Electronic Colloquium on Computational Complexity, Report No. 95 (2020)

On Basing Auxiliary-Input Cryptography on NP-hardness via Nonadaptive Black-Box Reductions

Mikito Nanashima*
Tokyo Institute of Technology

June 24, 2020

Abstract

A black-box (BB) reduction is a central proof technique in theoretical computer science. However, the limitations on BB reductions have been revealed for several decades, and the series of previous work gives strong evidence that we should avoid a nonadaptive BB reduction to base cryptography on NP-hardness (e.g., Akavia et al., 2006). Then should we also give up such a familiar proof technique even for an intermediate step towards cryptography?

In this paper, we continue to explore the capability of nonadaptive BB reductions and extend our knowledge on such a central technique out of the current (worst-to-average) framework. In particular, we investigate the attempt to base weaker cryptographic notions allowed to take auxiliary-input via nonadaptive BB reductions. As a result, we prove the following theorems:

- if we base an auxiliary-input pseudorandom generator (AIPRG) on NP-hardness via a nonadaptive BB reduction, then the polynomial hierarchy collapses;
- if we base an auxiliary-input one-way function (AIOWF) or auxiliary-input hitting set generator (AIHSG) on NP-hardness via a nonadaptive BB reduction, then an (i.o.-)one-way function also exists based on NP-hardness (via an adaptive BB reduction).

The first result gives new evidence that nonadaptive BB reductions are insufficient to base AIPRG. The second result also yields a weaker but still surprising consequence of nonadaptive BB reductions, that is, a one-way function based on NP-hardness. In fact, the second result is interpreted as the following two opposite ways. Pessimistically, it shows that basing AIOWF or AIHSG via nonadaptive BB reductions is harder than constructing a one-way function based on NP-hardness, which can be regarded as a negative result. Note that AIHSG is a weak primitive implied even by the hardness of learning; thus, this pessimistic view gives conceptually stronger limitations than the currently known limitations on nonadaptive BB reductions. Optimistically, our result gives a new hope: a breakthrough construction of auxiliary-input primitives might also be useful to construct standard cryptographic primitives. This optimistic view enhances the significance of further investigation on constructing auxiliary-input or other intermediate cryptographic primitives instead of standard cryptographic primitives.

*nanashima.m.aa@is.c.titech.ac.jp

ISSN 1433-8092

1 Introduction

How can we translate computational hardness into useful hardness in cryptography? This question is a central issue in theoretical computer science. Specifically, one of the most significant and long-standing challenges is constructing fundamental cryptographic primitives such as a one-way function based on NP-hardness. At present, several breakthroughs seem to be required for this challenge, as surveyed by Impagliazzo (1995).

A central ingredient for resolving the above challenge is a *reduction*; in other words, the way to translate recognizing a language into breaking a cryptographic primitive. A reduction is a powerful proof technique even if it is restricted to a quite simple form, and in fact, a nonadaptive black-box (BB) reduction has played a crucial role to show many brilliant results in theoretical computer science. Therefore, it is a natural attempt to apply such a familiar proof technique even for constructing secure cryptographic primitives.

However, Bogdanov and Trevisan (2006a) gave strong evidence that such a simple reduction is insufficient for cryptography based on NP-hardness. In general, breaking cryptographic primitives is formulated as an NP problem on an efficiently samplable distribution that is fixed in advance. They showed that there is no nonadaptive BB reduction from an NP-hard problem to such a distributional NP problem unless the polynomial hierarchy collapses. Therefore, as a corollary, their work excluded the attempt to apply nonadaptive BB reductions for basing cryptography under the reasonable assumption on the polynomial hierarchy. Moreover, subsequent work gave stronger consequences in more specific cases of basing several cryptographic primitives (Akavia et al., 2006; Gutfreund and Vadhan, 2008; Applebaum et al., 2008; Haitner et al., 2010; Bogdanov and Lee, 2013; Bogdanov and Brzuska, 2015; Liu and Vaikuntanathan, 2016; Hirahara and Watanabe, 2020).

Then should we also give up all nonadaptive BB strategies even for an intermediate step towards cryptography? This question essentially motivates our work. In this spirit, we focus on the capability of nonadaptive BB reduction for basing a weaker cryptographic notion, that is, an *auxiliary-input* cryptographic primitive, introduced first by Ostrovsky and Wigderson (1993). Informally speaking, an auxiliary-input cryptographic primitive is defined as a family of primitives indexed by the auxiliary-input and has a relaxed security requirement: at least one primitive in the family is required to be secure depending on each adversary (instead of one fixed primitive secure against all adversaries). In other words, adversaries must break all primitives in the worst-case sense on auxiliary-input, and the task is not directly formulated as a distributional NP-problem because the distribution is not uniquely determined beforehand due to the auxiliary-input. Thus the previous work on distributional NP problem cannot be directly applied to auxiliary-input cryptography.

Now let us mention the current status of nonadaptive BB reductions to auxiliary-input cryptography. Applebaum et al. (2008) observed that we cannot apply a nonadaptive fixed-auxiliary-input BB reduction, which is a restricted nonadaptive BB reduction given access to only one auxiliary-input, unless the polynomial hierarchy collapses. However, the restricted access to auxiliary-input seems to be too strict and it implicitly yields a reduction from an NP-hard language to some fixed cryptographic primitive (depending on the instance). In fact, the above result was shown in almost the same way to the previous result by Akavia et al. (2006) for standard cryptographic primitives. The same work and later Xiao (2009b) observed that generalizing their result to nonadaptive BB reductions seems hard by giving the explicit technical issue. To the best of our knowledge, we have no negative result on general nonadaptive BB reductions to base auxiliary-input cryptography on NP-hardness at present.

The recent progress on the minimum circuit size problem revealed that an auxiliary-input one-way function indeed implies a hard-on-average distributional NP problem (Allender and Hirahara, 2019; Hirahara, 2018). However, such an implication uses several non-black-box and adaptive

techniques (e.g., Håstad et al., 1999; Hirahara, 2018). Thus, the property on nonadaptive and black-box is lost in translating reductions for the auxiliary-input primitive into reductions for the distributional NP problem.

In this paper, based on the above status, we continue to investigate the capability of (general) nonadaptive BB reductions on basing auxiliary-input cryptographic primitives. The importance of our work is to extend our current knowledge on such a central proof technique out of the previous worst-to-average framework by Bogdanov and Trevisan (2006a) and to identify the inherent difficulty on constructing cryptographic primitives on NP-hardness more finely.

1.1 Our Contribution

Our main contribution is to give new knowledge about nonadaptive BB reductions from an NP-hard problem to an auxiliary-input cryptographic primitive. In particular, we handle the auxiliary-input analogs of the following three fundamental primitives: a one-way function, a pseudorandom generator, and a hitting set generator. A formal definition of each primitive will be given in Section 2 with the formal description of the main theorem. First, we informally present our main theorem.

Theorem (informal)**.** *If there is a nonadaptive BB reduction from an* NP*-hard language L to breaking an auxiliary-input cryptographic primitive P, then according to the type of P we have that:*

- *if P is an auxiliary-input pseudorandom generator, then the polynomial hierarchy collapses;*
- *if P is an auxiliary-input one-way function or an auxiliary-input hitting set generator, then there is also an adaptive reduction from L to inverting some (i.o.-)one-way function.*

The first result shows reasonable evidence that auxiliary-input pseudorandom generators (AIPRG) cannot be based on NP-hardness via nonadaptive BB reductions as standard cryptography. The second result shows that a nonadaptive BB reduction for basing the other auxiliary-input primitives yields another strong consequence: an "infinitely often" analog of one-way function based on NP-hardness. Remark that an auxiliary-input hitting set generator (AIHSG) is much weaker primitive than standard cryptographic primitives: for example, the existence is even weaker than the hardness of PAC learning (Nanashima, 2020). What is surprising is that even a nonadaptive BB reduction to such a weak primitive yields a solution close to the long-standing challenge.

