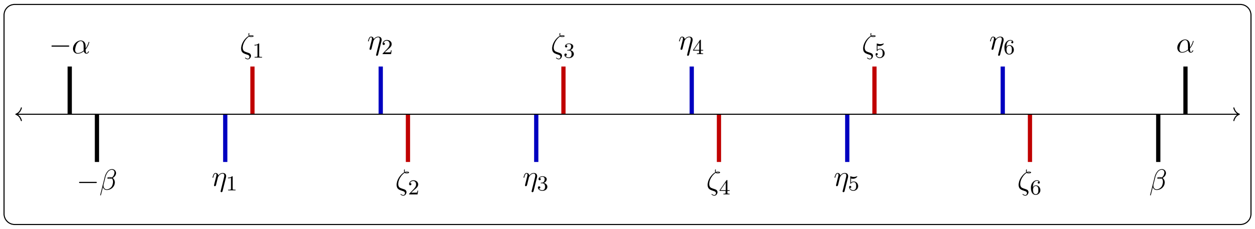
[2] Of course, if k is not a power of 2 and we insist on using uniform coin flips as our only source of randomness, then we must incur a tiny error since it is not possible to exactly sample $i \in [k]$ uniformly. We sweep this pedantic issue under the rug throughout the paper.

$j > 1$ then $b > \gamma_{j-1}$ since $b - a > \Delta$ and $a \geq \gamma_{j-2}$; thus $b \geq \gamma_i$ for at least $\frac{j-1}{2}$ many even i's $(2, 4, \ldots, j-1)$. Hence the probability of outputting 0 is at least $\frac{1}{k}\big(\frac{k-j+2}{2} + \frac{j-1}{2}\big) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{k}$.

That concludes the formal proof, but here is an intuitive way to visualize what is happening: Call γ_i for odd i "upper marks," and call γ_i for even i "lower marks," and assume for convenience all lower marks are in $(-\beta, \beta)$. Suppose $L(x) = 1$ and $b = -\beta$ so $a > \gamma_1$; then at least one upper mark is left of a and all $\frac{k-1}{2}$ lower marks are right of b, resulting in $\frac{k+1}{2}$ of the algorithm's possibilities outputting 1. Now as we continuously sweep a and b to the right, keeping $a - b$ fixed, a passes each upper mark before b passes the preceding lower mark, so at all times at least $\frac{k+1}{2}$ of the possibilities output 1. Suppose $L(x) = 0$ and $b = \beta$ so $a < \gamma_k$; then at least one upper mark is right of a and all $\frac{k-1}{2}$ lower marks are left of b, resulting in $\frac{k+1}{2}$ of the algorithm's possibilities outputting 0. Now as we continuously sweep a and b to the left, keeping $b - a$ fixed, a passes each upper mark before b passes the succeeding lower mark, so at all times at least $\frac{k+1}{2}$ of the possibilities output 0.

3.1.2 Proof of *(ii)*: $q = 1$

For $i \in [k]$ define $\zeta_i = -\beta + i\Delta$ and $\eta_i = -\alpha + i\Delta$. Note that $\alpha - \zeta_k = \Delta$ (so $\zeta_1, \ldots, \zeta_k$ divide the interval $[-\beta, \alpha]$ into $k+1$ subintervals each of length Δ) and $\beta - \eta_k = \Delta$ (so $\eta_1, \ldots, \eta_k$ divide the interval $[-\alpha, \beta]$ into $k+1$ subintervals each of length Δ). This figure shows an example with $k = 6$:



Our algorithm now picks one of these $2k$ possibilities uniformly at random:

- for some odd $i \in [k]$: output 1 iff $a \geq \zeta_i$,
- for some even $i \in [k]$: output 0 iff $b \geq \zeta_i$,
- for some even $i \in [k]$: output 1 iff $a \geq \eta_i$,
- for some odd $i \in [k]$: output 0 iff $b \geq \eta_i$.

First suppose $L(x) = 1$. We have $a > \zeta_1$ since $a - b > \Delta$ and $b \geq -\beta$. Consider the greatest odd $j \in [k]$ such that $a \geq \zeta_j$; thus $a \geq \zeta_i$ for $\frac{j+1}{2}$ many odd i's $(1, 3, \ldots, j)$. We have $b < \zeta_{j+1}$ since $a - b > \Delta$ and either $a < \zeta_{j+2}$ (if $j < k-1$) or $a \leq \alpha$ and $\alpha - \zeta_k = \Delta$ (if $j = k-1$); thus $b < \zeta_i$ for at least $\frac{k-j+1}{2}$ many even i's $(j+1, j+3, \ldots, k)$. Consider the greatest even $j' \in [k]$ such that $a \geq \eta_{j'}$, or let $j' = 0$ if it does not exist; thus $a \geq \eta_i$ for $\frac{j'}{2}$ many even i's $(2, 4, \ldots, j')$. If $j' < k$ then $b < \eta_{j'+1}$ since $a - b > \Delta$ and $a < \eta_{j'+2}$; thus $b < \eta_i$ for at least $\frac{k-j'}{2}$ many odd i's $(j'+1, j'+3, \ldots, k-1)$. Hence the probability of outputting 1 is at least $\frac{1}{2k}\big(\frac{j+1}{2} + \frac{k-j+1}{2} + \frac{j'}{2} + \frac{k-j'}{2}\big) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{k}$.

Now suppose $L(x) = 0$. Consider the least odd $j \in [k]$ such that $a < \zeta_j$, or let $j = k+1$ if it does not exist; thus $a < \zeta_i$ for $\frac{k-j+1}{2}$ many odd i's $(j, j+2, \ldots, k-1)$. If $j > 1$ then $b > \zeta_{j-1}$ since $b - a > \Delta$ and $a \geq \zeta_{j-2}$; thus $b \geq \zeta_i$ for at least $\frac{j-1}{2}$ many even i's $(2, 4, \ldots, j-1)$. We have $a < \eta_k$ since $b - a > \Delta$ and $b \leq \beta$ and $\beta - \eta_k = \Delta$. Consider the least even $j' \in [k]$ such that $a < \eta_{j'}$; thus $a < \eta_i$ for $\frac{k-j'+2}{2}$ many even i's $(j', j'+2, \ldots, k)$. We have $b > \eta_{j'-1}$ since $b - a > \Delta$ and either $a \geq \eta_{j'-2}$ (if $j' > 2$) or $a \geq -\alpha$ (if $j' = 2$); thus $b \geq \eta_i$ for at least $\frac{j'}{2}$ many odd i's $(1, 3, \ldots, j'-1)$. Hence the probability of outputting 0 is at least $\frac{1}{2k}\big(\frac{k-j+1}{2} + \frac{j-1}{2} + \frac{k-j'+2}{2} + \frac{j'}{2}\big) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{k}$.

That concludes the formal proof, but here is an intuitive way to visualize what is happening: Call ζ_i for odd i and η_i for even i "upper marks," and call ζ_i for even i and η_i for odd i "lower marks," and assume for convenience all lower marks are in $(-\beta, \beta)$. Suppose $L(x) = 1$ and $b = -\beta$ so $a > \zeta_1$; then at least one upper mark is left of a and all k lower marks are right of b, resulting in $k+1$ of the algorithm's possibilities outputting 1. Now as we continuously sweep a and b to the right, keeping $a - b$ fixed, a passes each upper mark (ζ_i or η_i) before b passes the corresponding preceding lower mark (ζ_{i-1} or η_{i-1} respectively), so at all times at least $k+1$ of the possibilities output 1. Suppose $L(x) = 0$ and $b = \beta$ so $a < \eta_k$; then at least one upper mark is right of a and all k lower marks are left of b, resulting in $k+1$ of the algorithm's possibilities outputting 0. Now as we continuously sweep a and b to the left, keeping $b - a$ fixed, a passes each upper mark (ζ_i or η_i) before b passes the corresponding succeeding lower mark (ζ_{i+1} or η_{i+1} respectively), so at all times at least $k+1$ of the possibilities output 0.

3.1.3 Proof of *(i)*: $q > 1$

For $i \in [k]$ define I_i as the set of q successive integers starting with i and wrapping around to 1 when k is exceeded: $I_i = \{i, i+1, \ldots, i+q-1\}$ if $i \le k-q+1$, and $I_i = \{i, i+1, \ldots, k, 1, 2, \ldots, i+q-1-k\}$ if $i > k-q+1$. Define $i^{\nwarrow} = \min(\text{odd } i' \in I_i) - k - 1$ and $i^{\swarrow} = \min(\text{even } i' \in I_i) - k - 1$; the $-k-1$ is a simple way to ensure $i^{\nwarrow}, i^{\swarrow} < \min(i' \in I_i)$. Since k, q are odd, the sorted order of $I_i \cup \{i^{\nwarrow}, i^{\swarrow}\}$ alternates between odd and even numbers.

Our algorithm picks $i \in [k]$ uniformly at random and for each $i' \in I_i$ does an oracle query to see whether $a \ge \gamma_{i'}$ if i' is odd, or whether $b \ge \gamma_{i'}$ if i' is even. Consider the greatest odd $i^\sharp \in I_i$ such that $a \ge \gamma_{i^\sharp}$, or let $i^\sharp = i^{\nwarrow}$ if it does not exist. Consider the greatest even $i^\flat \in I_i$ such that $b \ge \gamma_{i^\flat}$, or let $i^\flat = i^{\swarrow}$ if it does not exist. Our algorithm outputs 1 if $i^\sharp > i^\flat$, or 0 if $i^\flat > i^\sharp$.

The intuition is that if I_i were the whole set $[k]$, then with certainty we would have $i^\sharp > i^\flat$ if $a - b > \Delta$, and $i^\flat > i^\sharp$ if $b - a > \Delta$. Since I_i is a q-subset of $[k]$, comparing $i^\sharp$ and $i^\flat$ gives our best guess for $L(x)$ based on the "limited view" provided by these oracle queries. About q of the I_i sets are close enough to a to detect whether a or b is larger. Among the other $k - q$ sets, about half get it right through luck. Thus $q + \frac{k-q}{2}$ out of the k sets lead to correct output, which implies the advantage is $\frac{q}{k}$. Careful case analysis is needed for the I_i sets that wrap around. Here is the formal proof.

First suppose $L(x) = 1$. Consider the greatest odd $j \in [k]$ such that $a \ge \gamma_j$ (which exists since $a > \gamma_1$). We have $i^\sharp > i^\flat$ if one of the following mutually exclusive events holds:

(1) $j \in I_i$, since then $i^\sharp = j$ and $i^\flat \le j - 1$ (since $b < \gamma_{j+1}$ if $j < k$);

(2) i is odd and $i \le j - q - 1$, since then $i^\sharp = i + q - 1$ and trivially $i^\flat \le i + q - 2$;

(3) i is even and $j + 1 \le i \le j - q - 1 + k$, since then either:

- $i \le k - q$, in which case $i^\sharp = i^{\nwarrow} > i^{\swarrow} = i^\flat$, or
- $i = k - q + 2$, in which case $i^\sharp = 1$ and $i^\flat = i^{\swarrow} < 1$, or
- $i \ge k - q + 4$, in which case $i^\sharp = i + q - 1 - k$ and $i^\flat \le i + q - 2 - k$.

There are q many type-*(1)* i's. If $j > q$ then there are $\frac{j-q}{2}$ many type-*(2)* i's $(1, 3, \ldots, j-q-1)$ and $\frac{k-j}{2}$ many type-*(3)* i's $(j+1, j+3, \ldots, k-1)$. If $j \le q$ then there are $\frac{k-q}{2}$ many type-*(3)* i's $(j+1, j+3, \ldots, j-q-1+k)$. Either way, $i^\sharp > i^\flat$ holds for at least $q + \frac{k-q}{2} = \frac{k+q}{2}$ many i's, and hence the probability of outputting 1 is at least $\frac{1}{k} \cdot \frac{k+q}{2} = \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$.

Now suppose $L(x) = 0$. Consider the least odd $j \in [k]$ such that $a < \gamma_j$ (which exists since $a < \gamma_k$). As a special case, if $j = 1$ then $i^\sharp = i^{\nwarrow}$ and so $i^\flat > i^\sharp$ if $i^{\swarrow} > i^{\nwarrow}$, which happens for $\frac{k+q}{2}$

many i's $(1, 3, \ldots, k-q+1$ and $k-q+2, k-q+3, \ldots, k)$. Now assume $j > 1$. We have $i^\flat > i^\sharp$ if one of the following mutually exclusive events holds:

(1) $j-1 \in I_i$, since then $i^\sharp \leq j-2$ and $i^\flat \geq j-1$ (since $b > \gamma_{j-1}$ if $j > 1$);
(2) i is even and $i \leq j-q-2$, since then $i^\sharp = i+q-2$ and $i^\flat = i+q-1$;
(3) i is odd and $j \leq i \leq j-q-2+k$, since then either:

- $i \leq k-q+1$, in which case $i^\sharp = i^\nwarrow < i^\swarrow \leq i^\flat$, or
- $i \geq k-q+3$, in which case $i^\sharp = i+q-2-k$ and $i^\flat \geq i+q-1-k$.

There are q many type-*(1)* i's. If $j > q$ then there are $\frac{j-q-2}{2}$ many type-*(2)* i's $(2, 4, \ldots, j-q-2)$ and $\frac{k-j+2}{2}$ many type-*(3)* i's $(j, j+2, \ldots, k)$. If $j \leq q$ then there are $\frac{k-q}{2}$ many type-*(3)* i's $(j, j+2, \ldots, j-q-2+k)$. Either way, $i^\flat > i^\sharp$ holds for at least $q + \frac{k-q}{2} = \frac{k+q}{2}$ many i's, and hence the probability of outputting 0 is at least $\frac{1}{k} \cdot \frac{k+q}{2} = \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$.