The second result is not sufficient to exclude nonadaptive BB reductions on basing auxiliary-input primitives, and it has two opposite interpretations. However, let us stress that both interpretations are quite nontrivial and yield new knowledge about nonadaptive BB reductions. One interpretation is a pessimistic (or realistic) one. As mentioned in the introduction, no one has not come up with the construction of a one-way function based on NP-hardness for several decades despite its importance. Thus, this result is still strong evidence of difficulty finding such a simple reduction. The other interpretation is an optimistic one as a new approach to constructing a one-way function. We will further discuss this optimistic perspective and its novelty in Section 3.

A reader who is familiar with cryptography may wonder why the consequences are different between an auxiliary-input one-way function (AIOWF) and AIPRG. In fact, AIPRG is constructed from any AIOWF by applying the known BB construction of a pseudorandom generator from a one-way function. However, if such construction requires an adaptive security proof, then the property on nonadaptive is lost in translating reductions for AIOWF into reductions for AIPRG via the adaptive security reduction. To the best of our knowledge, all currently known constructions of pseudorandom generators (e.g., Håstad et al., 1999; Holenstein, 2006; Haitner et al., 2013) use adaptive techniques in the security proof; for example, construction of false entropy generators and the uniform hardcore lemma (Holenstein, 2005). This technical issue prevents us from applying the

first result for AIPRG to AIOWF. For a similar reason, our second result on AIOWF is incomparable with the previous work on basing hardness of learning by Applebaum et al. (2008)[1].

2 Formal Descriptions

Now we present formal descriptions of auxiliary-input primitives and our results. Let us introduce a few notations. For any $n \in \mathbb{N}$, let U_n be a random variable selected according to a uniform distribution over $\{0,1\}^n$. For any function $f : \mathcal{X} \to \mathcal{Y}$ and subsets $X \subseteq \mathcal{X}$, $Y \subseteq \mathcal{Y}$, let $f(X) = \{f(x) : x \in X\}$ and $f^{-1}(Y) = \{x \in X : f(x) \in Y\}$. For a language L, let (L, U) be a distributional problem of recognizing $L(x)$ on an instance x selected uniformly at random (for the detail, see Section 5.2). An auxiliary-input cryptographic primitive is defined as an auxiliary-input function with some additional security conditions.

Definition 1 (Auxiliary-input function)**.** *A (polynomial-time computable) auxiliary-input function is a family $f = \{f_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{\ell(|z|)}\}_{z\in\{0,1\}^*}$, where $n(|z|)$ and $\ell(|z|)$ are polynomially-related[2] to $|z|$, which satisfies that there exists a polynomial-time evaluation algorithm F such that for any $z \in \{0,1\}^*$ and $x \in \{0,1\}^{n(|z|)}$, $F(z,x)$ outputs $f_z(x)$.*

In this paper, we use the term "an auxiliary-input function (AIF)" to refer to polynomial-time computable one as in the above definition unless otherwise stated. For simplicity, we assume that $n(\cdot)$ and $\ell(\cdot)$ are increasing functions. Note that the length of auxiliary-input is possibly longer than the length of input and output, that is, $|z| > n(|z|)$ and $|z| > \ell(|z|)$. We may write $n(|z|)$ (resp. $\ell(|z|)$) as n (resp. ℓ) when the dependence of $|z|$ is obvious.

2.1 Auxiliary-Input Pseudorandom Generator

A pseudorandom generator is a primitive stretching a short random seed to a long binary string random-looking from all efficiently computable adversaries. The auxiliary-input analog is formally defined as follows:

Definition 2 (Auxiliary-input pseudorandom generator)**.** *Let $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. For a function $\gamma : \mathbb{N} \to (0,1)$, we say that a randomized algorithm A γ-distinguishes G if for any auxiliary-input $z \in \{0,1\}^*$,*

$$\left| \Pr_{A,U_n}[A(z, G_z(U_n)) = 1] - \Pr_{A,U_{\ell(n)}}[A(z, U_{\ell(n)}) = 1] \right| \geq \gamma(n).$$

We say that G is an auxiliary-input pseudorandom generator (AIPRG) if $\ell(n) > n$ and for any polynomial p, there exists no polynomial-time randomized algorithm $(1/p)$-distinguishing G.

A BB reduction for AIPRG is defined as follows. It is easily checked that the following BB reduction from a language L to distinguishing an AIF G shows that G is an AIPRG if $L \notin \mathsf{BPP}$.

Definition 3 (Black-box reduction to distinguishing AIF)**.** *Let L be a language and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function with $\ell(n) > n$. We say that there exists a black-box (BB) reduction from L to distinguishing G if for any polynomial p, there exists a*

[1] Although the hardness of learning is conceptually weaker than AIOWF, their work used the property of black-box in the formulation of a reduction to learning and indeed yielded a reduction to inverting AIOWF in the end.

[2] In the case of $n(|z|)$, it means that there exist $c, c' \in \mathbb{N}$ such that $|z| \leq c \cdot n(|z|)^c$ and $n(|z|) \leq c' \cdot |z|^{c'}$.

randomized polynomial-time oracle machine $R^?$ such that for any oracle $\mathcal{O}$ that $(1/p)$-distinguishes G and $x \in \{0,1\}^*$, R satisfies that

$$\Pr_R[R^{\mathcal{O}}(x) = L(x)] \geq 2/3.$$

Moreover, we say that there exists a nonadaptive BB reduction from L to distinguishing G if all R make its queries independently of any answer by oracle for previous queries.

Now we present the first main result on AIPRG.

Theorem 1. *For any auxiliary-input function $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ with $\ell(n) > n$, there exists no nonadaptive BB reduction from an NP-hard language L to distinguishing G unless the polynomial hierarchy collapses.*

2.2 Auxiliary-Input One-Way Function

A one-way function is a function which evaluating is easy, but inverting is hard, and it is a fundamental primitive in the sense that most cryptographic tools do not exist without a one-way function (Impagliazzo and Luby, 1989; Rompel, 1990). The formal definition is the following:

Definition 4 (One-way function)**.** *Let s, ℓ be polynomials. We say that a family of function $f = \{f_n\}_{n\in\mathbb{N}}$ where $f_n : \{0,1\}^{s(n)} \to \{0,1\}^{\ell(n)}$ is an (i.o.-)one-way function (OWF) if f is polynomial-time computable, and there exists a polynomial p such that for any polynomial-time randomized algorithm A, there exists infinitely many $n \in \mathbb{N}$ such that*

$$\Pr_{A,U_{s(n)}}[A(1^n, f_n(U_{s(n)})) \notin f_n^{-1}(f_n(U_{s(n)}))] \geq 1/p(n).$$

For simplicity, we may omit to write the input 1^n to A.

The auxiliary-input analog of OWF is the following, which is first introduced by Ostrovsky and Wigderson (1993).

Definition 5 (Auxiliary-input one-way function)**.** *Let $f = \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ be an auxiliary-input function and $\gamma : \mathbb{N} \to (0,1)$ be a function. We say that a randomized algorithm A γ-inverts f if for any $z \in \{0,1\}^*$,*

$$\Pr_{A,U_n}[A(z, f_z(U_n)) \in f_z^{-1}(f_z(U_n))] \geq \gamma(n).$$

We say that f is an auxiliary-input one-way function (AIOWF) if there exists a polynomial p such that no polynomial-time randomized algorithm $(1 - 1/p)$-inverts f.

In fact, the existence of AIOWF and AIPRG is equivalent (Håstad et al., 1999). However, we cannot directly apply Theorem 1 to AIOWF due to the adaptive security reduction, as we mentioned in Section 1.1.

A BB reduction for AIOWF is defined as follows. It is easily checked that for any polynomial p, the following BB reduction from a language L to $(1 - 1/p)$-inverting an AIF f shows that f is an AIOWF if $L \notin \mathsf{BPP}$.

Definition 6 (Black-box reduction to inverting AIF)**.** *Let L be a language, p be a polynomial, and $f := \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. We say that a randomized*

polynomial-time oracle machine $R^?$ is a black-box (BB) reduction from L to $(1-1/p)$-inverting f if for any oracle $\mathcal{O}$ that $(1-1/p)$-inverts f and $x \in \{0,1\}^*$, R satisfies that

$$\Pr_R[R^{\mathcal{O}}(x) = L(x)] \geq 2/3.$$

Moreover, we say that R is nonadaptive if all R's queries are made independently of any answer by oracle for previous queries.

Now we present the second main result on AIOWF.

Theorem 2. *For any auxiliary-input function $f = \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ and polynomial p, if there exists a nonadaptive BB reduction from an NP-hard language L to $(1-1/p)$-inverting f, then $\mathsf{NP} \nsubseteq \mathsf{BPP}$ also implies that a one-way function exists (via an adaptive BB reduction).*

2.3 Auxiliary-Input Hitting Set Generator

A hitting set generator is a weak variant of a pseudorandom generator, introduced in the context of derandomization by Andreev et al. (1998). For the original purpose, we consider (possibly) exponential-time computable generators. In this paper, however, we focus on polynomial-time computable generators as in cryptography. Now we define the auxiliary-input analog.

Definition 7 (Auxiliary-input hitting set generator)**.** *Let $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. For a function $\gamma : \mathbb{N} \to (0,1)$, we say that a randomized adversary A γ-avoids G if for any (public) auxiliary-input $z \in \{0,1\}^*$ and (private) input $x \in \{0,1\}^{n(|z|)}$,*

$$\Pr_A[A(z, G_z(x)) = 0] \geq 2/3 \quad \textit{and} \quad \Pr_{y\sim\{0,1\}^{\ell(n(|z|))}}\left[\Pr_A[A(z,y)=1] \geq 2/3\right] \geq \min(\gamma(n), \tau_z),$$

where τ_z be a trivial limitation[3] defined as $\tau_z = 1 - \frac{|G_z(\{0,1\}^n)|}{2^{\ell(n)}}$.

We say that G is a γ-secure auxiliary-input hitting set generator (AIHSG) if $\ell(n) > n$ and there exists no polynomial-time randomized algorithm $(1-\gamma)$-avoiding G.

Although it is easily checked that AIPRG is also AIHSG (with any security $\gamma(n) = 1/\mathsf{poly}(n)$), the opposite direction is open at present. In fact, the hardness of learning implies the existence of AIHSG (Nanashima, 2020); on the other hand, we must overcome the barrier by oracle separation to show the existence of AIPRG (equivalently, AIOWF) from the hardness of learning (Xiao, 2009a). Thus, AIHSG seems to be a much weaker notion than AIOWF and AIPRG under current knowledge.

A BB reduction for AIHSG is defined as follows. It is easily checked that the following BB reduction from a language L to $(1-\gamma)$-avoiding an AIF G shows that G is a γ-secure AIHSG if $L \notin \mathsf{BPP}$.

Definition 8 (Black-box reduction to avoiding AIF)**.** *Let L be a language, γ be a function, and $G := \{G_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. We say that a randomized polynomial-time oracle machine $R^?$ is a black-box (BB) reduction from L to $(1-\gamma)$-avoiding G if for any oracle $\mathcal{O}$ that $(1-\gamma)$-avoids G and $x \in \{0,1\}^*$, R satisfies that*

$$\Pr_R[R^{\mathcal{O}}(x) = L(x)] \geq 2/3.$$

Moreover, we say that R is nonadaptive if all R's queries are made independently of any answer by oracle for previous queries.

[3] In this paper, we consider general settings of γ and ℓ. Thus, we adopted the trivial limitation in the definition to avoid arguing about invalid settings where γ-avoiding the generator is impossible by definition.

Now we present the third main result on AIHSG.

Theorem 3. *Let p be a polynomial and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function where $\ell(n) > (1+\epsilon)\cdot n$ for some constant $\epsilon > 0$. If there exists a nonadaptive BB reduction from an* NP*-hard language L to $(1-1/p)$-avoiding G, then* NP $\nsubseteq$ BPP *also implies that a one-way function exists (via an adaptive BB reduction).*

3 Discussion and Future Directions

As mentioned in Section 1.1, Theorems 2 and 3 are also regarded as approaches to construct one-way functions based on NP-hardness. In this section, we discuss the novelty of this optimistic perspective and give future directions, including the investigation of the validity.

Our results are rephrased as follows: Assume that we could connect NP-hardness to some auxiliary-input primitives (i.e., AIOWF or AIHSG) with a novel (nonadaptive BB) reduction, then we can automatically extend the connection to standard cryptographic primitives, that is, OWF. At present, the latter task of removing auxiliary-input from primitives seems to be quite non-trivial. To see this, we give a simple oracle separation between AIOWF and OWF as the fourth result.

Theorem 4. *There exists an oracle $\mathcal{O}$ such that relative to $\mathcal{O}$ an auxiliary-input one-way function exists, but a one-way function does not exist.*

Thus, we cannot expect any relativized technique to remove auxiliary-input from cryptographic primitives. Additionally, there are several barriers by other oracle separations at the intermediate levels to base OWF on NP-hardness (e.g., Xiao, 2009a; Impagliazzo, 2011). Although such barriers on relativization are common throughout theoretical computer science (e.g., the P vs. NP problem by Baker et al., 1975), there are only a few success stories of overcoming such barriers at present. Unfortunately, Theorems 2 and 3 do not give any solution to break these barriers, and a new non-relativized technique is still required. Specifically, if a nonadaptive BB reduction to AIOWF or AIHSG is also relativized[4], then our proof also yields relativized reductions that contradict Theorem 4 or the oracle separation by Impagliazzo (2011).

However, our result gives one hope. Although there seems to be several barriers towards cryptography based on NP-hardness, the essential barrier we must overcome might be few. Theorems 2 to 4 certainly show that if we could find a non-relativized breakthrough at an intermediate level toward cryptography (that is, auxiliary-input primitives), then it will be lifted and break the other barriers at the higher level. From this perspective, we conjecture that the hardness of basing OWF might heavily rely on a much smaller part at an intermediate level. This conjecture seems to be somewhat controversial but enhances the significance of further investigation on basing auxiliary-input or other intermediate cryptographic primitives instead of standard ones.

The above discussion leads to the following two possible directions. The first direction is to find other scenarios where a breakthrough at an intermediate level also brings benefits at the higher level. This direction might reduce constructing standard cryptographic primitives to the task at the low level and give new insights into complexity-based cryptography. The second direction is to refute such an attempt on intermediate primitives with convincing evidence if it gives the wrong direction. Particularly, in our case, there is a possibility that nonadaptive BB reductions to base AIOWF and AIHSG indeed yield the collapse of the polynomial-hierarchy as in the case of AIPRG.

For the second direction, we list two concrete ways: (1) finding a new construction of AIPRG from AIOWF with nonadaptive security proof; (2) generalizing the previous results for OWF (Akavia

[4]Note that oracle separations do not necessarily rule out BB reductions form particular languages, not as fully BB reductions defined by Reingold et al. (2004).

et al., 2006) or HSG (Hirahara and Watanabe, 2020) to each auxiliary-input analog for the stronger consequence. At least the latter approach seems to require some new technique to simulate non-adaptive BB reductions, as observed by Applebaum et al. (2008) and Xiao (2009b).

4 Proof Sketches

In this section, we give proof ideas of Theorems 1 to 4, and each formal proof will be given in Sections 6 to 9, respectively. Note that Theorem 3 heavily relies on Theorem 2, and Theorem 2 heavily relies on Theorem 1. Therefore, although each proof idea may look pretty simple and intuitive, our construction of OWF for Theorem 3 becomes complicated and quite non-trivial as a whole.

4.1 On Basing AIPRG: Proof Idea of Theorem 1

First, we formally introduce a hitting set generator, which takes a crucial role in our proof.

Definition 9 (Hitting set generator)**.** *Let $\gamma(n)$ be a function. A function $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ with $\ell(n) > n$ is a (polynomial-time computable) γ-secure hitting set generator (HSG) if G is polynomially computable and there is no polynomial-time randomized adversary A γ-avoiding G, that is, satisfying that for any sufficiently large $n \in \mathbb{N}$,*

$$\forall x \in \{0,1\}^n \Pr_A[A(G(x)) = 0] \geq 2/3 \quad \textit{and} \quad \Pr_{y \sim \{0,1\}^{\ell(n)}}\left[\Pr_A[A(y) = 1] \geq 2/3\right] \geq \min(\gamma(n), \tau_n),$$

where τ_n be a trivial limitation defined as $\tau_n := 1 - \frac{|G(\{0,1\}^n)|}{2^{\ell(n)}}$.