3.1.4 Proof of *(ii)*: $q > 1$

We retain the definition of I_i from § 3.1.3. Now we have separate cases for whether q is even or odd. The case q is even involves a natural combination of the ideas from § 3.1.2 and § 3.1.3, but the case q is odd is more subtle.

If q is even: Our algorithm picks $i \in [k]$ uniformly at random, and with probability $\frac{1}{2}$ each:

- Define $i^\nwarrow = \min(\text{odd } i' \in I_i) - k$ and $i^\swarrow = \min(\text{even } i' \in I_i) - k$. For each $i' \in I_i$ do an oracle query to see whether $a \geq \zeta_{i'}$ if i' is odd, or whether $b \geq \zeta_{i'}$ if i' is even. Consider the greatest odd $i^\sharp \in I_i$ such that $a \geq \zeta_{i^\sharp}$, or let $i^\sharp = i^\nwarrow$ if it does not exist. Consider the greatest even $i^\flat \in I_i$ such that $b \geq \zeta_{i^\flat}$, or let $i^\flat = i^\swarrow$ if it does not exist. Output 1 if $i^\sharp > i^\flat$, or 0 if $i^\flat > i^\sharp$.
- Define $i^\nwarrow = \min(\text{even } i' \in I_i) - k$ and $i^\swarrow = \min(\text{odd } i' \in I_i) - k$. For each $i' \in I_i$ do an oracle query to see whether $a \geq \eta_{i'}$ if i' is even, or whether $b \geq \eta_{i'}$ if i' is odd. Consider the greatest even $i^\sharp \in I_i$ such that $a \geq \eta_{i^\sharp}$, or let $i^\sharp = i^\nwarrow$ if it does not exist. Consider the greatest odd $i^\flat \in I_i$ such that $b \geq \eta_{i^\flat}$, or let $i^\flat = i^\swarrow$ if it does not exist. Output 1 if $i^\sharp > i^\flat$, or 0 if $i^\flat > i^\sharp$.

First suppose $L(x) = 1$. Assume the algorithm picks the first bullet. Consider the greatest odd $j \in [k]$ such that $a \geq \zeta_j$ (which exists since $a > \zeta_1$). We have $i^\sharp > i^\flat$ if one of the following mutually exclusive events holds:

(1) $j \in I_i$, since then $i^\sharp = j$ and $i^\flat \leq j-1$ (since $b < \zeta_{j+1}$);
(2) i is even and $i \leq j-q-1$, since then $i^\sharp = i+q-1$ and trivially $i^\flat \leq i+q-2$;
(3) i is even and $j+1 \leq i \leq j-q-1+k$, since then either:

- $i \leq k-q$, in which case $i^\sharp = i^\nwarrow > i^\swarrow = i^\flat$, or
- $i = k-q+2$, in which case $i^\sharp = 1$ and $i^\flat = i^\swarrow < 1$, or
- $i \geq k-q+4$, in which case $i^\sharp = i+q-1-k$ and $i^\flat \leq i+q-2-k$.

There are q many type-*(1)* i's. If $j > q$ then there are $\frac{j-q-1}{2}$ many type-*(2)* i's $(2, 4, \ldots, j-q-1)$ and $\frac{k-j+1}{2}$ many type-*(3)* i's $(j+1, j+3, \ldots, k)$. If $j \leq q$ then there are $\frac{k-q}{2}$ many type-*(3)* i's $(j+1, j+3, \ldots, j-q-1+k)$. Either way, $i^\sharp > i^\flat$ holds for at least $q + \frac{k-q}{2} = \frac{k+q}{2}$ many i's.

Assume the algorithm picks the second bullet. As a special case, if $a < \eta_2$ (so $b < \eta_1$) then $i^\sharp = i^\nwarrow$ and $i^\flat = i^\swarrow$ and so $i^\sharp > i^\flat$ happens for $\frac{k+q}{2}$ many i's $(1, 3, \ldots, k-q+1$ and $k-q+2, k-q+3, \ldots, k)$.

Otherwise, consider the greatest even $j' \in [k]$ such that $a \geq \eta_{j'}$. We have $i^\sharp > i^\flat$ if one of the following mutually exclusive events holds:

(1) $j' \in I_i$, since then $i^\sharp = j'$ and $i^\flat \leq j' - 1$ (since $b < \eta_{j'+1}$ if $j' < k$);
(2) i is odd and $i \leq j' - q - 1$, since then $i^\sharp = i + q - 1$ and trivially $i^\flat \leq i + q - 2$;
(3) i is odd and $j' + 1 \leq i \leq j' - q - 1 + k$, since then either:

- $i \leq k - q + 1$, in which case $i^\sharp = i^\nwarrow > i^\swarrow = i^\flat$, or
- $i \geq k - q + 3$, in which case $i^\sharp = i + q - 1 - k$ and $i^\flat \leq i + q - 2 - k$.

There are q many type-*(1)* i's. If $j' > q$ then there are $\frac{j'-q}{2}$ many type-*(2)* i's $(1, 3, \ldots, j' - q - 1)$ and $\frac{k-j'}{2}$ many type-*(3)* i's $(j' + 1, j' + 3, \ldots, k - 1)$. If $j' \leq q$ then there are $\frac{k-q}{2}$ many type-*(3)* i's $(j' + 1, j' + 3, \ldots, j' - q - 1 + k)$. Either way, $i^\sharp > i^\flat$ holds for at least $q + \frac{k-q}{2} = \frac{k+q}{2}$ many i's.

In summary, out of the $2k$ possible random outcomes, at least $k + q$ of them result in $i^\sharp > i^\flat$, and hence the probability of outputting 1 is at least $\frac{1}{2k}(k + q) = \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$.

Now suppose $L(x) = 0$. Assume the algorithm picks the first bullet. Consider the least odd $j \in [k]$ such that $a < \zeta_j$, or let $j = k + 1$ if it does not exist. As a special case, if $j = 1$ then $i^\sharp = i^\nwarrow$ and so $i^\flat > i^\sharp$ if $i^\swarrow > i^\nwarrow$, which happens for $\frac{k+q}{2}$ many i's $(1, 3, \ldots, k - q + 1$ and $k - q + 2, k - q + 3, \ldots, k)$. Now assume $j > 1$. We have $i^\flat > i^\sharp$ if one of the following mutually exclusive events holds:

(1) $j - 1 \in I_i$, since then $i^\sharp \leq j - 2$ and $i^\flat \geq j - 1$ (since $b > \zeta_{j-1}$ if $j > 1$);
(2) i is odd and $i \leq j - q - 2$, since then $i^\sharp = i + q - 2$ and $i^\flat = i + q - 1$;
(3) i is odd and $j \leq i \leq j - q - 2 + k$, since then either:

- $i \leq k - q + 1$, in which case $i^\sharp = i^\nwarrow < i^\swarrow \leq i^\flat$, or
- $i \geq k - q + 3$, in which case $i^\sharp = i + q - 2 - k$ and $i^\flat \geq i + q - 1 - k$.

There are q many type-*(1)* i's. If $j > q$ then there are $\frac{j-q-1}{2}$ many type-*(2)* i's $(1, 3, \ldots, j - q - 2)$ and $\frac{k-j+1}{2}$ many type-*(3)* i's $(j, j + 2, \ldots, k - 1)$. If $j \leq q$ then there are $\frac{k-q}{2}$ many type-*(3)* i's $(j, j + 2, \ldots, j - q - 2 + k)$. Either way, $i^\flat > i^\sharp$ holds for at least $q + \frac{k-q}{2} = \frac{k+q}{2}$ many i's.

Assume the algorithm picks the second bullet. Consider the least even $j' \in [k]$ such that $a < \eta_{j'}$ (which exists since $a < \eta_k$). We have $i^\flat > i^\sharp$ if one of the following mutually exclusive events holds:

(1) $j' - 1 \in I_i$, since then $i^\sharp \leq j' - 2$ and $i^\flat \geq j' - 1$ (since $b > \eta_{j'-1}$);
(2) i is even and $i \leq j' - q - 2$, since then $i^\sharp = i + q - 2$ and $i^\flat = i + q - 1$;
(3) i is even and $j' \leq i \leq j' - q - 2 + k$, since then either:

- $i \leq k - q$, in which case $i^\sharp = i^\nwarrow < i^\swarrow \leq i^\flat$, or
- $i = k - q + 2$, in which case $i^\sharp = i^\nwarrow < 1$ and $i^\flat \geq 1$, or
- $i \geq k - q + 4$, in which case $i^\sharp = i + q - 2 - k$ and $i^\flat \geq i + q - 1 - k$.

There are q many type-*(1)* i's. If $j' > q$ then there are $\frac{j'-q-2}{2}$ many type-*(2)* i's $(2, 4, \ldots, j' - q - 2)$ and $\frac{k-j'+2}{2}$ many type-*(3)* i's $(j', j' + 2, \ldots, k)$. If $j' \leq q$ then there are $\frac{k-q}{2}$ many type-*(3)* i's $(j', j' + 2, \ldots, j' - q - 2 + k)$. Either way, $i^\flat > i^\sharp$ holds for at least $q + \frac{k-q}{2} = \frac{k+q}{2}$ many i's.

In summary, out of the $2k$ possible random outcomes, at least $k + q$ of them result in $i^\flat > i^\sharp$, and hence the probability of outputting 0 is at least $\frac{1}{2k}(k + q) = \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$.

If q is odd and $\beta > \alpha - \Delta$: We handle the case $\beta \le \alpha - \Delta$ later, in a different way. The assumption $\beta > \alpha - \Delta$ ensures the ζ and η marks are perfectly interspersed (as shown in the figure in § 3.1.2), which is essential for the algorithm we now describe.

For this case, we form $\zeta_1, \ldots, \zeta_k$ and $\eta_1, \ldots, \eta_k$ into one big cycle, rather than two separate cycles. Thus when I_i "wraps around," we switch between making "ζ queries" and making "η queries." To facilitate this idea, we partition I_i as follows.

$$\begin{aligned} I_i^{\ge,\mathrm{odd}} &= \{\text{odd } i' \ge i \text{ in } I_i\} & \qquad I_i^{\ge,\mathrm{even}} &= \{\text{even } i' \ge i \text{ in } I_i\} \\ I_i^{<,\mathrm{odd}} &= \{\text{odd } i' < i \text{ in } I_i\} & \qquad I_i^{<,\mathrm{even}} &= \{\text{even } i' < i \text{ in } I_i\} \end{aligned}$$

Our algorithm picks $i \in [k]$ uniformly at random, and with probability $\frac{1}{2}$ each:

- Define $i^{\nwarrow} = \min(I_i^{\ge,\mathrm{odd}} \cup I_i^{<,\mathrm{even}}) - k$ and $i^{\swarrow} = \min(I_i^{\ge,\mathrm{even}} \cup I_i^{<,\mathrm{odd}}) - k$. For each $i' \in I_i$ do an oracle query to see whether

  $$\begin{aligned} a &\ge \zeta_{i'} \quad \text{if } i' \in I_i^{\ge,\mathrm{odd}}, & \qquad b &\ge \zeta_{i'} \quad \text{if } i' \in I_i^{\ge,\mathrm{even}}, \\ b &\ge \eta_{i'} \quad \text{if } i' \in I_i^{<,\mathrm{odd}}, & \qquad a &\ge \eta_{i'} \quad \text{if } i' \in I_i^{<,\mathrm{even}}. \end{aligned}$$

  Consider the greatest $i^\sharp \in I_i^{\ge,\mathrm{odd}} \cup I_i^{<,\mathrm{even}}$ such that the corresponding oracle query returns 1, or let $i^\sharp = i^{\nwarrow}$ if it does not exist. Consider the greatest $i^\flat \in I_i^{\ge,\mathrm{even}} \cup I_i^{<,\mathrm{odd}}$ such that the corresponding oracle query returns 1, or let $i^\flat = i^{\swarrow}$ if it does not exist. Output 1 if $i^\sharp > i^\flat$, or 0 if $i^\flat > i^\sharp$.
- Define $i^{\nwarrow} = \min(I_i^{\ge,\mathrm{even}} \cup I_i^{<,\mathrm{odd}}) - k$ and $i^{\swarrow} = \min(I_i^{\ge,\mathrm{odd}} \cup I_i^{<,\mathrm{even}}) - k$. For each $i' \in I_i$ do an oracle query to see whether

  $$\begin{aligned} b &\ge \eta_{i'} \quad \text{if } i' \in I_i^{\ge,\mathrm{odd}}, & \qquad a &\ge \eta_{i'} \quad \text{if } i' \in I_i^{\ge,\mathrm{even}}, \\ a &\ge \zeta_{i'} \quad \text{if } i' \in I_i^{<,\mathrm{odd}}, & \qquad b &\ge \zeta_{i'} \quad \text{if } i' \in I_i^{<,\mathrm{even}}. \end{aligned}$$

  Consider the greatest $i^\sharp \in I_i^{\ge,\mathrm{even}} \cup I_i^{<,\mathrm{odd}}$ such that the corresponding oracle query returns 1, or let $i^\sharp = i^{\nwarrow}$ if it does not exist. Consider the greatest $i^\flat \in I_i^{\ge,\mathrm{odd}} \cup I_i^{<,\mathrm{even}}$ such that the corresponding oracle query returns 1, or let $i^\flat = i^{\swarrow}$ if it does not exist. Output 1 if $i^\sharp > i^\flat$, or 0 if $i^\flat > i^\sharp$.