The essential part of the proof is to give a construction of HSG from AIPRG with a nonadaptive BB security reduction from distinguishing AIPRG to avoiding HSG. Note that such an implication has been already known if we allow the non-black-box technique by Hirahara (2018). For our purpose, however, the conditions on nonadaptive and black-box are crucial, as seen below. Thus we will give a much simpler construction to show the same implication. Although the reader may think that our construction is too fundamental and looks somewhat trivial, to the best of our knowledge, no one has mentioned such a clear relationship between AIPRG and HSG.

To see why the conditions on nonadaptive and black-box are crucial, first assume a nonadaptive BB security reduction from distinguishing AIPRG to avoiding HSG. Avoiding HSG is directly formulated as the following distributional NP problem (with zero-error): for uniformly chosen y, determine whether y is contained in the image of HSG. Therefore, the reduction also yields a nonadaptive BB reduction from distinguishing AIPRG to the distributional NP problem (formally, Lemma 4). Thus, any nonadaptive BB reduction from an NP-hard problem to distinguishing AIPRG indeed yields a nonadaptive BB reduction from the same NP-hard problem to the distributional NP problem. By the previous result by Bogdanov and Trevisan (2006a) (formally, Fact 2), such a reduction implies the collapse of the polynomial-hierarchy.

Our construction of HSG from AIPRG is the following (formally, Lemma 3): just considering the both of auxiliary-input and input to AIPRG as usual input to HSG. More specifically, let $G = \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z \in \{0,1\}^*}$ be an AIPRG. Then the construction of HSG G' is given as $G'(z \circ x) = G_z(x)$. Note that, when $z + n(|z|) < \ell(n(|z|))$ holds, G' does not satisfy the syntax on stretching input. In the formal proof, therefore, we first stretch the output of G by the standard technique in cryptography. We can easily check that the security reduction for this stretching (shown by the famous hybrid argument) is nonadaptive.

Let $\gamma(n)$ be a reciprocal of polynomial. The security reduction from γ-avoiding G' to distinguishing G is also simple: just considering an adversary A for G' as an adversary for G. Obviously, this reduction is nonadaptive. To see the correctness, assume that A γ-avoids G'. For simplicity, we also assume that A is deterministic and $\gamma(n) < \tau_n$. Whenever the input y is pseudorandom string contained in the image of G', $A(y)$ does not output 1. On the other hand, when y is a truly random string, then $A(y)$ outputs 1 with probability at least $\gamma(n)$. Thus, A can distinguish the uniform distribution from all distributions on the image of G' with an advantage at least $\gamma(n)$. For any auxiliary-input z, $G_z(U_{n(|z|)})$ is distributed on the image of G'. Thus, A also γ-distinguishes G.

4.2 On Basing AIOWF: Proof Idea of Theorem 2

To focus on the idea, we omit all arguments about the success probabilities of adversaries in this section. First, let us prepare several reductions. Let $R_{L\to f}$ be the nonadaptive BB reduction from L to inverting f in the assumption. By the construction of PRG from OWF (e.g., Håstad et al., 1999), there exist an auxiliary-input generator G and an adaptive BB reduction $R_{f\to G}$ from inverting f to distinguishing G. By the result in Section 4.1, there exist an NP-language L' and a nonadaptive BB reduction $R_{G\to L'}$ from distinguishing G to a distributional NP problem (L', U) (with zero-error). Since $L' \in \mathsf{NP}$ and L is NP-hard, there exists a Karp reduction $R_{L'\to L}$ from L' to L.

Now we consider the following procedure:

1. select an instance x' of L' at random;
2. translate x' into an instance x of L as $x = R_{L'\to L}(x')$;
3. plug x into $R_{L\to f}$ with a random tape r;

At this stage, $R_{L\to f}$ makes polynomially many queries $(z_1, y_1), \ldots, (z_q, y_q)$.

4. answer the queries by some inverting oracle $\mathcal{O}$;
5. output the same decision $b \in \{0, 1\}$ to $R_{L\to f}$.

Note that if the oracle $\mathcal{O}$ correctly inverts f, then the resulting decision b is $L(x)$ with high probability, and $L(x)$ is equal to $L'(x')$.

The crucial observation is that there is no worst-case sense at all in the above procedure because both x' and r are selected at random. Therefore, all queries at the stage 3 are indeed efficiently samplable, and the inverting oracle no longer needs to invert f for every auxiliary-input at the stage 4. This observation leads to our construction of a standard OWF g.

The function g takes three inputs x', r, and x^f, which intuitively represents a random instance of L', randomness for $R_{L\to f}$, and input for f, respectively. Then $g(x', r, x^f)$ imitates the above procedure as follows: (2') translate x' into an instance x of L as $x = R_{L'\to L}(x')$, (3') plug x into $R_{L\to f}$ with randomness r, then randomly pick one of auxiliary-input z in queries by $R_{L\to f}$ and output $f_z(x^f)$.

We will show that the above g is one-way if $\mathsf{NP} \nsubseteq \mathsf{BPP}$. For contradiction, assume that there exists an adversary A inverting g. Remember that g simulates a distribution on auxiliary-input in the above procedure. Thus, intuitively, we can replace the inverting oracle $\mathcal{O}$ with the adversary A at the stage 4 with high probability in the execution of $R_{L\to f}$. In fact, this is a little technical part, and we will give further detail in Section 7. Then the above procedure no longer needs any oracle and yields a randomized algorithm solving (L', U) on average. By applying reductions $R_{G\to L'}$, $R_{f\to G}$, and $R_{L\to f}$ in this order, this also yields a randomized polynoimal-time algorithm for L. Since L is NP-hard, we conclude that $\mathsf{NP} \subseteq \mathsf{BPP}$.

Remark that $R_{G\to L'}$ is a nonadaptive BB reduction thanks to our simple construction in Section 4.1. Therefore, if we also have a construction of AIPRG G from AIOWF f with a nonadaptive BB reduction from inverting f to distinguishing G, then the above proof leads to a nonadaptive BB reduction from L to (L', U), which implies the collapse of the polynomial hierarchy as in Theorem 1. Thus, finding such a simple construction of AIPRG is one direction for excluding a nonadaptive BB reduction to base AIOWF, as mentioned in Section 3.

4.3 On Basing AIHSG: Proof Idea of Theorem 3

The key idea for the proof is to classify each query generated by the nonadaptive BB reduction (in the theorem) into a "light" query and a "heavy" query. A similar technique was also used in the previous work for HSG by Gutfreund and Vadhan (2008); Hirahara and Watanabe (2020). We first see the previous case of HSG and then explain the difference to our case of AIHSG.

The Case of Hitting Set Generator (Previous work)

Let $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ be a generator with $\ell(n) \geq (1+\Omega(1)) \cdot n$ and $R^?$ be a nonadaptive BB reduction from an NP-language L to avoiding G. W.l.o.g., we can assume that all (marginal) distributions on queries by R are identical regardless of each query position by applying a random permutation on queries before asking them to oracle. Thus, for each input $x \in \{0,1\}^n$, one distribution Q_x on queries is determined. We choose a threshold (roughly) $\tau = 1/\tilde{\Theta}(2^n)$ and define a light (resp. heavy) query $y \in \{0,1\}^{\ell(n)}$ as a query generated according to Q_x with probability less (resp. greater) than the threshold τ.

The essential part of the proof is to simulate the avoiding oracle for G by using the classification of queries. First, assume that we could (somehow) distinguish the heavy case and the light case for a given query. Then we can also simulate one of avoiding oracles simply as follows: for each query y generated by $R(x)$, (1) determine whether y is heavy or light; (2) answer 0 (resp. 1) if y is heavy (resp. light) query. Let $\mathcal{O}'$ be the induced oracle by the above simulation. Note that the probability that $\mathcal{O}'(y)$ outputs 0 is exponentially small because the fraction of light query is $\tilde{\Theta}(2^n)/2^{\ell(n)} \leq 2^{-\Omega(1)n}$ for the length $\ell(n)$ of query. Thus, $\mathcal{O}'$ satisfies the condition on the probability of outputting 1. However, $\mathcal{O}'$ is not avoiding oracle for G, because possibly there is a query y such that y is heavy but contained in $\mathrm{Im}G$. In this case, $\mathcal{O}'(y)$ outputs 1 for $y \in \mathrm{Im}G$.

The key observation to overcome this issue is the following:

> ($\star$) For each length $\ell(n)$ of query (where the input size is n), the size of $\mathrm{Im}G$ is at most 2^n; thus the probability that R asks some light query contained in $\mathrm{Im}G$ (that is, "bad" query) is bounded above by $2^n/\tilde{\Theta}(2^n) \leq 1/\mathsf{poly}(n)$.

Therefore, $\mathcal{O}'$ is consistent with some avoiding oracle, and $R^{\mathcal{O}'}(x)$ correctly recognizes x with high probability over the execution of R.

By the above argument, we can reduce avoiding a generator to distinguishing heavy and light queries. For the latter task, Gutfreund and Vadhan (2008) gave a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm by approximation of counting by Jerrum et al. (1986), and Hirahara and Watanabe (2020) gave an $\mathsf{AM}\cap\mathsf{coAM}$ algorithm by the generalized version of the protocol by Feigenbaum and Fortnow (1991).

The Case of Auxiliary-input Hitting Set Generator (Our work)

Now we move on to our case of AIHSG. Let $G = \{G_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{\ell(n(|z|))}\}_{z\in\{0,1\}^*}$ be an auxiliary-input generator with $\ell(n) \geq (1+\Omega(1)) \cdot n$ and $R^?$ be a nonadaptive BB reduction from an

NP-language L to avoiding G. We can also assume that all query distributions of $R^?(x)$ are identical to Q_x regardless of query position.

On applying the above argument to our case of AIHSG, the problematic part is the key observation ($\star$). Remember that an adversary for AIHSG must avoid G_z for all $z \in \{0,1\}^*$, and auxiliary-input is possibly longer than output. Therefore, we cannot bound the size of the image of the generator in general because the image may span the whole range (for example, consider the following generator $G_z(x) = z \oplus (x \circ 0^{|z|-|x|})$ for $|z| > n(|z|)$).

To resolve this, we need to consider each case of auxiliary-input z separately. Therefore, we change the definitions of "light" and "heavy" and let them adapt to auxiliary-input. Let $p_x(z)$ be a probability that Q_x generates a query of auxiliary-input z. If we can bound the probability that R makes light query (z, y) with $y \in \mathrm{Im}G_z$ by $1/(\mathsf{poly}(n) \cdot p_x(z))$ for any z, then R makes such a "bad" query (z, y) with probability at most $\sum_z p_x(z) \cdot 1/(\mathsf{poly}(n) \cdot p_x(z)) = 1/\mathsf{poly}(n)$. Then we can use the same argument in the case of HSG and reduce avoiding G to distinguishing heavy and light cases. This idea naturally leads to the following new definition of "light" and "heavy": separating each query (z, y) by the conditional probability $p_x(y|z)$ that y is asked conditioned on the event that z is asked. In fact, as shown in Section 8, this modification will work well even for AIHSG.

However, one issue remains: how can we distinguish heavy and light queries? To this end, we must verify the largeness of the conditional probability of the given query. This part essentially prevents us from applying the previous results. Since we consider a polynomial-time computable generator, the simulation with NP oracle does not give any nontrivial result, not as the work by Gutfreund and Vadhan (2008)[5]. Even for the simulation in $\mathsf{AM} \cap \mathsf{coAM}$ by Hirahara and Watanabe (2020), there seem to be several technical issues. We cannot trivially verify the size of conditional probability by such protocols due to the restricted use of the upper bound protocol (Aiello and Håstad, 1987). Moreover, we cannot possibly even sample the conditional distribution efficiently for fixed auxiliary-input (for example, consider the query distribution on $((z, vk), y)$ where y is a secure signature to z verified with a public-key vk).

Our idea is to adopt universal extrapolation by Impagliazzo and Levin (1990). Intuitively speaking, the universal extrapolation is a tool to reduce approximating the probability $p_y = \Pr_{U_n}[y = f(U_n)]$ to inverting f for any polynomial-time computable f under the situation where $y = f(x)$ and $x \in \{0,1\}^n$ is selected at random. In fact, the universal extrapolation holds even for auxiliary-input function, and a similar technique was also used by Ostrovsky and Wigderson (1993). By using the universal extrapolation for each circuit sampling query and auxiliary-input, we have a good approximation of $p_x(y|z)$ for query (z, y) generated by $R^?(x)$. Thus, the universal extrapolation enables us to classify the given (z, y) correctly. Note that the auxiliary-input in the universal extrapolation essentially corresponds to the input x for each circuit sampling query and auxiliary-input.

To show Theorem 3, we need further observations. Since R makes its queries nonadaptively, we can also invoke the universal extrapolation nonadaptively. Moreover, the universal extrapolation algorithm indeed uses an inverting adversary for a certain AIOWF as black-box and nonadaptively (we see this formally in Appendix A). As a result, a nonadaptive BB reduction from an NP-hard language L to avoiding AIHSG yields a nonadaptive BB reduction from L to inverting AIOWF. Thus, by Theorem 2, R also yields a one-way function under the assumption that $\mathsf{NP} \nsubseteq \mathsf{BPP}$.

[5]Their work concerned the original aim of HSG, that is, derandomization (e.g., Impagliazzo and Wigderson, 2001). For this purpose, we consider (possibly) exponential-time computable HSG G, and avoiding G in $\mathsf{BPP}^{\mathsf{NP}}$ is quite nontrivial. However, in our case where G is polynomial-time computable, avoiding G is in NP trivially.

4.4 Oracle Separation between OWF and AIOWF: Proof Idea of Theorem 4

To show Theorem 4, we use a random function $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \to \{0,1\}^n\}_{n\in\mathbb{N}}$, where each $\mathcal{F}_n$ is selected uniformly from length-preserving functions of input size n. As shown by Impagliazzo and Rudich (1989), any polynomial-time oracle machine cannot invert $\mathcal{F}$ with non-negligible probability (with probability 1 over the choice of $\mathcal{F}$). In other words, if a primitive given access to $\mathcal{F}$ directly outputs the value of $\mathcal{F}$, such a primitive must be one-way. Therefore, all we have to do is to let a random function $\mathcal{F}$ available for auxiliary-input primitives but unavailable for standard primitives.

To this end, we simply add n-bit auxiliary-input to a random function of the input size n. Then we choose one auxiliary-input z_n from 2^n possibilities of $\{0,1\}^n$ as a target auxiliary-input and embed the random function to the position indexed by z_n. Let $\mathcal{F} = \{F_z : \{0,1\}^{|z|} \to \{0,1\}^{|z|}\}_{z\in\{0,1\}^*}$ be such an embedded random function. Note that the similar random embedding technique was also used in the previous work for other oracle separations (e.g., Xiao, 2009a). If an auxiliary-input primitive f given access to $\mathcal{F}$ identifies the auxiliary-input of F with own auxiliary-input, then f must be AIOWF because an adversary for f must invert f_z for any auxiliary-input z, including the random function. On the other hand, any polynomial-time computable primitive (without auxiliary-input) cannot find the target auxiliary-input of $\mathcal{F}$ with non-negligible probability because they were selected at random. Thus, any (usual) primitive does not take nontrivial advantage of $\mathcal{F}$.

For the oracle separation, we combine the above embedded random function $\mathcal{F}$ with the PSPACE oracle (w.l.o.g., the oracle TQBF determining satisfiability of quantified Boolean formulae). Let $\mathcal{O}_{\mathcal{F}}$ be this new oracle. Since the random function in $\mathcal{F}$ is selected independently of TQBF, the additional access to TQBF does not help to invert the random function at all (formally, Lemma 7). Thus, AIOWF still exists relative to $\mathcal{O}_{\mathcal{F}}$.