First suppose $L(x) = 1$. As a special case, if $a < \eta_2$ (so $b < \eta_1$) then $i^\flat = i^{\swarrow}$ and so $i^\sharp > i^\flat$ if either $i^{\nwarrow} > i^{\swarrow}$ or $i^\sharp \ge 1$, which happens for $\frac{k+q+1}{2}$ many i's in the first bullet (1 and $2, 4, \ldots, k-q+1$ and $k-q+2, k-q+3, \ldots, k$) and $\frac{k+q-1}{2}$ many i's in the second bullet ($1, 3, \ldots, k-q$ and $k-q+2, k-q+3, \ldots, k$).

Otherwise, consider the greatest odd $j \in [k]$ such that $a \ge \zeta_j$ and the greatest even $j' \in [k]$ such that $a \ge \eta_{j'}$, and note that $j' \in \{j-1, j+1\}$ since $\beta > \alpha - \Delta$.

Assume the algorithm picks the first bullet. We have $i^\sharp > i^\flat$ if one of the following mutually exclusive events holds:[3]

(1) $j \in I_i^{\ge,\mathrm{odd}}$, since then $i^\sharp = j$ and $i^\flat \le j-1$ (since $b < \zeta_{j+1}$);

(2) i is odd and $i \le j - q - 1$, since then $i^\sharp = i + q - 1$ and trivially $i^\flat \le i + q - 2$;

(3) i is even and $j + 1 \le i \le j' - q - 1 + k$, since then either:

  - $i \le k - q + 1$, in which case $i^\sharp = i^{\nwarrow} > i^{\swarrow} = i^\flat$, or

[3] *(1)* and *(4)* cannot happen simultaneously, since that would force $q = k$.

- $i \geq k-q+3$, in which case $i^\sharp = i+q-1-k$ and $i^\flat \leq i+q-2-k$;

(4) $j' \in I_i^{<,\text{even}}$, since then $i^\sharp = j'$ (since $i \geq j'+2$) and $i^\flat \leq j'-1$ (since $b < \eta_{j'+1}$).

If $j' > q$ then there are q many type-*(1)* i's $(j-q+1, j-q+2, \ldots, j)$ and $\frac{j-q}{2}$ many type-*(2)* i's $(1, 3, \ldots, j-q-1)$ and $\frac{k-j+1}{2}$ many type-*(3)* i's $(j+1, j+3, \ldots, k)$, so $i^\sharp > i^\flat$ holds for at least $\frac{k+q+1}{2}$ many i's. If $j' \leq q$ then there are j many type-*(1)* i's $(1, 2, \ldots, j)$ and $\frac{k-q+j'-j}{2}$ many type-*(3)* i's $(j+1, j+3, \ldots, j'-q-1+k)$ and $q-j'$ many type-*(4)* i's $(j'-q+1+k, j'-q+2+k, \ldots, k)$, so $i^\sharp > i^\flat$ holds for at least $\frac{k+q-j'+j}{2}$ many i's.

Assume the algorithm picks the second bullet. We have $i^\sharp > i^\flat$ if one of the following mutually exclusive events holds:[3]

(1) $j' \in I_i^{\geq,\text{even}}$, since then $i^\sharp = j'$ and $i^\flat \leq j'-1$ (since $b < \eta_{j'+1}$ if $j' < k$);

(2) i is even and $i \leq j'-q-1$, since then $i^\sharp = i+q-1$ and trivially $i^\flat \leq i+q-2$;

(3) i is odd and $j'+1 \leq i \leq j-q-1+k$, since then either:

- $i \leq k-q$, in which case $i^\sharp = i^\nwarrow > i^\swarrow = i^\flat$, or
- $i = k-q+2$, in which case $i^\sharp = 1$ and $i^\flat = i^\swarrow < 1$, or
- $i \geq k-q+4$, in which case $i^\sharp = i+q-1-k$ and $i^\flat \leq i+q-2-k$;

(4) $j \in I_i^{<,\text{odd}}$, since then $i^\sharp = j$ (since $i \geq j+2$) and $i^\flat \leq j-1$ (since $b < \zeta_{j+1}$).

If $j' > q$ then there are q many type-*(1)* i's $(j'-q+1, j'-q+2, \ldots, j')$ and $\frac{j'-q-1}{2}$ many type-*(2)* i's $(2, 4, \ldots, j'-q-1)$ and $\frac{k-j'}{2}$ many type-*(3)* i's $(j'+1, j'+3, \ldots, k-1)$, so $i^\sharp > i^\flat$ holds for at least $\frac{k+q-1}{2}$ many i's. If $j' \leq q$ then there are j' many type-*(1)* i's $(1, 2, \ldots, j')$ and $\frac{k-q+j-j'}{2}$ many type-*(3)* i's $(j'+1, j'+3, \ldots, j-q-1+k)$ and $q-j$ many type-*(4)* i's $(j-q+1+k, j-q+2+k, \ldots, k)$, so $i^\sharp > i^\flat$ holds for at least $\frac{k+q-j+j'}{2}$ many i's.

In summary, out of the $2k$ possible random outcomes, at least $k+q$ of them result in $i^\sharp > i^\flat$ (at least $\frac{k+q+1}{2} + \frac{k+q-1}{2}$ if $a < \eta_2$ or $j' > q$, and at least $\frac{k+q-j'+j}{2} + \frac{k+q-j+j'}{2}$ if $j' \leq q$), and hence the probability of outputting 1 is at least $\frac{1}{2k}(k+q) = \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$.

Now suppose $L(x) = 0$. Consider the least odd $j \in [k]$ such that $a < \zeta_j$, or let $j = k+1$ if it does not exist, and consider the least even $j' \in [k]$ such that $a < \eta_{j'}$ (which exists since $a < \eta_k$), and note that $j' \in \{j-1, j+1\}$ since $\beta > \alpha - \Delta$.

As a special case, if $j = 1$ then $i^\sharp = i^\nwarrow$ and so $i^\flat > i^\sharp$ if either $i^\swarrow > i^\nwarrow$ or $i^\flat \geq 1$, which happens for $\frac{k+q-1}{2}$ many i's in the first bullet $(1, 3, \ldots, k-q$ and $k-q+2, k-q+3, \ldots, k)$ and $\frac{k+q+1}{2}$ many i's in the second bullet (1 and $2, 4, \ldots, k-q+1$ and $k-q+2, k-q+3, \ldots, k)$. Otherwise, assume $j > 1$.

Assume the algorithm picks the first bullet. We have $i^\flat > i^\sharp$ if one of the following mutually exclusive events holds:[3]

(1) $j-1 \in I_i^{\geq,\text{even}}$, since then $i^\sharp \leq j-2$ and $i^\flat \geq j-1$ (since $b > \zeta_{j-1}$ if $j > 1$);

(2) i is even and $i \leq j-q-2$, since then $i^\sharp = i+q-2$ and $i^\flat = i+q-1$;

(3) i is odd and $j \leq i \leq j'-q-2+k$, since then either:

- $i \leq k-q$, in which case $i^\sharp = i^\nwarrow < i^\swarrow \leq i^\flat$, or
- $i = k-q+2$, in which case $i^\sharp = i^\nwarrow < 1$ and $i^\flat \geq 1$, or
- $i \geq k-q+4$, in which case $i^\sharp = i+q-2-k$ and $i^\flat \geq i+q-1-k$;

(4) $j'-1 \in I_i^{<,\text{odd}}$, since then $i^\sharp \leq j'-2$ (since $i \geq j'+1$) and $i^\flat \geq j'-1$ (since $b > \eta_{j'-1}$).

If $j > q$ then there are q many type-*(1)* i's $(j-q, j-q+1, \ldots, j-1)$ and $\frac{j-q-2}{2}$ many type-*(2)* i's $(2, 4, \ldots, j-q-2)$ and $\frac{k-j+1}{2}$ many type-*(3)* i's $(j, j+2, \ldots, k-1)$, so $i^\flat > i^\sharp$ holds for at least $\frac{k+q-1}{2}$ many i's. If $j \leq q$ then there are $j-1$ many type-*(1)* i's $(1, 2, \ldots, j-1)$ and $\frac{k-q+j'-j}{2}$ many type-*(3)* i's $(j, j+2, \ldots, j'-q-2+k)$ and $q-j'+1$ many type-*(4)* i's $(j'-q+k, j'-q+1+k, \ldots, k)$, so $i^\flat > i^\sharp$ holds for at least $\frac{k+q-j'+j}{2}$ many i's.

Assume the algorithm picks the second bullet. We have $i^\flat > i^\sharp$ if one of the following mutually exclusive events holds:[3]

(1) $j' - 1 \in I_i^{\geq,\mathrm{odd}}$, since then $i^\sharp \leq j' - 2$ and $i^\flat \geq j' - 1$ (since $b > \eta_{j'-1}$);
(2) i is odd and $i \leq j' - q - 2$, since then $i^\sharp = i + q - 2$ and $i^\flat = i + q - 1$;
(3) i is even and $j' \leq i \leq j - q - 2 + k$, since then either:

- $i \leq k - q + 1$, in which case $i^\sharp = i^\nwarrow < i^\swarrow \leq i^\flat$, or
- $i \geq k - q + 3$, in which case $i^\sharp = i + q - 2 - k$ and $i^\flat \geq i + q - 1 - k$;

(4) $j - 1 \in I_i^{<,\mathrm{even}}$, since then $i^\sharp = j - 2$ (since $i \geq j + 1$) and $i^\flat \geq j - 1$ (since $b > \zeta_{j-1}$).

If $j > q$ then there are q many type-*(1)* i's $(j'-q, j'-q+1, \ldots, j'-1)$ and $\frac{j'-q-1}{2}$ many type-*(2)* i's $(1, 3, \ldots, j'-q-2)$ and $\frac{k-j'+2}{2}$ many type-*(3)* i's $(j', j'+2, \ldots, k)$, so $i^\flat > i^\sharp$ holds for at least $\frac{k+q+1}{2}$ many i's. If $j \leq q$ then there are $j'-1$ many type-*(1)* i's $(1, 2, \ldots, j'-1)$ and $\frac{k-q+j-j'}{2}$ many type-*(3)* i's $(j', j'+2, \ldots, j-q-2+k)$ and $q-j+1$ many type-*(4)* i's $(j-q+k, j-q+1+k, \ldots, k)$, so $i^\flat > i^\sharp$ holds for at least $\frac{k+q-j+j'}{2}$ many i's.

In summary, out of the $2k$ possible random outcomes, at least $k+q$ of them result in $i^\flat > i^\sharp$ (at least $\frac{k+q-1}{2} + \frac{k+q+1}{2}$ if $j = 1$ or $j > q$, and at least $\frac{k+q-j'+j}{2} + \frac{k+q-j+j'}{2}$ if $j \leq q$), and hence the probability of outputting 0 is at least $\frac{1}{2k}(k+q) = \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$.

If q is odd and $\beta \leq \alpha - \Delta$: We can reduce this case back to *(i)*. Specifically, for $i \in [k-1]$ we define $\gamma_i = (i - \frac{k}{2})\Delta$ (where $\Delta = \frac{1}{2} \cdot \frac{1}{k+1}$) and use the algorithm from § 3.1.3 with the odd number $k-1$ in place of k. Note that $\beta \leq \alpha - \Delta$ ensures $\beta \leq \frac{k}{2}\Delta$ (since $\alpha + \beta = \frac{1}{2}$) and thus $\beta - \gamma_{k-1} \leq \Delta$ and $\gamma_1 - (-\beta) \leq \Delta$, which is all that is needed for the analysis to go through. Thus we can achieve advantage $\frac{q}{k-1}$, which is even better than $\frac{q}{k}$.

3.2 Separations

The relativized separations follow from the corresponding decision tree complexity separations:

(iii) If q, k are even: $\mathsf{BPP}_{1/(k-1)}^{\mathsf{NP}[1]\mathsf{dt}} \not\subseteq \mathsf{BPP}_{>q/k}^{\mathsf{NP}\|[q]\mathsf{dt}}$.

(iv) If q is odd: $\mathsf{BPP}_{1/k}^{\mathsf{NP}[1]\mathsf{dt}} \not\subseteq \mathsf{BPP}_{>q/k}^{\mathsf{NP}\|[q]\mathsf{dt}}$.

We prove *(iii)* in § 3.2.1 and *(iv)* in § 3.2.2; the arguments are similar in structure. Our proof of *(iv)* also works if q is even, but in that case the result is subsumed by *(iii)*. The case $q = 1$, $k = 2$ of *(iv)* was proven in [Wat20], but our proof is somewhat different even specialized to that case.

For completeness, in § 3.2.3 we explain the standard argument for translating these decision tree separations into relativized separations for the corresponding time-bounded complexity classes. See [Ver99] for a general discussion of this phenomenon.

Let wt($\cdot$) refer to Hamming weight. Henceforth fix the constants q and k, and assume $q < k$ since otherwise there is nothing to prove.

3.2.1 Proof of *(iii)*

Define the partial function $f\colon \{0,1\}^n \to \{0,1\}$ that interprets its input as $(x,y) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$, such that

$$f(x,y) \;=\; \begin{cases} 1 & \text{if } \mathrm{wt}(x) = \mathrm{wt}(y) + 1 \le \frac{k}{2} \\ 0 & \text{if } \mathrm{wt}(x) = \mathrm{wt}(y) \le \frac{k}{2} - 1 \end{cases}.$$

Lemma 1. $\mathsf{BPP}^{\mathsf{NP}[1]\mathsf{dt}}_{1/(k-1)}(f) \le \frac{k}{2}$.

Lemma 2. $\mathsf{BPP}^{\mathsf{NP}\|[q]\mathsf{dt}}_{q/k+\delta}(f) \ge \Omega(\delta n)$ *for every* $\delta(n)$.

The separation follows by taking $\delta = \log^{-c} n$ for any constant c.