On the other hand, consider a function f which is polynomial-time computable with access to $\mathcal{O}_{\mathcal{F}}$ arbitrarily. Since the target auxiliary-input is selected independently of TQBF, the additional access to TQBF does not help to find the target auxiliary-input at all. Thus, f cannot still take nontrivial advantage of $\mathcal{F}$ and is regarded as a function given only access to TQBF. We can easily check that any polynomial-time computable function with access to TQBF is efficiently invertible by TQBF. Since the above argument holds for any f, OWF does not exist relative to $\mathcal{O}_{\mathcal{F}}$ (formally, Lemma 8). Thus, we have the oracle separation between AIOWF and OWF.

In the subsequent sections, we will give full arguments based on the above sketches.

5 Preliminaries

For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. For two strings $x, y \in \{0,1\}^*$, let $\langle x, y\rangle \in \{0,1\}^*$ be a (proper) binary encoding of the tuple (x, y). For $n, k \in \mathbb{N}$, $x \in \{0,1\}^n$, and $n_1, \ldots, n_k \in [n]$ with $\sum_i n_i = n$, we use the notation $x \to_{n_1,\ldots,n_k} (x^{(1)}, \ldots, x^{(k)})$ to refer to the separation of x into k substrings satisfying that $x = x^{(1)} \circ \cdots \circ x^{(k)}$ and $|x^{(i)}| = n_i$ for each $i \in [k]$. For any $x \in \{0,1\}^n$ and $k \in [n]$, let $x_{[k]} = x_1 \circ \cdots \circ x_k$. For $x \in \{0,1\}^n$, we use $x_{\mathbb{N}}$ to refer to the integer given by regarding x as its binary representation (that is, $0 \le x_{\mathbb{N}} \le 2^n - 1$).

We fix a proper encoding for Boolean circuits. For any circuit C, we use $\langle C\rangle$ to explicitly denote the binary encoding of C. Otherwise, we may abuse the same notation C for the encoding. For convenience, we assume the followings: (1) the output length of S-size circuit is at most S; (2) every $u \in \{0,1\}^*$ represents some circuit (by assigning invalid encodings to the trivial circuit $C(x) \equiv 0$); (3) zero-padding is available. These assumptions allow us to assure that there exists a function $e(\cdot)$ such that any n-input circuit of size $S(n)$ has a binary representation of the length $e(S(n))(= \tilde{O}(S(n)))$.

For a randomized algorithm A using $r(n)$ random bits on n-bit input, we use $A(x;s)$ to refer to the execution of $A(x)$ with random tape s for $x \in \{0,1\}^n$ and $s \in \{0,1\}^{r(n)}$.

For a set S, we write $x \leftarrow_u S$ for a random sampling of x according to the uniform distribution over S. We assume the basic facts about probability theory, including the union bound, Markov's inequality, and the Borel-Cantelli lemma. We will make extensive use of the following tail bound by Hoeffding (1963).

Fact 1 (Hoeffding inequality)**.** *For real values $a, b \in \mathbb{R}$, let $X_1, \ldots, X_m$ be independent and identically distributed random variables with $X_i \in [a,b]$ and $\mathrm{E}[X_i] = \mu$ for each $i \in [m]$. Then for any $\epsilon > 0$, the following inequalities hold:*

$$\Pr_{X_1,\ldots,X_m}\left[\frac{1}{m}\sum_{i=1}^{m} X_i - \mu \geq \epsilon\right] \leq e^{-\frac{2m\epsilon^2}{(b-a)^2}} \quad \text{and} \quad \Pr_{X_1,\ldots,X_m}\left[\frac{1}{m}\sum_{i=1}^{m} X_i - \mu \leq -\epsilon\right] \leq e^{-\frac{2m\epsilon^2}{(b-a)^2}}.$$

We introduce the following useful lemma, given as a corollary of Markov's inequality.

Lemma 1. *Let X and Y be (possibly correlated) random variables on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let E be a (bad) event determined only by X and Y. For any $p \in (0,1]$, we define a bad set $B_p^{\mathcal{X}} \subseteq \mathcal{X}$ by*

$$B_p^{\mathcal{X}} = \left\{x \in \mathcal{X} : \Pr_Y[E|X=x] \geq p\right\}.$$

For any $\epsilon \in [0,1]$, if $\Pr_{X,Y}[E] \leq \epsilon p$, then $\Pr_{x\sim X}[x \in B_p^{\mathcal{X}}] \leq \epsilon$.

Proof. By applying Markov's inequality for the nonnegative random variable $P_x = \Pr_Y[E|X=x]$ over the choice of x according to X, we have the lemma as follows:

$$\Pr_{x\sim X}[x \in B_p^{\mathcal{X}}] = \Pr_{x\sim X}[P_x \geq p] \leq \frac{\mathrm{E}_X[P_x]}{p} = \frac{\Pr_{X,Y}[E]}{p} \leq \frac{\epsilon p}{p} = \epsilon.$$

□

5.1 Universal Extrapolation

We formally introduce the key ingredient for our proof, that is, the universal extrapolation.

Lemma 2 (Impagliazzo and Levin 1990: universal extrapolation)**.** *Let $\epsilon \in (0,1]$ and $\delta : \mathbb{N} \to (0,1]$ be a reciprocal of polynomial. If there exists no auxiliary-input one-way function, then for any polynomial $s(n)$, there exists a polynomial-time randomized algorithm $Ext_{s(n)}$ such that for any n-input circuit C of size $s(n)$,*

$$\Pr_{Ext_{s(n)},x\sim\{0,1\}^n}\left[Ext_{s(n)}(C, C(x)) \in [1/2 \cdot p_C(x), 2^{(1+\epsilon)}p_C(x)]\right] \geq 1 - \delta(n),$$

where $p_C(x) := \Pr_{x'\sim\{0,1\}^n}[C(x') = C(x)]$.

Moreover, there exists an auxiliary-input function $f = \{f_z\}_{z\in\{0,1\}^*}$ such that $Ext_{s(n)}$ accesses an inverting algorithm for f nonadaptively as oracle.

Note that we adopted a slightly modified statement for our purpose. Although the proof sketch for the original statement was given in the original paper, the author could not unfortunately find the full version of the proof. To show the correctness and nonadaptiveness explicitly, we will also give the full proof of Lemma 2 based on the original sketch in Appendix A.

5.2 Average-case Complexity

We introduce the basics of average-case complexity. For further details, see the survey by Bogdanov and Trevisan (2006b).

A distributional problem (L, D) is a pair of a language L and a family of distributions $D = \{D_n\}_{n\in\mathbb{N}}$, where D_n is a polynomial-time samplable distribution on instances of length n. Moreover, if $L \in \mathsf{NP}$, we call (L, D) a distributional NP problem. We use the notation $U = \{U_n\}_{n\in\mathbb{N}}$ to denote the family of uniform distributions. The notion of "average-case tractable" by a deterministic or randomized algorithm is defined as follows:

Definition 10 (Errorless heuristic scheme)**.** *Let (L, D) be a distributional problem and $\delta : \mathbb{N} \to (0, 1]$ be a function. We say that a deterministic algorithm A is an errorless heuristic scheme for (L, D) of failure probability δ if A satisfies that for any $n \in \mathbb{N}$,*

1. *$A(x) \in \{L(x), \bot\}$ for any $x \in \{0, 1\}^n$ in the support of D_n;*
2. *$\Pr_{x\leftarrow D_n}[A(x) = \bot] \le \delta(n)$.*

Definition 11 (Randomized errorless heuristic scheme)**.** *Let (L, D) be a distributional problem and $\delta : \mathbb{N} \to (0, 1]$ be a function. We say that a randomized algorithm A is a randomized errorless heuristic scheme for (L, D) of failure probability δ if A satisfies that for any $n \in \mathbb{N}$,*

1. *$A(x) \in \{0, 1, \bot\}$ and $\Pr_A[A(x) = \neg L(x)] \le 1/4$ for any $x \in \{0, 1\}^n$ in the support of D_n;*
2. *$\Pr_{x\leftarrow D_n}[\Pr[A(x) = \bot] \ge 1/4] \le \delta(n)$.*

Bogdanov and Trevisan (2006a) ruled out nonadaptive BB reductions from an NP-hard problem to a distributional NP problem, usually called worst-case to average-case reduction, unless the polynomial hierarchy collapses at the third level.