Proof of Lemma 1. Given (x,y), pick one of these $k-1$ possibilities uniformly at random:

- for some $i \in [\frac{k}{2}]$: output 1 iff $\mathrm{wt}(x) \ge i$,
- for some $i \in [\frac{k}{2} - 1]$: output 0 iff $\mathrm{wt}(y) \ge i$.

The decision tree does not directly query any bits of (x,y), and the DNF has width $i \le \frac{k}{2}$ (it is the OR over all i-subsets of either x's bits or y's bits, of the AND of those bits), so the cost is $\frac{k}{2}$. If $f(x,y) = 1$ with $\mathrm{wt}(x) = j$ and $\mathrm{wt}(y) = j-1$, then the probability of outputting 1 is $\frac{j+((k/2-1)-(j-1))}{k-1} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{k-1}$ since conditioned on picking x, the output is 1 iff $i \le j$, and conditioned on picking y, the output is 1 iff $i \ge j$. Similarly, if $f(x,y) = 0$ with $\mathrm{wt}(x) = \mathrm{wt}(y) = j$, then the probability of outputting 1 is $\frac{j+((k/2-1)-j)}{k-1} = \frac{1}{2} - \frac{1}{2} \cdot \frac{1}{k-1}$. □

Proof of Lemma 2. By the minimax principle, it suffices to show that for some distribution on valid inputs (x,y) to f, every cost-$o(\delta n)$ $\mathsf{P}^{\mathsf{NP}\|[q]}$-type decision tree T has advantage $< \frac{q}{k} + \delta$ over a random input. Let $T(x,y)$ denote the output produced after T receives the answers to its DNF queries. Let u be the leaf reached after seeing only 0's, and say u is labeled with DNFs $(\varphi_1, \ldots, \varphi_q)$ and function $\mathit{out}\colon \{0,1\}^q \to \{0,1\}$ (so if (x,y) leads to u then $T(x,y) = \mathit{out}(\varphi_1(x,y), \ldots, \varphi_q(x,y))$).

We generate the distribution on valid inputs (x,y) as follows. Let $v^0 = w^0 \in \{0,1\}^{n/2}$ be the all-0 string, and for $i = 1, \ldots, \frac{k}{2}$ obtain v^i by flipping a uniformly random 0 of v^{i-1} to a 1, and for $i = 1, \ldots, \frac{k}{2} - 1$ obtain w^i by flipping a uniformly random 0 of w^{i-1} to a 1. Pick a uniformly random $j \in [\frac{k}{2}]$, and then let (x,y) be either the 1-input (v^j, w^{j-1}) or the 0-input (v^{j-1}, w^{j-1}) with probability $\frac{1}{2}$ each.

Let v denote $(v^0, \ldots, v^{k/2})$ and w denote $(w^0, \ldots, w^{k/2-1})$, and call (v,w) *good* iff:

- for each $j \in [\frac{k}{2}]$: both inputs (v^j, w^{j-1}) and (v^{j-1}, w^{j-1}) lead to u, and
- for each $j \in [\frac{k}{2}]$ and each $i \in [q]$: $\varphi_i(v^j, w^{j-1}) \ge \varphi_i(v^{j-1}, w^{j-1}) \ge \varphi_i(v^{j-1}, w^{j-2})$
  (the latter inequality is only required if $j > 1$).

We claim that

(1) $\mathbb{P}[(v,w) \text{ is bad}] < \frac{\delta}{2}$, and
(2) $\mathbb{P}\big[T(x,y) = f(x,y) \,\big|\, (v,w) \text{ is good}\big] \le \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$,

from which it follows that

$$\mathbb{P}[T(x,y) = f(x,y)] \;\le\; \mathbb{P}\big[T(x,y) = f(x,y) \,\big|\, (v,w) \text{ is good}\big] + \mathbb{P}[(v,w) \text{ is bad}] \;<\; \tfrac{1}{2} + \tfrac{1}{2}(\tfrac{q}{k} + \delta).$$

We argue claim *(1)*. Since the path to u queries $o(\delta n)$ locations, with probability $\geq 1 - o(k\delta) > 1 - \frac{\delta}{4}$ each of the 1's placed throughout v and w avoids these locations, in which case the first bullet holds in the definition of good. Fixing j and i in the second bullet, if we condition on $\varphi_i(v^{j-1}, w^{j-1}) = 1$ and choose an arbitrary term of φ_i that accepts (v^{j-1}, w^{j-1}), then since the term has width $o(\delta n)$, with probability $\geq 1 - o(\delta)$ the 1 that is placed to obtain v^j from v^{j-1} avoids this term, in which case the term continues to accept (v^j, w^{j-1}) and so $\varphi_i(v^j, w^{j-1}) = 1$. Thus $\mathbb{P}\big[\varphi_i(v^j, w^{j-1}) \geq \varphi_i(v^{j-1}, w^{j-1})\big] \geq \mathbb{P}\big[\varphi_i(v^j, w^{j-1}) = 1 \,\big|\, \varphi_i(v^{j-1}, w^{j-1}) = 1\big] \geq 1 - o(\delta)$. Similarly, $\mathbb{P}\big[\varphi_i(v^{j-1}, w^{j-1}) \geq \varphi_i(v^{j-1}, w^{j-2})\big] \geq 1 - o(\delta)$. A union bound over j and i shows that the second bullet holds with probability $\geq 1 - o(kq\delta) > 1 - \frac{\delta}{4}$, so finally the two bullets hold simultaneously with probability $> 1 - \frac{\delta}{2}$.

We argue claim *(2)*. Condition on any particular good (v, w). We abbreviate the q-tuple $(\varphi_1(x, y), \ldots, \varphi_q(x, y))$ as $\varphi(x, y) \in \{0,1\}^q$. Consider the sequence of k inputs $(v^0, w^0), (v^1, w^0), (v^1, w^1), (v^2, w^1), \ldots$ (like climbing a ladder but placing both feet on each rung). Each of these possibilities for (x, y) leads to u and thus $T(x, y) = \mathit{out}(\varphi(x, y))$. Also, the corresponding sequence of $\varphi(x, y)$'s is monotonically nondecreasing in each of the q coordinates. Thus the sequence of inputs can be partitioned into segments of lengths say $\ell_0, \ell_1, \ldots, \ell_q$ (which sum to k) such that for the first ℓ_0 (x, y)'s in the sequence, $\varphi(x, y)$ has weight 0 (hence $T(x, y)$ is the same), and for the next ℓ_1 (x, y)'s in the sequence, $\varphi(x, y)$ is the same weight-1 string (hence $T(x, y)$ is the same), and so on. Since each segment alternates between 0-inputs and 1-inputs of f, we have $T(x, y) = f(x, y)$ for at most $\lceil \frac{\ell_i}{2} \rceil \leq \frac{\ell_i + 1}{2}$ inputs in the i^{th} segment.

Thus, out of the k possibilities for (x, y) given (v, w), at most $\sum_{i=0}^{q} \frac{\ell_i + 1}{2} = \frac{k}{2} + \frac{q+1}{2}$ are such that $T(x, y) = f(x, y)$. This implies that $\mathbb{P}\big[T(x, y) = f(x, y) \,\big|\, (v, w) \text{ is good}\big] \leq \frac{1}{2} + \frac{1}{2} \cdot \frac{q+1}{k}$, which is almost what we want. This issue can be fixed by observing that since k is even and $q + 1$ (the number of segments) is odd, at least one segment must have even length, in which case $\lceil \frac{\ell_i}{2} \rceil = \frac{\ell_i}{2}$. Thus, out of the k possibilities for (x, y) given (v, w), $T(x, y) = f(x, y)$ holds for at most $\frac{k}{2} + \frac{q}{2}$ of them, which gives *(2)*. □

3.2.2 Proof of *(iv)*

Define the partial function $f\colon \{0,1\}^n \to \{0,1\}$ that interprets its input as $(x, y) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$, such that

$$f(x, y) \;=\; \begin{cases} 1 & \text{if } \operatorname{wt}(x) = \operatorname{wt}(y) + 1 \leq k \\ 0 & \text{if } \operatorname{wt}(y) = \operatorname{wt}(x) + 1 \leq k \end{cases}.$$

Lemma 3. $\mathsf{BPP}^{\mathsf{NP}[1]\mathsf{dt}}_{1/k}(f) \leq k$.

Lemma 4. $\mathsf{BPP}^{\mathsf{NP}\|[q]\mathsf{dt}}_{q/k+\delta}(f) \geq \Omega(\delta n)$ *for every* $\delta(n)$.

The separation follows by taking $\delta = \log^{-c} n$ for any constant c.

Proof of Lemma 3. Given (x, y), pick one of these $2k$ possibilities uniformly at random:

- for some $i \in [k]$: output 1 iff $\operatorname{wt}(x) \geq i$,
- for some $i \in [k]$: output 0 iff $\operatorname{wt}(y) \geq i$.

The decision tree does not directly query any bits of (x, y), and the DNF has width $i \leq k$ (it is the OR over all i-subsets of either x's bits or y's bits, of the AND of those bits), so the cost is k. If $f(x, y) = 1$ with $\operatorname{wt}(x) = j$ and $\operatorname{wt}(y) = j - 1$, then the probability of outputting 1 is

$\frac{j+(k-(j-1))}{2k} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{k}$ since conditioned on picking x, the output is 1 iff $i \leq j$, and conditioned on picking y, the output is 1 iff $i \geq j$. The correctness argument is analogous if $f(x,y) = 0$. □

Proof of Lemma 4. By the minimax principle, it suffices to show that for some distribution on valid inputs (x,y) to f, every cost-$o(\delta n)$ $\mathsf{P}^{\mathsf{NP}\|[q]}$-type decision tree T has advantage $< \frac{q}{k} + \delta$ over a random input. Let $T(x,y)$ denote the output produced after T receives the answers to its DNF queries. Let u be the leaf reached after seeing only 0's, and say u is labeled with DNFs $(\varphi_1, \ldots, \varphi_q)$ and function *out*: $\{0,1\}^q \to \{0,1\}$ (so if (x,y) leads to u then $T(x,y) = \mathit{out}(\varphi_1(x,y), \ldots, \varphi_q(x,y))$).

We generate the distribution on valid inputs (x,y) as follows. Let $v^0 = w^0 \in \{0,1\}^{n/2}$ be the all-0 string, and for $i = 1, \ldots, k$ obtain v^i by flipping a uniformly random 0 of v^{i-1} to a 1, and obtain w^i by flipping a uniformly random 0 of w^{i-1} to a 1. Pick a uniformly random $j \in [k]$, and then let (x,y) be either the 1-input (v^j, w^{j-1}) or the 0-input (v^{j-1}, w^j) with probability $\frac{1}{2}$ each.

Let v denote $(v^0, \ldots, v^k)$ and w denote $(w^0, \ldots, w^k)$, and call (v,w) *good* iff:

- for each $j \in [k]$: both inputs (v^j, w^{j-1}) and (v^{j-1}, w^j) lead to u, and
- for each $j \in [k-1]$ and each $i \in [q]$: $\varphi_i(v^j, w^{j+1}) \geq \varphi_i(v^j, w^{j-1})$ and $\varphi_i(v^{j+1}, w^j) \geq \varphi_i(v^{j-1}, w^j)$.

We claim that

(1) $\mathbb{P}[(v,w) \text{ is bad}] < \frac{\delta}{2}$, and
(2) $\mathbb{P}\big[T(x,y) = f(x,y) \,\big|\, (v,w) \text{ is good}\big] \leq \frac{1}{2} + \frac{1}{2} \cdot \frac{q}{k}$,

from which it follows that

$$\mathbb{P}[T(x,y) = f(x,y)] \;\leq\; \mathbb{P}\big[T(x,y) = f(x,y) \,\big|\, (v,w) \text{ is good}\big] + \mathbb{P}[(v,w) \text{ is bad}] \;<\; \tfrac{1}{2} + \tfrac{1}{2}(\tfrac{q}{k} + \delta).$$

We argue claim *(1)*. Since the path to u queries $o(\delta n)$ locations, with probability $\geq 1 - o(k\delta) > 1 - \frac{\delta}{4}$ each of the 1's placed throughout v and w avoids these locations, in which case the first bullet holds in the definition of good. Fixing j and i in the second bullet, if we condition on $\varphi_i(v^j, w^{j-1}) = 1$ and choose an arbitrary term of φ_i that accepts (v^j, w^{j-1}), then since the term has width $o(\delta n)$, with probability $\geq 1 - o(\delta)$ both of the 1's placed to obtain w^{j+1} from w^{j-1} avoid this term, in which case the term continues to accept (v^j, w^{j+1}) and so $\varphi_i(v^j, w^{j+1}) = 1$. Thus $\mathbb{P}\big[\varphi_i(v^j, w^{j+1}) \geq \varphi_i(v^j, w^{j-1})\big] \geq \mathbb{P}\big[\varphi_i(v^j, w^{j+1}) = 1 \,\big|\, \varphi_i(v^j, w^{j-1}) = 1\big] \geq 1 - o(\delta)$. Similarly, $\mathbb{P}\big[\varphi_i(v^{j+1}, w^j) \geq \varphi_i(v^{j-1}, w^j)\big] \geq 1 - o(\delta)$. A union bound over j and i shows that the second bullet holds with probability $\geq 1 - o(kq\delta) > 1 - \frac{\delta}{4}$, so finally the two bullets hold simultaneously with probability $> 1 - \frac{\delta}{2}$.