Fact 2 (Bogdanov and Trevisan 2006a)**.** *For any polynomial p, a language L, and a distributional NP language (L', D), if there exists a nonadaptive BB reduction from L to an errorless heuristic for (L', D) of failure probability $1/p$, then $L \in \mathsf{coNP/poly}$. Moreover, if L is NP-hard, then $\mathsf{PH} = \Sigma_3^p$.*

6 On Basing Auxiliary-Input Pseudorandom Generator

In this section, we formally rule out nonadaptive BB reductions from an NP-hard problem to distinguishing AIPRG based on Section 4.1. Let us state the main theorem again.

Theorem 1. *For any auxiliary-input function $G = \{G_z : \{0, 1\}^n \to \{0, 1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ with $\ell(n) > n$, there exists no nonadaptive BB reduction from an NP-hard language L to distinguishing G unless the polynomial hierarchy collapses.*

First, we give the nonadaptive BB reduction from distinguishing AIPRG to avoiding HSG.

Lemma 3. *Let G be an auxiliary-input function stretching its input and $m : \mathbb{N} \to \mathbb{N}$ be a polynomial. There exists a polynomial-time computable function $G' : \{0, 1\}^n \to \{0, 1\}^{m(n)}$ and a randomized polynomial-time oracle machine $R^?$ satisfying the following: for any polynomial γ', there exists a polynomial γ such that for any oracle $\mathcal{O}$ which $1/\gamma'$-avoids G', $R^{\mathcal{O}}$ $1/\gamma$-distinguishes G. Moreover, $R^?$ is nonadaptive.*

Proof. W.l.o.g, we can assume that G has the stretch $\ell(n) = n+1$ by discarding the suffix of output. We define the generator $G' : \{0,1\}^{n'} \to \{0,1\}^{m(n')}$ in the lemma by

$$G'(x) = \begin{cases} b_1 \circ \cdots \circ b_{m(n)} & \text{if } (n' =)|x| = a + n(a) \text{ for some } a \in \mathbb{N} \\ 0^{m(n)} & \text{otherwise,} \end{cases}$$

where each $b_i \in \{0,1\}$ is a bit determined by the following procedure: (1) $x \to_{a,n(a)} (z, x^{(0)})$; (2) $G_z(x^{(i-1)}) \to_{n(a),1} (x^{(i)}, b_i)$ for each $i \in [m(n)]$. It is easily checked that G' is polynomial-time computable.

Now we define the nonadaptive reduction $R^?$ in the lemma as Algorithm 1.

Algorithm 1: R (a nonadaptive BB reduction from distinguishing G to avoiding G')

Input : an auxiliary-input $z \in \{0,1\}^a$ and $y \in \{0,1\}^{n(a)+1}$
Oracle : $\mathcal{O}$ ($1/\gamma'$-avoiding G')

let $m := m(a + n(a))$ and select $k \leftarrow_u [m]$;
let $x^{(k)} := y_{[n(a)]}$;
for $i = 1$ **to** m **do**
  if $i < k$ **then** select $\sigma_i \leftarrow_u \{0,1\}$;
  else if $i = k$ **then** $\sigma_k = y_{n(a)+1}$;
  else execute $G_z(x^{(i-1)}) \to_{n(a),1} (x^{(i)}, \sigma_i)$;
end
query $b \leftarrow \mathcal{O}(\sigma_1 \circ \cdots \circ \sigma_m)$;
return b;

We define another auxiliary-input generator $\{G''_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{m(|z|+n(|z|))}\}_{z \in \{0,1\}^*}$ by $G''_z(x) := G'(z \circ x)$. If the given oracle $\mathcal{O}$ $1/\gamma'$-avoids G', then for any $z \in \{0,1\}^a$,

$$\begin{aligned} &\left|\Pr\left[\mathcal{O}(G''_z(U_{n(a)}) = 1\right] - \Pr\left[\mathcal{O}(U_{m(a+n(a))}) = 1\right]\right| \\ &\qquad = \left|\Pr\left[\mathcal{O}(G'(z \circ U_{n(a)})) = 1\right] - \Pr\left[\mathcal{O}(U_{m(a+n(a))}) = 1\right]\right| \geq \frac{1}{\gamma'(a+n(a))}. \end{aligned}$$

By the standard hybrid argument (see e.g., Goldreich, 2006), we have that for any $z \in \{0,1\}^a$,

$$\left|\Pr\left[R^{\mathcal{O}}(z, G_z(U_{n(a)})) = 1\right] - \Pr\left[R^{\mathcal{O}}(z, U_{n(a)+1}) = 1\right]\right| \geq \frac{1}{m(a+n(a)) \cdot \gamma'(a+n(a))}.$$

By taking a polynomial γ satisfying $m(a+n(a)) \cdot \gamma'(a+n(a)) \leq \gamma(n(a))$, the above inequality shows that $R^{\mathcal{O}}$ $1/\gamma$-distinguishes G for any $\mathcal{O}$ $1/\gamma'$-avoiding G'. □

Lemma 3 also implies a nonadaptive BB reduction from distinguishing AIPRG to a distributional NP problem.

Lemma 4. *For any auxiliary-input function G stretching its input and polynomial δ, there exist a language $L \in \mathsf{NP}$, a polynomial γ, and a randomized polynomial-time oracle machine $R^?$ such that for any errorless heuristic oracle $\mathcal{O}$ for (L, U) of failure probability $1/\delta$, $R^{\mathcal{O}}$ $1/\gamma$-distinguishes G. Moreover, $R^?$ is nonadaptive.*

Proof. Define polynomials $\gamma'(n) = \frac{\delta(2n)}{\delta(2n)-2}$ and $m(n) = 2n$. By Lemma 3 for G and m, there exist a polynomial γ and a nonadaptive BB reduction R_1 from $1/\gamma$-distinguishing G to $1/\gamma'$-avoiding G'.

We define the language L in the lemma by $L := \mathrm{Im}G' = \{G'(x) : x \in \{0,1\}^*\}$. Since G' is polynomial-time computable, $L \in \mathsf{NP}$.

Since δ is polynomial, there exists $n_0 \in \mathbb{N}$ such that $2^{n/2} \geq 1/\delta(n)$ for any $n \geq n_0$. Now we construct a nonadaptive BB reduction R_2 from $1/\gamma'$-avoiding G' to an errorless heuristic scheme for (L, U) of failure probability $1/\delta$ as Algorithm 2.

Algorithm 2: R_2 (a nonadaptive BB reduction from avoiding G' to (L, U))

Input : $y \in \{0,1\}^{2n}$
Oracle : $\mathcal{O}$ (an errorless heuristic scheme for (L, U) of failure probability $1/\delta$)

1 **if** $2n < n_0$ **then**
2 | check whether $y \in \mathrm{Im}(G)$ by the brute-force search, if so, **return** 0, otherwise, **return** 1
3 **end**
4 **query** $b \leftarrow \mathcal{O}(y)$;
5 **if** $b \in \{1, \perp\}$ **then return** 0;
6 **else return** 1;

We show that R_2 is a reduction from $1/\gamma'$-avoiding G' to an errorless heuristic scheme for (L, U) of failure probability $1/\delta$. Then by combining R_1 and R_2, we have a nonadaptive BB reduction R from $1/\gamma$-distinguishing G' to an errorless heuristic scheme for (L, U) of failure probability $1/\delta$.

Let $y \in \{0,1\}^{2n}$ be the input for R_2. When $2n < n_0$ holds, R_2 can perfectly determine whether $y \in \mathrm{Im}G'$ and achieve the trivial threshold τ_n in the definition. Therefore, we consider only the case where $2n \geq n_0$.