We argue claim *(2)*. Condition on any particular good (v,w). We abbreviate the q-tuple $(\varphi_1(x,y), \ldots, \varphi_q(x,y))$ as $\varphi(x,y) \in \{0,1\}^q$. Consider the sequence of k inputs $(v^1, w^0), (v^1, w^2), (v^3, w^2), (v^3, w^4), \ldots$ (climbing the ladder starting with the left foot). Each of these possibilities for (x,y) leads to u and thus $T(x,y) = \mathit{out}(\varphi(x,y))$. Also, the corresponding sequence of $\varphi(x,y)$'s is monotonically nondecreasing in each of the q coordinates. Thus the sequence of inputs can be partitioned into segments of lengths say $\ell_0, \ell_1, \ldots, \ell_q$ (which sum to k) such that for the first ℓ_0 (x,y)'s in the sequence, $\varphi(x,y)$ has weight 0 (hence $T(x,y)$ is the same), and for the next ℓ_1 (x,y)'s in the sequence, $\varphi(x,y)$ is the same weight-1 string (hence $T(x,y)$ is the same), and so on. Since each segment alternates between 0-inputs and 1-inputs of f, we have $T(x,y) = f(x,y)$ for at most $\lceil \frac{\ell_i}{2} \rceil \leq \frac{\ell_i + 1}{2}$ inputs in the i^{th} segment.

Similarly, the sequence of k inputs $(v^0, w^1), (v^2, w^1), (v^2, w^3), (v^4, w^3), \ldots$ (climbing the ladder starting with the right foot) can be partitioned into segments of lengths say $\ell'_0, \ell'_1, \ldots, \ell'_q$ such that

$T(x,y) = f(x,y)$ for at most $\frac{\ell'_i+1}{2}$ inputs in the i^{th} segment. Thus, out of the $2k$ possibilities for (x,y) given (v,w), at most $\sum_{i=0}^{q}\bigl(\frac{\ell_i+1}{2}+\frac{\ell'_i+1}{2}\bigr) = k+q+1$ are such that $T(x,y) = f(x,y)$. This implies that $\mathbb{P}\bigl[T(x,y) = f(x,y) \bigm| (v,w) \text{ is good}\bigr] \le \frac{1}{2} + \frac{1}{2}\cdot\frac{q+1}{k}$, which is almost what we want.

This issue can be fixed using the following observation. Since there is only one string in $\{0,1\}^q$ of weight 0, $T(x,y)$ must actually be the same for all $\ell_0+\ell'_0$ inputs in the union of the 0^{th} segments from the two sequences. Since the number of 0-inputs and the number of 1-inputs in this union differ by at most 1, we have $T(x,y) = f(x,y)$ for at most $\frac{\ell_0+\ell'_0+1}{2}$ of these inputs. Now, out of the $2k$ possibilities for (x,y) given (v,w), $T(x,y) = f(x,y)$ holds for at most $\frac{\ell_0+\ell'_0+1}{2} + \sum_{i=1}^{q}\bigl(\frac{\ell_i+1}{2}+\frac{\ell'_i+1}{2}\bigr) = k+q+\frac{1}{2}$ of them. Since this count is an integer, it is in fact at most $k+q$, which gives *(2)*. (Alternatively, the $+\frac{1}{2}$ can be removed using a similar observation for the q^{th} segments.) □

3.2.3 Decision tree separations imply relativized separations

To illustrate this, we just consider the case $q = 1$, $k = 2$, but exactly the same approach works for all cases, as well as for the separations in Theorem 2 and Theorem 3.

We showed that $\mathsf{BPP}_{1/2}^{\mathsf{NP[1]dt}} \not\subseteq \mathsf{BPP}_{>1/2}^{\mathsf{NP[1]dt}}$. Now we explain how to construct an oracle language $O\colon \{0,1\}^* \to \{0,1\}$ such that $\bigl(\mathsf{BPP}_{1/2}^{\mathsf{NP[1]}}\bigr)^O \not\subseteq \bigl(\mathsf{BPP}_{>1/2}^{\mathsf{NP[1]}}\bigr)^O$. For all even N, let $f_N\colon \{0,1\}^N \to \{0,1\}$ be the partial function from § 3.2.2 with

$$f_N \in \mathsf{BPP}_{1/2}^{\mathsf{NP[1]dt}} \quad \text{and} \quad f_N \notin \mathsf{BPP}_{1/2+\log^{-c} N}^{\mathsf{NP[1]dt}} \text{ for every constant } c.$$

For any $O\colon \{0,1\}^* \to \{0,1\}$, let $O_n\colon \{0,1\}^n \to \{0,1\}$ be its restriction to input length n, and also interpret this truth table as a bit string $O_n \in \{0,1\}^N$ of length $N = 2^n$ indexed by the elements of $\{0,1\}^n$. Say that O is *valid* iff O_n is a valid input to f_N for every n. For any valid O, define the unary language $L_O\colon \{1\}^* \to \{0,1\}$ by $L_O(1^n) = f_N(O_n)$. We claim that

$$\forall O:\ L_O \in \bigl(\mathsf{BPP}_{1/2}^{\mathsf{NP[1]}}\bigr)^O \quad \text{and} \quad \exists O:\ L_O \notin \bigl(\mathsf{BPP}_{1/2+n^{-c}}^{\mathsf{NP[1]}}\bigr)^O \text{ for every constant } c$$

where the quantifiers are over valid O.

To see $L_O \in \bigl(\mathsf{BPP}_{1/2}^{\mathsf{NP[1]}}\bigr)^O$, note that an algorithm for L_O on input 1^n can run the $\mathsf{BPP}_{1/2}^{\mathsf{NP[1]}}$-type decision tree for f_N (from the proof of Lemma 3) on input O_n: Denoting the halves of O_n as $(x,y) \in \{0,1\}^{N/2} \times \{0,1\}^{N/2}$, pick one of these 4 possibilities uniformly at random:

- ask the NP^O oracle whether $\mathrm{wt}(x) \ge 1$, and output the same answer
- ask the NP^O oracle whether $\mathrm{wt}(x) \ge 2$, and output the same answer
- ask the NP^O oracle whether $\mathrm{wt}(y) \ge 1$, and output the opposite answer
- ask the NP^O oracle whether $\mathrm{wt}(y) \ge 2$, and output the opposite answer

To achieve $L_O \notin \bigl(\mathsf{BPP}_{1/2+n^{-c}}^{\mathsf{NP[1]}}\bigr)^O$ for every constant c, we design a valid O such that for every polynomial-time randomized algorithm M, every polynomial-time nondeterministic algorithm M', and every constant c, L_O is not solved with advantage $\frac{1}{2}+n^{-c}$ by running M with oracle access to O and one query to the language decided by M' with oracle access to O.

We enumerate the (M, M', c) triples in an arbitrary order, defining O_n for various input lengths n as we go along (finitely many at a time). For each (M, M', c), we select some n such that O_n has not been defined yet, and we use it to diagonalize against (M, M', c). For all $n' \ne n$ such that $O_{n'}$ has not been defined yet but running $M(1^n)$ with M' (for the NP^O oracle) might cause a query to a bit of $O_{n'}$, we define $O_{n'}$ to be an arbitrary valid input to $f_{N'}$ (where $N' = 2^{n'}$). Now when we run $M(1^n)$ with M', both algorithms have oracle access to the bits of O_n, and all other bits of O they might access have already been fixed.

We claim that if $M(1^n)$ with M' outputs $f_N(O_n)$ with advantage $\frac{1}{2}+n^{-c}$ for all valid O_n, then we can turn the computation into a $\mathsf{BPP}^{\mathsf{NP}[1]}_{1/2+n^{-c}}$-type decision tree for f_N: First the decision tree samples the same random string as M does. Then it adaptively queries bits of O_n as M does. Then when M produces z and *out*, the decision tree uses the same *out* and forms a DNF φ which evaluates $M'(z)$ as a function of O_n—for each possible witness, the computation of $M'(z)$ is a deterministic decision tree that queries bits of O_n, and it is a standard fact that the disjunction of these trees (over all possible witnesses) can be expressed as a DNF. (Each term in φ corresponds to a root-to-leaf path that outputs 1 in one of these trees. Each positive literal is a query M' makes to O_n that returns 1, and each negative literal is a query M' makes to O_n that returns 0.)

Since M and M' run in time poly(n), this $\mathsf{BPP}^{\mathsf{NP}[1]}_{1/2+n^{-c}}$-type decision tree would have cost polylog(N), but Lemma 4 says such a tree must have cost $\Omega(N/\log^c N)$, which is a contradiction if n is large enough. Thus there exists an O_n such that $M(1^n)$ with M' fails to compute $L_O(1^n) = f_N(O_n)$ with advantage $\frac{1}{2}+n^{-c}$. We fix this choice of O_n and move on to the next triple (M, M', c).

4 One-sided error

We now prove Theorem 2, restated here for convenience.

Theorem 2 (One-sided error, restated).

(i) $\mathsf{RP}^{\mathsf{NP}[1]}_{>1/2} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{1>}$.

(ii) $\mathsf{RP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{1/2} \cap \mathsf{RP}^{\mathsf{NP}\|[2]}_{1>}$.

(iii) $\mathsf{RP}^{\mathsf{NP}[1]}_{1/2} \not\subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{>1/2}$ *relative to an oracle.*

We prove the inclusions *(i)* and *(ii)* in § 4.1 and the separation *(iii)* in § 4.2.

4.1 Inclusions

We prove *(i)* in § 4.1.1 and *(ii)* in § 4.1.2.

4.1.1 Proof of *(i)*

For some constant c we have $L \in \mathsf{RP}^{\mathsf{NP}[1]}_{1/2+n^{-c}}$, witnessed by a polynomial-time randomized algorithm M (taking input x and coin tosses $s \in \{0,1\}^r$) and a language $L' \in \mathsf{NP}$. For an arbitrary constant d, we wish to show $L \in \mathsf{RP}^{\mathsf{NP}[1]}_{1-2^{-n^d}}$.

Fix an input x. The first step is to sample a sequence of $m = O(n^{2c+d})$ many independent strings $s^1, \ldots, s^m \in \{0,1\}^r$, so if $L(x) = 1$ then with probability $\geq 1 - 2^{-n^d}$, the sequence is *good* in the sense that on input x, M still has advantage strictly greater than $\frac{1}{2}$ when its coin tosses are chosen uniformly from the multiset $\{s^1, \ldots, s^m\}$. Then we design a polynomial-time deterministic algorithm which, given $s^1, \ldots, s^m$, makes one NP oracle query and outputs 1 if $L(x) = 1$ and $s^1, \ldots, s^m$ is good, and outputs 0 if $L(x) = 0$. Hence, over the random $s^1, \ldots, s^m$,

$$\mathbb{P}[\text{output is 1}] \begin{cases} \geq \mathbb{P}[s^1, \ldots, s^m \text{ is good}] \geq 1 - 2^{-n^d} & \text{if } L(x) = 1 \\ = 0 & \text{if } L(x) = 0 \end{cases}.$$

Henceforth fix a sequence $s^1, \dots, s^m$, and let z^h and $out^h \colon \{0,1\} \to \{0,1\}$ be the query string and truth table produced by $M_{s^h}(x)$ (so the output is $out^h(L'(z^h))$). We assume w.l.o.g. that out^h is nonconstant, and is hence either identity or negation.

If identity is more common among $out^1, \dots, out^m$, then our algorithm makes an NP oracle query to test whether there exists an h such that $out^h = \text{id}$ and $L'(z^h) = 1$, and outputs 1 if so and 0 otherwise. If $L(x) = 1$ and $s^1, \dots, s^m$ is good, then there must exist such an h (since the set of h's for which $M_{s^h}(x)$ outputs 1 has size $> \frac{m}{2}$ and so must intersect the set of h's for which $out^h = \text{id}$). If $L(x) = 0$ then there is no such h (since otherwise $M(x)$ would output 1 with positive probability).

If negation is at least as common as identity among $out^1, \dots, out^m$, then our algorithm makes an NP oracle query to test whether there *does not* exist an h such that $out^h = \text{neg}$ and $L'(z^h) = 0$ (a witness for the nonexistence of such an h consists of a witness for $L'(z^h) = 1$ for each h such that $out^h = \text{neg}$), and outputs 0 if so and 1 otherwise. If $L(x) = 1$ and $s^1, \dots, s^m$ is good, then there must exist such an h (since the set of h's for which $M_{s^h}(x)$ outputs 1 has size $> \frac{m}{2}$ and so must intersect the set of h's for which $out^h = \text{neg}$). If $L(x) = 0$ then there is no such h (since otherwise $M(x)$ would output 1 with positive probability).

4.1.2 Proof of *(ii)*

Let $q \in \{1, 2\}$. We show $\mathsf{RP}_{>0}^{\mathsf{NP}[1]} \subseteq \mathsf{RP}_{q/2>}^{\mathsf{NP}\|[q]}$ (the argument is very similar to *(i)*), then later we show how to strengthen the $q = 1$ case using a trick from [CP08].

For some constant c we have $L \in \mathsf{RP}_{n^{-c}}^{\mathsf{NP}[1]}$, witnessed by a polynomial-time randomized algorithm M (taking input x and coin tosses $s \in \{0,1\}^r$) and a language $L' \in \mathsf{NP}$. For an arbitrary constant d, we wish to show $L \in \mathsf{RP}_{q/2-2^{-n^d}}^{\mathsf{NP}\|[q]}$.

Fix an input x. The first step is to sample a sequence of $m = O(n^{c+d})$ many independent strings $s^1, \dots, s^m \in \{0,1\}^r$, so if $L(x) = 1$ then with probability $\geq 1 - 2^{-n^d}$, the sequence is *good* in the sense that M still has advantage strictly greater than 0 when its coin tosses are chosen uniformly from the multiset $\{s^1, \dots, s^m\}$. Then we design a polynomial-time randomized algorithm which, given $s^1, \dots, s^m$, makes q nonadaptive NP oracle queries and outputs 1 with probability $\geq \frac{q}{2}$ if $L(x) = 1$ and $s^1, \dots, s^m$ is good, and always outputs 0 if $L(x) = 0$. Hence, over the random $s^1, \dots, s^m$ and the other randomness of our algorithm,

$$\mathbb{P}[\text{output is 1}] \begin{cases} \geq \mathbb{P}[s^1, \dots, s^m \text{ is good}] \cdot \frac{q}{2} \geq \frac{q}{2} - 2^{-n^d} & \text{if } L(x) = 1 \\ = 0 & \text{if } L(x) = 0 \end{cases}.$$

Henceforth fix a sequence $s^1, \dots, s^m$, and let z^h and $out^h \colon \{0,1\} \to \{0,1\}$ be the query string and truth table produced by $M_{s^h}(x)$ (so the output is $out^h(L'(z^h))$). We assume w.l.o.g. that out^h is nonconstant, and is hence either identity or negation.

If $q = 2$ then our algorithm does the "id" NP oracle query ($\exists\, h : out^h = \text{id}$ and $L'(z^h) = 1\,?$) and the "neg" NP oracle query ($\neg\exists\, h : out^h = \text{neg}$ and $L'(z^h) = 0\,?$). These two queries tell us whether there exists an h for which $M_{s^h}(x)$ outputs 1 (which is the case if $L(x) = 1$ and $s^1, \dots, s^m$ is good), so we output 1 if so and 0 otherwise.

If $q = 1$ then our algorithm picks one of the two queries with probability $\frac{1}{2}$ each, and outputs 1 iff the result of that query indicates the existence of an h for which $M_{s^h}(x)$ outputs 1. If $L(x) = 1$ and $s^1, \dots, s^m$ is good, then at least one of the two queries will cause us to output 1.

To strengthen the $q = 1$ result to $\mathsf{RP}_{>0}^{\mathsf{NP}[1]} \subseteq \mathsf{RP}_{1/2}^{\mathsf{NP}[1]}$, suppose the bit length of witnesses for L' is n^b, and then use $d = b + 1$ and consider the following algorithm: Pick uniformly random $h \in [m]$

and $w \in \{0,1\}^{n^b}$; if $out^h = \text{id}$ and w witnesses $L'(z^h) = 1$, then output 1, otherwise do the "id" query with probability $\frac{1}{2} - 2^{-n^d}$ and do the "neg" query with probability $\frac{1}{2} + 2^{-n^d}$ (and output 1 iff the query indicates the existence of an h for which $M_{s^h}(x)$ outputs 1). If $L(x) = 0$, then this still always outputs 0. If $L(x) = 1$ and $s^1, \ldots, s^m$ is good, then at least one of the following holds.

- There is an h with $out^h = \text{id}$ and $L'(z^h) = 1$, in which case we find one with probability $\geq \frac{1}{m} \cdot 2^{-n^b} \geq 2^{-n^d+2}$ in the first phase, and thus output 1 with probability $\geq 2^{-n^d+2} + (1 - 2^{-n^d+2})(\frac{1}{2} - 2^{-n^d}) \geq \frac{1}{2} + 2^{-n^d}$ because of the "id" query.
- There is an h with $out^h = \text{neg}$ and $L'(z^h) = 0$, in which case we output 1 with probability $\geq \frac{1}{2} + 2^{-n^d}$ because of the "neg" query.

Either way, overall we have $\mathbb{P}[\text{output is 1}] \geq \mathbb{P}[s^1, \ldots, s^m \text{ is good}] \cdot (\frac{1}{2} + 2^{-n^d}) \geq \frac{1}{2}$ if $L(x) = 1$.

4.2 Separation: Proof of *(iii)*

We prove the corresponding decision tree complexity separation $\mathsf{RP}_{1/2}^{\mathsf{NP}[1]\mathsf{dt}} \not\subseteq \mathsf{RP}_{>1/2}^{\mathsf{NP}[1]\mathsf{dt}}$; the relativized separation follows routinely from this by the same approach as in § 3.2.3.

Let $\text{wt}(\cdot)$ refer to Hamming weight. Define the partial function $f\colon \{0,1\}^n \to \{0,1\}$ that interprets its input as $(x,y) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$, such that

$$f(x,y) \;=\; \begin{cases} 1 & \text{if } \text{wt}(x) = \text{wt}(y) \leq 1 \\ 0 & \text{if } \text{wt}(x) = 0 \text{ and } \text{wt}(y) = 1 \end{cases}.$$

Lemma 5. $\mathsf{RP}_{1/2}^{\mathsf{NP}[1]\mathsf{dt}}(f) \leq 1$.

Lemma 6. $\mathsf{RP}_{1/2+\delta}^{\mathsf{NP}[1]\mathsf{dt}}(f) \geq \Omega(\delta n)$ *for every* $\delta(n)$.

The separation follows by taking $\delta = \log^{-c} n$ for any constant c.

Proof of Lemma 5. Given (x,y):

- with probability $\frac{1}{2}$, output 1 iff $\text{wt}(x) \geq 1$,
- with probability $\frac{1}{2}$, output 0 iff $\text{wt}(y) \geq 1$.

This has cost 1 (since the OR function is a width-1 DNF), and it outputs 1 with probability $\frac{1}{2}$ if $f(x,y) = 1$ and with probability 0 if $f(x,y) = 0$. □

Proof of Lemma 6. By the minimax principle, it suffices to show that for some distribution on 1-inputs (x,y) to f, every cost-$o(\delta n)$ $\mathsf{P}^{\mathsf{NP}[1]}$-type decision tree T has either $\mathbb{P}[T(x,y) = 1] < \frac{1}{2} + \delta$ over this distribution or $T(x,y) = 1$ for some 0-input (x,y), where $T(x,y)$ denotes the output produced after T receives the answer to its DNF query. Let u be the leaf reached after seeing only 0's, and say u is labeled with DNF φ and function out: $\{0,1\} \to \{0,1\}$ (so if (x,y) leads to u then $T(x,y) = out(\varphi(x,y))$). W.l.o.g., out is nonconstant and φ contains no terms with multiple positive literals from x or from y, since such terms would never accept a valid input to f.

We generate the distribution on 1-inputs (x,y) as follows. With probability $\frac{1}{2}$ let $x = y = 0^{n/2}$, and with probability $\frac{1}{2}$ let x and y be independent uniformly random weight-1 strings. If $out = \text{id}$ then either φ has a term with no positive literals, in which case some 0-input leads to u and is accepted by φ, or every term has a positive literal, in which case 0^n leads to u and is rejected by φ

and so $\mathbb{P}[T(x,y)=1] \le \mathbb{P}[(x,y) \ne 0^n] = \frac{1}{2} < \frac{1}{2}+\delta$. Now assume $out =$ neg and there is no 0-input that leads to u and is rejected by φ. Note that if a 0-input $(0^{n/2}, y)$ leads to u and we choose an arbitrary term of φ that accepts $(0^{n/2}, y)$, then with probability $\ge 1 - o(\delta)$ the 1 that is placed in a uniformly random weight-1 x avoids both this term and all the bits queried on the path to u, in which case (x,y) continues to lead to u and be accepted by that term and hence by φ, so $T(x,y) = 0$. Thus,

$$\begin{aligned}\mathbb{P}[T(x,y)=1] &\le \tfrac{1}{2} + \tfrac{1}{2}\mathbb{P}\big[T(x,y)=1 \,\big|\, \mathrm{wt}(x)=\mathrm{wt}(y)=1\big] \\ &\le \tfrac{1}{2} + \tfrac{1}{2}\big(\mathbb{P}\big[(0^{n/2},y) \text{ does not lead to } u \,\big|\, \mathrm{wt}(y)=1\big] + \\ &\qquad \mathbb{P}\big[T(x,y)=1 \,\big|\, \mathrm{wt}(x)=\mathrm{wt}(y)=1 \text{ and } (0^{n/2},y) \text{ leads to } u\big]\big) \\ &\le \tfrac{1}{2} + \tfrac{1}{2}(o(\delta)+o(\delta)) \;<\; \tfrac{1}{2}+\delta. \qquad \square\end{aligned}$$

5 Zero-sided error

We now prove Theorem 3, restated here for convenience.

Theorem 3 (Zero-sided error, restated). *For integers $1 \le q \le k \le 4$:*

(i) If $k = 4$: $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$.

(ii) If $k \le 3$: $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$.

(iii) $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/k} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/k}$ *relative to an oracle.*

Moreover, the "$q/k>$" in the inclusion subscripts can be improved to "q/k" if $q < k$ and $k \ge 3$.

We prove the inclusions *(i)* and *(ii)* in § 5.1 and the separations *(iii)* in § 5.2.

5.1 Inclusions

Straightforwardly generalizing the proof of $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{ZPP}^{\mathsf{NP}[1]}_{1/4}$ in [CP08] yields *(i)*, but we take a different tack by showing in § 5.1.1 that *(i)* follows directly from Theorem 2. We prove *(ii)* from first principles in § 5.1.2; our proof for the case $k = 1$ is equivalent to the one in [CP08], but we include it for completeness.

5.1.1 Proof of *(i)*

Let $L \in \mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{>0}$. By Theorem 2 and closure of $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>0}$ under complement,

$L \in \mathsf{RP}^{\mathsf{NP}[1]}_{1/2}$ by some algorithm M^1, $\quad$ $L \in \mathsf{RP}^{\mathsf{NP}\|[2]}_{1>}$ by some algorithm M^2,

$\overline{L} \in \mathsf{RP}^{\mathsf{NP}[1]}_{1/2}$ by some algorithm $\overline{M}^1$, $\quad$ $\overline{L} \in \mathsf{RP}^{\mathsf{NP}\|[2]}_{1>}$ by some algorithm $\overline{M}^2$.

We let each of these four M-algorithms refer to the entire computation, including the NP oracle queries, which we elide for convenience. (Note that $\overline{M}^i$ does not mean "complement of M^i"—it is a different algorithm.) We assume M^2 and $\overline{M}^2$ have advantage $\ge 1 - 2^{-n^d}$ for an arbitrary constant d. Furthermore, we assume all four algorithms have been modified to output $\perp$ instead of 0, and $\overline{M}^1$ and $\overline{M}^2$ have been modified to output 0 instead of 1.

If $q = 1$: $L \in \mathsf{ZPP}^{\mathsf{NP}[1]}_{1/4}$ by running M^1 or $\overline{M}^1$ with probability $\frac{1}{2}$ each.

If $q = 2$: $L \in \mathsf{ZPP}^{\mathsf{NP}\|[2]}_{1/2}$ by running M^1 and $\overline{M}^1$, and if one of them outputs a bit, outputting that bit or $\perp$ otherwise.

If $q = 4$: $L \in \mathsf{ZPP}^{\mathsf{NP}\|[4]}_{1>}$ by running M^2 and $\overline{M}^2$, and if one of them outputs a bit, outputting that bit or $\perp$ otherwise.

If $q = 3$: $L \in \mathsf{ZPP}^{\mathsf{NP}\|[3]}_{3/4>}$ by running M^1 and $\overline{M}^2$ with probability $\frac{1}{2}$, or M^2 and $\overline{M}^1$ with probability $\frac{1}{2}$, and if one of them outputs a bit, outputting that bit or $\perp$ otherwise. This falls slightly short of our promise of showing $L \in \mathsf{ZPP}^{\mathsf{NP}\|[3]}_{3/4}$, but that can be fixed by noting that the proof of Theorem 2 actually shows that M^1 and $\overline{M}^1$ can have advantage $\geq \frac{1}{2} + 2^{-n^e}$ for some constant e depending on L. Then taking $d \geq e$ ensures we get advantage $\geq \frac{1}{2}\big(\frac{1}{2} + 2^{-n^e}\big) + \frac{1}{2}\big(1 - 2^{-n^d}\big) \geq \frac{3}{4}$.

5.1.2 Proof of *(ii)*

We just prove $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$; the "moreover" part follows by exactly the same trick (due to [CP08]) for strengthening $\mathsf{RP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{1/2>}$ to $\mathsf{RP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{RP}^{\mathsf{NP}[1]}_{1/2}$, which is described in § 4.1.2.

For some constant c we have $L \in \mathsf{ZPP}^{\mathsf{NP}[1]}_{1/(k+1)+n^{-c}}$, witnessed by a polynomial-time randomized algorithm M (taking input x and coin tosses $s \in \{0,1\}^r$) and a language $L' \in \mathsf{NP}$. For an arbitrary constant d, we wish to show $L \in \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k-2^{-n^d}}$.

Fix an input x. The first step is to sample a sequence of $m = O(n^{2c+d})$ many independent strings $s^1, \ldots, s^m \in \{0,1\}^r$, so with probability $\geq 1 - 2^{-n^d}$, the sequence is *good* in the sense that on input x, M still has advantage strictly greater than $\frac{1}{k+1}$ when its coin tosses are chosen uniformly from the multiset $\{s^1, \ldots, s^m\}$. Then we design a polynomial-time randomized algorithm which, given a good sequence, outputs $L(x)$ with probability $\geq \frac{q}{k}$ after making q nonadaptive NP oracle queries, and which has zero-sided error for all sequences (good and bad). Hence, over the random $s^1, \ldots, s^m$ and the other randomness of our algorithm,

$$\mathbb{P}[\text{output is } L(x)] \;\geq\; \mathbb{P}\big[\text{output is } L(x) \,\big|\, s^1, \ldots, s^m \text{ is good}\big] - \mathbb{P}[s^1, \ldots, s^m \text{ is bad}] \;\geq\; \tfrac{q}{k} - 2^{-n^d}.$$

Henceforth fix a good sequence $s^1, \ldots, s^m$, and let z^h and $out^h \colon \{0,1\} \to \{0,1,\perp\}$ be the query string and truth table produced by $M_{s^h}(x)$ (so the output is $out^h(L'(z^h))$). We assume w.l.o.g. that out^h is nonconstant. If there is an h such that $out^h \in \{\text{id}, \text{neg}\}$, then our algorithm simply uses the NP oracle to evaluate $L'(z^h)$ and then outputs $out^h(L'(z^h)) = L(x)$. Otherwise, each out^h is one of the four functions out_{ab} (for $ab \in \{0,1\}^2$) that maps a to b and $1 - a$ to $\perp$:

	out_{00}	out_{01}	out_{10}	out_{11}
0	0	1	$\perp$	$\perp$
1	$\perp$	$\perp$	0	1

Now $[m]$ is partitioned into four sets $H_{00} \cup H_{01} \cup H_{10} \cup H_{11}$ where $H_{ab} = \{h \in [m] : out^h = out_{ab}\}$. Let $H = \{h \in [m] : M_{s^h}(x) \text{ outputs } L(x)\}$ and note that $|H| > \frac{m}{k+1}$ by the assumption that $s^1, \ldots, s^m$ is good. If $h \in H \cap H_{ab}$ then $M_{s^h}(x)$ outputs b, so if we detect that $H \cap H_{ab} \neq \emptyset$ then we can safely output b. Note that $H \subseteq H_{0b} \cup H_{1b}$ for $b = L(x)$.

For each $ab \in \{0,1\}^2$ consider the "ab" query, which asks whether $H \cap H_{ab} \neq \emptyset$:

$$\exists\, h : \; out^h = out_{ab} \text{ and } L'(z^h) = a\,?$$

If $a = 1$ then the "ab" query can be expressed as an NP oracle query: a witness consists of an h with $out^h = out_{ab}$ and a witness for $L'(z^h) = 1$. If $a = 0$ then the "ab" query can be expressed as the negation of an NP oracle query: a witness for the nonexistence of such an h consists of a witness for $L'(z^h) = 1$ for each h such that $out^h = out_{ab}$. We say the "ab" query returns yes iff it indicates the existence of an $h \in H \cap H_{ab}$ (i.e., the NP oracle returns the bit a). If the "ab" query returns yes, we can safely output b since there exists an h such that $out^h(L'(z^h)) = out_{ab}(a) = b = L(x)$.

Our algorithm is:

1. Identify a set $P \subseteq \{0,1\}^2$ of size k for which there is guaranteed to exist an $ab \in P$ such that the "ab" query would return yes.
2. Pick a uniformly random $Q \subseteq P$ of size q.
3. For each $ab \in Q$ do the "ab" query and output b if it returns yes.
4. Finally output $\perp$ if all queries returned no.

This outputs $L(x)$ with probability $\geq \frac{q}{k}$. We just need to prove that we can indeed find such a P in step 1.

If $k = 3$: Let P contain all ab's except the one with the smallest H_{ab} (which has size $\leq \frac{m}{4}$), breaking ties arbitrarily. Then $H \cap H_{ab} \neq \emptyset$ for at least one $ab \in P$ assuming $|H| > \frac{m}{4}$.

If $k = 2$: If $|H_{00} \cup H_{10}| \leq \frac{m}{3}$ then $L(x) = 1$ assuming $|H| > \frac{m}{3}$, so we can let $P = \{01, 11\}$. Similarly, if $|H_{01} \cup H_{11}| \leq \frac{m}{3}$ then we can let $P = \{00, 10\}$. (Although we know $L(x)$ in these cases assuming $s^1, \ldots, s^m$ is good, we must still do queries to ensure zero-sided error if $s^1, \ldots, s^m$ is bad.) Otherwise, the smaller of H_{00}, H_{10} has size $\leq \frac{m}{3}$, and the smaller of H_{01}, H_{11} has size $\leq \frac{m}{3}$, so we can let P contain the two ab's corresponding to the larger of H_{00}, H_{10} and the larger of H_{01}, H_{11}, breaking ties arbitrarily.

If $k = 1$: If $|H_{00} \cup H_{10}| \leq \frac{m}{2}$ then $L(x) = 1$ assuming $|H| > \frac{m}{2}$, and furthermore the smaller of H_{01}, H_{11} has size $\leq \frac{m}{2}$, so we can let P contain the ab corresponding to the larger of H_{01}, H_{11}. Similarly, if $|H_{01} \cup H_{11}| < \frac{m}{2}$ then we can let P contain the ab corresponding to the larger of H_{00}, H_{10}.

5.2 Separations: Proof of *(iii)*

We prove the corresponding decision tree complexity separations $\mathsf{ZPP}^{\mathsf{NP[1]dt}}_{1/k} \not\subseteq \mathsf{ZPP}^{\mathsf{NP[1]dt}}_{>1/k}$; the relativized separations follow routinely from these by the same approach as in § 3.2.3.[4]

Henceforth fix the constant $k \in \{2, 3, 4\}$. Define the partial function $f \colon \{0,1\}^n \to \{0,1\}$ that interprets its input as $(a, b, x) \in \{0,1\}^{\sqrt{n}} \times \{0,1\}^{\sqrt{n}} \times \{0,1\}^{n-2\sqrt{n}}$, viewing x as a $\sqrt{n} \times (\sqrt{n} - 2)$ matrix and letting x_i be the i^{th} row, such that for $B \in \{0,1\}$,

$$f(a,b,x) = B \text{ if } out_{a_i b_i}(\textsc{or}(x_i)) \in \{B, \perp\} \text{ for all } i, \text{ and}$$
$$out_{a_i b_i}(\textsc{or}(x_i)) = B \text{ for at least } \tfrac{\sqrt{n}}{k} \text{ many } i\text{'s}$$

where $out_{a_i b_i}$ was defined in § 5.1.2.

Lemma 7. $\mathsf{ZPP}^{\mathsf{NP[1]dt}}_{1/k}(f) \leq 3$.

[4] For the $k = 2$ case of *(iii)*, the slightly weaker relativized separation $\mathsf{ZPP}^{\mathsf{NP[1]}}_{1/2>} \not\subseteq \mathsf{ZPP}^{\mathsf{NP[1]}}_{>1/2}$ follows from the facts that $\mathsf{AM} \cap \mathsf{coAM} \subseteq \mathsf{ZPP}^{\mathsf{NP[1]}}_{1/2>}$ and $\mathsf{ZPP}^{\mathsf{NP[1]}}_{>1/2} \subseteq \mathsf{PP}$ relativize [GPW18] and $\mathsf{AM} \cap \mathsf{coAM} \not\subseteq \mathsf{PP}$ relative to an oracle [Ver95].

Proof. Pick a uniformly random $i \in \left[\sqrt{n}\right]$, query the bits a_i and b_i and the DNF $\textsc{or}(x_i)$, and output $out_{a_ib_i}(\textsc{or}(x_i))$. The cost has a contribution of 2 from querying a_i and b_i, and 1 from the width of $\textsc{or}$. □

Lemma 8. $\mathsf{ZPP}^{\mathsf{NP[1]dt}}_{1/k+\delta}(f) \geq \Omega(\delta\sqrt{n})$ *for every* $\delta(n)$.

The separation follows by taking $\delta = \log^{-c} n$ for any constant c.

We prove Lemma 8 for the rest of this section. By the minimax principle, it suffices to show that for some distribution on valid inputs (a,b,x) to f, every cost-$o(\delta\sqrt{n})$ $\mathsf{P}^{\mathsf{NP[1]}}$-type decision tree T has either $\mathbb{P}[T(a,b,x) = f(a,b,x)] < \frac{1}{k} + \delta$ over this distribution or $T(a,b,x) \notin \{f(a,b,x), \perp\}$ for some valid input (a,b,x), where $T(a,b,x)$ denotes the output produced after T receives the answer to its DNF query.

For a leaf u, say u is labeled with DNF φ^u and function $out^u \colon \{0,1\} \to \{0,1,\perp\}$ (so if (a,b,x) leads to u then $T(a,b,x) = out^u(\varphi^u(a,b,x))$). W.l.o.g., out^u is nonconstant, and no term of φ^u is violated by the bits read along the path to u, and if the path to u reads any bit from a_i, b_i, x_i then it reads both a_i and b_i, and if any term of φ^u has a literal using a variable from a_i, b_i, x_i then that term has literals using both a_i and b_i (at most tripling the cost of T). We call a leaf u *blind* iff the path to u reads no 1's from x.

Claim 1. *If there exists a blind leaf u such that out^u is identity or negation, then $T(a,b,x) \notin \{f(a,b,x), \perp\}$ for some valid input (a,b,x).*

Proof. We show that if u is blind then there exists an (a,b,x) that leads to u such that $\varphi^u(a,b,x) \neq f(a,b,x)$, which proves the claim for identity. By symmetry, interchanging the roles of 0 and 1 proves the claim for negation.

If every term of φ^u contains $a_i \wedge b_i \wedge x_{ij}$ for some i and j, then construct the following input: For each i:

- If the path to u reads $a_ib_i \in \{10, 01, 11\}$ then let a_ib_i be these bits, and let x_i be all-0's.
- If the path to u reads $a_ib_i = 00$ then let a_ib_i be these bits, and let x_i be all-0's except for a 1 in a location not read on the path to u.
- If the path to u does not read a_ib_i (or any bit of x_i) then let $a_ib_i = 01$, and let x_i be all-0's.

This (a,b,x) leads to u (since u is blind) and $\varphi^u(a,b,x) = 0$ and $f(a,b,x) = 1$ (since the path to u touches $o(\delta\sqrt{n})$ many i's and hence $out_{a_ib_i}(\textsc{or}(x_i)) = 1$ for $(1 - o(\delta))\sqrt{n} \geq \frac{\sqrt{n}}{k}$ many i's, namely at least those with $a_ib_i = 01$).

Otherwise, there exists a term C of φ^u such that for every i, if C contains $a_i \wedge b_i$ then it does not contain x_{ij} for any j. Then construct the following input: For each i:

- If C contains $a_i \wedge \overline{b_i}$ or $\overline{a_i} \wedge b_i$ or $\overline{a_i} \wedge \overline{b_i}$ then let a_ib_i and any x_{ij} variables mentioned in C be set consistent with satisfying C, and let all other bits of x_i be 0's except for a 1 in a location not read on the path to u and not mentioned in C (though the latter is not necessary if C already contains a positive x_{ij} literal).
- If the path to u reads $a_ib_i \in \{10, 01, 00\}$ but the previous case does not hold, then let a_ib_i be these bits, and let x_i be all-0's except for a 1 in a location not read on the path to u.
- If C contains $a_i \wedge b_i$ or the path to u reads $a_ib_i = 11$ then let $a_ib_i = 11$, and let x_i be all-0's.
- If neither C nor the path to u mentions/reads a_ib_i (or any bit of x_i) then let $a_ib_i = 10$, and let x_i have a 1 in any location.

This (a,b,x) leads to u (since u is blind) and $\varphi^u(a,b,x) = 1$ (since C is satisfied) and $f(a,b,x) = 0$ (since C and the path to u touch $o(\delta\sqrt{n})$ many i's and hence $out_{a_ib_i}(\textsc{or}(x_i)) = 0$ for $(1 - o(\delta))\sqrt{n} \geq \frac{\sqrt{n}}{k}$ many i's, namely at least those with $a_ib_i = 10$). □

Henceforth assume $T(a,b,x) \in \{f(a,b,x), \bot\}$ for all valid inputs (a,b,x), so by Claim 1, $\mathit{out}^u \in \{\mathit{out}_{00}, \mathit{out}_{01}, \mathit{out}_{10}, \mathit{out}_{11}\}$ if u is a blind leaf.

If $\boldsymbol{k = 4}$: We generate the distribution on valid inputs (a,b,x) as follows. With probability 1, let $a_i b_i = 00$ for the first $\frac{\sqrt{n}}{4}$ i's, $a_i b_i = 01$ for the next $\frac{\sqrt{n}}{4}$ i's, $a_i b_i = 10$ for the next $\frac{\sqrt{n}}{4}$ i's, and $a_i b_i = 11$ for the last $\frac{\sqrt{n}}{4}$ i's, and let $x^{00}, x^{01}, x^{10}, x^{11} \in \{0,1\}^{(\sqrt{n}/4)\times(\sqrt{n}-2)}$ refer to the corresponding groups of rows of x. Define $w^{00}, w^{01}, w^{10}, w^{11} \in \{0,1\}^{(\sqrt{n}/4)\times(\sqrt{n}-2)}$ by letting each row independently have a single 1 in a uniformly random column, and define $\hat{0}$ as the $\frac{\sqrt{n}}{4} \times (\sqrt{n}-2)$ all-0 matrix. With probability $\frac{1}{4}$ each, let x be one of:

$$\begin{array}{llll} \hat{0}\ \ w^{01}\ \hat{0}\ \hat{0} & (\text{so } f(a,b,x) = 0), & w^{00}\ w^{01}\ w^{10}\ \ \hat{0} & (\text{so } f(a,b,x) = 0), \\ w^{00}\ \ \hat{0}\ \ \hat{0}\ \hat{0} & (\text{so } f(a,b,x) = 1), & w^{00}\ w^{01}\ \ \hat{0}\ \ w^{11} & (\text{so } f(a,b,x) = 1). \end{array}$$

Let u denote the blind leaf reached after seeing only 0's in x and seeing bits of a and b fixed as in our distribution, and let $\varphi = \varphi^u$ and $\mathit{out} = \mathit{out}^u$. Let w denote $(w^{00}, w^{01}, w^{10}, w^{11})$, and call w *good* iff:

- for each of the four possibilities of x, (a,b,x) leads to u, and
- $\varphi\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big) \geq \varphi\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big)$ and $\varphi\big(a,b,w^{00}\,w^{01}\,w^{10}\,\hat{0}\big) \geq \varphi\big(a,b,w^{00}\,\hat{0}\,\hat{0}\,\hat{0}\big)$.

We claim that

(1) $\mathbb{P}[w \text{ is bad}] < \delta$, and
(2) $\mathbb{P}\big[T(a,b,x) = f(a,b,x) \,\big|\, w \text{ is good}\big] \leq \frac{1}{4}$,

from which it follows that

$$\mathbb{P}[T(a,b,x) = f(a,b,x)] \;\leq\; \mathbb{P}\big[T(a,b,x) = f(a,b,x) \,\big|\, w \text{ is good}\big] + \mathbb{P}[w \text{ is bad}] \;<\; \tfrac{1}{4} + \delta.$$

We argue claim *(1)*. Since the path to u queries $o(\delta\sqrt{n})$ locations of x, each of which has a $\frac{1}{\sqrt{n}-2}$ probability of having a 1 in w, by a union bound with probability $\geq 1 - o(\delta) > 1 - \frac{\delta}{2}$ each of the 1's placed throughout w avoids these locations, in which case the first bullet holds in the definition of good. For the second bullet, if we condition on $\varphi\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big) = 1$ and choose an arbitrary term of φ that accepts $\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big)$, then since the term has width $o(\delta\sqrt{n})$, with probability $\geq 1 - o(\delta)$ all the 1's in w^{00} and w^{11} avoid this term, in which case the term continues to accept $\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big)$ and so $\varphi\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big) = 1$. Thus the first part of the second bullet, and similarly also the second part, holds with probability $\geq 1 - o(\delta) > 1 - \frac{\delta}{4}$. By a union bound, the second bullet holds with probability $> 1 - \frac{\delta}{2}$, so finally the two bullets hold simultaneously with probability $> 1 - \delta$.

We argue claim *(2)*. Condition on any particular good w. For each of the four possibilities of x, $\mathit{out}(\varphi(a,b,x)) = T(a,b,x) \in \{f(a,b,x), \bot\}$.

- If $\mathit{out} = \mathit{out}_{00}$ then $T(a,b,x) = \bot$ for both 1-inputs, and $T\big(a,b,w^{00}\,w^{01}\,w^{10}\,\hat{0}\big) = \bot$ also since otherwise $\varphi\big(a,b,w^{00}\,\hat{0}\,\hat{0}\,\hat{0}\big) \leq \varphi\big(a,b,w^{00}\,w^{01}\,w^{10}\,\hat{0}\big) = 0$, in which case $T\big(a,b,w^{00}\,\hat{0}\,\hat{0}\,\hat{0}\big) = 0 \neq 1 = f\big(a,b,w^{00}\,\hat{0}\,\hat{0}\,\hat{0}\big)$.
- If $\mathit{out} = \mathit{out}_{01}$ then $T(a,b,x) = \bot$ for both 0-inputs, and $T\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big) = \bot$ also since otherwise $\varphi\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big) \leq \varphi\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big) = 0$, in which case $T\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big) = 1 \neq 0 = f\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big)$.

- If $out = out_{10}$ then $T(a,b,x) = \perp$ for both 1-inputs, and $T\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big) = \perp$ also since otherwise $\varphi\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big) \geq \varphi\big(a,b,\hat{0}\,w^{01}\,\hat{0}\,\hat{0}\big) = 1$, in which case $T\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big) = 0 \neq 1 = f\big(a,b,w^{00}\,w^{01}\,\hat{0}\,w^{11}\big)$.
- If $out = out_{11}$ then $T(a,b,x) = \perp$ for both 0-inputs, and $T\big(a,b,w^{00}\,\hat{0}\,\hat{0}\,\hat{0}\big) = \perp$ also since otherwise $\varphi\big(a,b,w^{00}\,w^{01}\,w^{10}\,\hat{0}\big) \geq \varphi\big(a,b,w^{00}\,\hat{0}\,\hat{0}\,\hat{0}\big) = 1$, in which case $T\big(a,b,w^{00}\,w^{01}\,w^{10}\,\hat{0}\big) = 1 \neq 0 = f\big(a,b,w^{00}\,w^{01}\,w^{10}\,\hat{0}\big)$.

If $\boldsymbol{k = 3}$: We generate the distribution on valid inputs (a,b,x) as follows. With probability 1, let $a_ib_i = 00$ for the first $\frac{\sqrt{n}}{3}$ i's, $a_ib_i = 01$ for the next $\frac{\sqrt{n}}{3}$ i's, and $a_ib_i = 10$ for the last $\frac{\sqrt{n}}{3}$ i's, and let $x^{00}, x^{01}, x^{10} \in \{0,1\}^{(\sqrt{n}/3)\times(\sqrt{n}-2)}$ refer to the corresponding groups of rows of x. Define $w^{00}, w^{01}, w^{10} \in \{0,1\}^{(\sqrt{n}/3)\times(\sqrt{n}-2)}$ by letting each row independently have a single 1 in a uniformly random column, and define $\hat{0}$ as the $\frac{\sqrt{n}}{3} \times (\sqrt{n}-2)$ all-0 matrix. With probability $\frac{1}{3}$ each, let x be one of:

$$\hat{0}\ w^{01}\ \hat{0}\ \ (\text{so } f(a,b,x)=0), \qquad w^{00}\ \hat{0}\ \hat{0}\ \ (\text{so } f(a,b,x)=1), \qquad w^{00}\ w^{01}\ w^{10}\ \ (\text{so } f(a,b,x)=0).$$

Let u denote the blind leaf reached after seeing only 0's in x and seeing bits of a and b fixed as in our distribution, and let $\varphi = \varphi^u$ and $out = out^u$. Let w denote (w^{00}, w^{01}, w^{10}), and call w *good* iff:

- for each of the three possibilities of x, (a,b,x) leads to u, and
- $\varphi\big(a,b,w^{00}\,w^{01}\,w^{10}\big) \geq \varphi\big(a,b,w^{00}\,\hat{0}\,\hat{0}\big)$.

We claim that

(1) $\mathbb{P}[w \text{ is bad}] < \delta$, and
(2) $\mathbb{P}\big[T(a,b,x) = f(a,b,x) \,\big|\, w \text{ is good}\big] \leq \frac{1}{3}$,

from which it follows that

$$\mathbb{P}[T(a,b,x) = f(a,b,x)] \ \leq\ \mathbb{P}\big[T(a,b,x) = f(a,b,x) \,\big|\, w \text{ is good}\big] + \mathbb{P}[w \text{ is bad}] \ <\ \tfrac{1}{3} + \delta.$$

The argument for claim *(1)* is essentially identical to the corresponding argument from the case $k = 4$, so we omit it.

We argue claim *(2)*. Condition on any particular good w. For each of the three possibilities of x, $out(\varphi(a,b,x)) = T(a,b,x) \in \{f(a,b,x), \perp\}$.

- If $out \in \{out_{01}, out_{11}\}$ then $T(a,b,x) = \perp$ for both 0-inputs.
- If $out = out_{00}$ then $T\big(a,b,w^{00}\,\hat{0}\,\hat{0}\big) = \perp$ and hence also $T\big(a,b,w^{00}\,w^{01}\,w^{10}\big) = \perp$ since $\varphi\big(a,b,w^{00}\,w^{01}\,w^{10}\big) \geq \varphi\big(a,b,w^{00}\,\hat{0}\,\hat{0}\big) = 1$.
- If $out = out_{10}$ then $T\big(a,b,w^{00}\,\hat{0}\,\hat{0}\big) = \perp$, and $T\big(a,b,\hat{0}\,w^{01}\,\hat{0}\big) = \perp$ also since otherwise an argument completely analogous to the proof of Claim 1 would show there exists a 1-input (a',b',x') that leads to u and $\varphi(a',b',x') \geq \varphi\big(a,b,\hat{0}\,w^{01}\,\hat{0}\big) = 1$, in which case $T(a',b',x') = 0$.

If $\boldsymbol{k = 2}$: We generate the distribution on valid inputs (a,b,x) as follows. With probability 1, let $a_ib_i = 00$ for the first $\frac{\sqrt{n}}{2}$ i's and $a_ib_i = 01$ for the last $\frac{\sqrt{n}}{2}$ i's, and let $x^{00}, x^{01} \in \{0,1\}^{(\sqrt{n}/2)\times(\sqrt{n}-2)}$ refer to the corresponding groups of rows of x. Define $w^{00}, w^{01} \in \{0,1\}^{(\sqrt{n}/2)\times(\sqrt{n}-2)}$ by letting each row independently have a single 1 in a uniformly random column, and define $\hat{0}$ as the $\frac{\sqrt{n}}{2} \times (\sqrt{n}-2)$ all-0 matrix. With probability $\frac{1}{2}$ each, let x be one of:

$$\hat{0}\ w^{01} \ \ (\text{so } f(a,b,x)=0), \qquad\qquad w^{00}\ \hat{0} \ \ (\text{so } f(a,b,x)=1).$$

Let u denote the blind leaf reached after seeing only 0's in x and seeing bits of a and b fixed as in our distribution, and let $\varphi = \varphi^u$ and $out = out^u$. Let w denote (w^{00}, w^{01}), and call w *good* iff for both possibilities of x, (a,b,x) leads to u. We claim that

(1) $\mathbb{P}[w \text{ is bad}] < \delta$, and
(2) $\mathbb{P}\big[T(a,b,x) = f(a,b,x) \,\big|\, w \text{ is good}\big] \le \frac{1}{2}$,

from which it follows that

$$\mathbb{P}[T(a,b,x) = f(a,b,x)] \ \le\ \mathbb{P}\big[T(a,b,x) = f(a,b,x) \,\big|\, w \text{ is good}\big] + \mathbb{P}[w \text{ is bad}] \ <\ \tfrac{1}{2} + \delta.$$

The argument for claim *(1)* is as in the $k=4$ and $k=3$ cases. For claim *(2)*, if $out \in \{out_{01}, out_{11}\}$ then $T\big(a,b,\hat{0}\,w^{01}\big) = \perp$, and if $out \in \{out_{00}, out_{10}\}$ then $T\big(a,b,w^{00}\,\hat{0}\big) = \perp$.

6 Open problems

For all integers $k \ge 1$, we proved that $\mathsf{BPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{BPP}^{\mathsf{NP}[1]}_{1/k>}$ and $\mathsf{BPP}^{\mathsf{NP}[1]}_{1/k} \not\subseteq \mathsf{BPP}^{\mathsf{NP}[1]}_{>1/k}$ relative to an oracle, but it remains open whether $\mathsf{BPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{BPP}^{\mathsf{NP}[1]}_{1/k}$, and we do not have a conjecture about whether this should hold.

We conjecture that the third bullet in Theorem 3 also holds for $q > 1$, which would mean all the inclusions are essentially tight, leading to the following ideal statement (which echoes the statement of Theorem 1).

Conjecture 1 (Zero-sided error). *For integers* $1 \le q \le k \le 4$*:*

- *If* $k \le 3$*:* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>1/(k+1)} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$ *and* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/k} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{>q/k}$ *relative to an oracle.*
- *If* $k = 4$*:* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{>0} \subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{q/k>}$ *and* $\mathsf{ZPP}^{\mathsf{NP}[1]}_{1/k} \not\subseteq \mathsf{ZPP}^{\mathsf{NP}\|[q]}_{>q/k}$ *relative to an oracle.*

Acknowledgments

I thank anonymous referees for their comments. This work was supported by NSF grants CCF-1657377 and CCF-1942742. An extended abstract of this paper was published as [Wat19].

References

[Bei91] Richard Beigel. Bounded queries to SAT and the boolean hierarchy. *Theoretical Computer Science*, 84(2):199–223, 1991. doi:10.1016/0304-3975(91)90160-4.

[CC06] Jin-Yi Cai and Venkatesan Chakaravarthy. On zero error algorithms having oracle access to one query. *Journal of Combinatorial Optimization*, 11(2):189–202, 2006. doi:10.1007/s10878-006-7130-0.

[CP08] Richard Chang and Suresh Purini. Amplifying $\mathrm{ZPP}^{\mathrm{SAT}[1]}$ and the two queries problem. In *Proceedings of the 23rd Conference on Computational Complexity (CCC)*, pages 41–52. IEEE, 2008. doi:10.1109/CCC.2008.32.

[GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Computational Complexity*, 27(2):245–304, 2018. doi:10.1007/s00037-018-0166-6.

[Roh95] Pankaj Rohatgi. Saving queries with randomness. *Journal of Computer and System Sciences*, 50(3):476–492, 1995. doi:10.1006/jcss.1995.1038.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985. doi:10.1137/0214060.

[Tri10] Rahul Tripathi. The 1-versus-2 queries problem revisited. *Theory of Computing Systems*, 46(2):193–221, 2010. doi:10.1007/s00224-008-9126-x.

[Ver95] Nikolai Vereshchagin. Lower bounds for perceptrons solving some separation problems and oracle separation of AM from PP. In *Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems (ISTCS)*, pages 46–51. IEEE, 1995. doi:10.1109/ISTCS.1995.377047.

[Ver99] Nikolai Vereshchagin. Relativizability in complexity theory. In *Provability, Complexity, Grammars*, volume 192 of *AMS Translations, Series 2*, pages 87–172. American Mathematical Society, 1999.

[Wat19] Thomas Watson. Amplification with one NP oracle query. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP), Track A*, pages 96:1–96:13. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.ICALP.2019.96.

[Wat20] Thomas Watson. A $\mathrm{ZPP}^{\mathrm{NP}[1]}$ lifting theorem. *ACM Transactions on Computation Theory*, 12(4):27:1–27:20, 2020. doi:10.1145/3428673.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il