Assume that the given oracle $\mathcal{O}$ is an errorless heuristic scheme of failure probability at most $1/\delta$, then we have that

$$y \in L\,(= \mathrm{Im}G) \Longrightarrow \mathcal{O}(y) \in \{1, \perp\} \text{ and } \Pr_{y \sim \{0,1\}^{2n}}[\mathcal{O}(y) = \perp] \leq \delta(2n).$$

By the first implication and line 5, when y is generated by G, $R_2^{\mathcal{O}}(y)$ always outputs 0. The upper bound on the probability that $R_2^{\mathcal{O}}$ outputs 0 is given as follows:

$$\begin{aligned}
\Pr_{y \sim \{0,1\}^{2n}}[R^{\mathcal{O}}(y) = 0] &= \Pr_{y \sim \{0,1\}^{2n}}[\mathcal{O}(y) = \perp \text{ or } 1] \\
&\leq \Pr_{y \sim \{0,1\}^{2n}}[\mathcal{O}(y) = \perp] + \Pr_{y \sim \{0,1\}^{2n}}[\mathcal{O}(y) = 1] \\
&\leq \Pr_{y \sim \{0,1\}^{2n}}[\mathcal{O}(y) = \perp] + \Pr_{y}[y \in G(\{0,1\}^n)] \quad (\because \mathcal{O}(y) = 1 \Rightarrow y \in G(\{0,1\}^n)) \\
&\leq \frac{1}{\delta(2n)} + 2^{-n} \leq \frac{2}{\delta(2n)} = 1 - \frac{1}{\gamma'(n)}. \quad (\because 2n \geq n_0)
\end{aligned}$$

□

Lemma 4 and Fact 2 immediately imply Theorem 1 as follows.

Proof of Theorem 1. By Lemma 4, there exist an NP-language L', a polynomial γ, and a nonadaptive BB reduction $R^?$ from $1/\gamma$-distinguishing G to an errorless heuristic scheme for (L', U) of failure probability $1/n$. By combining R with the nonadaptive BB reduction in the assumption from L to $1/\gamma$-distinguishing G, we can construct a nonadaptive BB reduction from L to an errorless heuristic

scheme for (L', U) of failure probability $1/n$. Thus, by Fact 2, the polynomial hierarchy collapses at the third [6] level. □

7 On Basing Auxiliary-Input One-Way Function

In this section, we formally show Theorem 2 based on the idea in Section 4.2.

Theorem 2. *For any auxiliary-input function $f = \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z\in\{0,1\}^*}$ and polynomial p, if there exists a nonadaptive BB reduction from an* NP*-hard language L to $(1-1/p)$-inverting f, then* NP $\not\subseteq$ BPP *also implies that a one-way function exists (via an adaptive BB reduction).*

First, we introduce the following reduction from inverting AIOWF to a distributional NP problem, which immediately follows from Lemma 4 in Section 6.

Lemma 5. *For any auxiliary-input function f and reciprocals δ, δ' of polynomial, there exist an* NP*-language L and a randomized polynomial-time oracle machine $R^?$ such that for any errorless heuristic oracle $\mathcal{O}$ for (L, U) of failure probability δ', $R^{\mathcal{O}}$ $(1-\delta)$-inverting f.*

Proof. The lemma follows from Lemma 4 and the construction of auxiliary-input pseudorandom generator based on an auxiliary-input (weak) one-way function (e.g., Håstad et al., 1999). □

Now we give the full proof of Theorem 2.

Proof of Theorem 2. Let $R_{L\to f}$ be the nonadaptive BB reduction from L to $(1-\delta)$-inverting f. W.l.o.g, we can assume that the failure probability is at most $1/16$ instead of $1/3$ (by taking majority vote of parallel executions) and the distributions on query are identical regardless of the query position (by adapting random permutation before asking them). We can also assume that the running time $t^{R_{L\to f}}(m)$, query complexity $q^{R_{L\to f}}(m)$, and the length of random bits $r^{R_{L\to f}}(m)$ are increasing for the input size m.

By Lemma 5, there exist an NP-language L' and a BB reduction $R_{f\to L'}$ from $(1-\delta)$-inverting f to an errorless heuristic scheme for (L', U) of failure probability δ. Since L' is in NP and L is NP-hard, there exists a Karp reduction $R_{L'\to L}$ from L' to L. W.l.o.g., $|R_{L'\to L}(x)| \le p(|x|)$ for some (increasing) polynomial p.

We define polynomials $q(\cdot)$, $r(\cdot)$, and $a(\cdot)$ as follows:

$$q(m) := q^{R_{L\to f}}(p(m)), \quad r(m) := r^{R_{L\to f}}(p(m)), \quad a(m) := t^{R_{L\to f}}(p(m))$$

On the execution of $R_{L\to f}(R_{L'\to L}(x))$ where $x \in \{0,1\}^m$, the number of queries, the number of random bits, and the length of queries are bounded above by $q(m)$, $r(m)$, and $a(m)$, respectively.

We also define a Turing machine $Q_m : \{0,1\}^m \times \{0,1\}^{r(m)} \to \{0,1\}^{\le a(m)}$ as $Q_m(x, s)$ outputs an auxiliary-input of the first query generated by $R_{L\to f}(R_{L'\to L}(x); s)$.

Now we construct a family of functions $g = \{g_m : \{0,1\}^{m+r(m)+n(a(m))} \to \{0,1\}^*\}_{m\in\mathbb{N}}$ by

$$g_m(x) = \left\langle z, f_z(x^f_{[n(|z|)]}) \right\rangle,$$

where $x \to_{m,r(m),n(a(m))} x^{L'} \circ s \circ x^f$ and $z = Q_m(x^{L'}, s)$.

Since f and Q_m are polynomial-time computable, g is also polynomial-time computable. We will show that if g is not one-way, then NP $\subseteq$ BPP. This immediately yields Theorem 2.

[6] In fact, by more careful simulation technique for HSG by Hirahara and Watanabe (2020), we can improve the consequence on the collapse of polynomial hierarchy at the second level.

For simplicity, we consider that g_m takes as input a triple of length m, $r(m)$, and $N(m) := n(a(m))$, respectively. Assume that g is not one-way. Then there exists a randomized polynomial-time algorithm A such that for any $m \in \mathbb{N}$,

$$\Pr_{A,U_m,U_{r(m)},U_{N(m)}}\left[A(g_m(U_m,U_{r(m)},U_{N(m)})) \notin g_m^{-1}(g_m(U_m,U_{r(m)},U_{N(m)}))\right] \le \frac{\delta(m)\cdot\delta(N(m))}{512\cdot q(m)}.$$

We also define a randomized polynomial-time algorithm A^f by

$$A^f(z,y;s_A) = \begin{cases} x^{(3)}_{[n(|z|)]} & (\text{if } (x^{(1)},x^{(2)},x^{(3)}) \leftarrow A(z,y;s_A) \text{ and } z = Q_m(x^{(1)},x^{(2)})) \\ \bot & (\text{otherwise}). \end{cases}$$

For any $m \in \mathbb{N}$, $x^{L'} \in \{0,1\}^m$, $s \in \{0,1\}^{r(m)}$, $x^f \in \{0,1\}^{N(m)}$, random bits $s_A \in \{0,1\}^*$ for A and A^f, and $z := Q_m(x^{L'},s)$, we have that

$$\begin{aligned}
&A(g_m(x^{L'},s,x^f);s_A) \in g_m^{-1}(g_m(x^{L'},s,x^f)) \\
&\quad\iff g_m(A(g_m(x^{L'},s,x^f);s_A)) = g_m(x^{L'},s,x^f)\ \left(= \left\langle z, f_z(x^f_{[n(|z|)]})\right\rangle\right) \\
&\quad\iff z = Q_m(x^{(1)},x^{(2)}) \text{ and } f_z(x^{(3)}_{[n(|z|)]}) = f_z(x^f_{[n(|z|)]}) \text{ where } (x^{(1)},x^{(2)},x^{(3)}) \leftarrow A(g_m(x^{L'},s,x^f);s_A) \\
&\quad\iff f_z(A^f(g_m(x^{L'},s,x^f);s_A)) = f_z(x^f_{[n(|z|)]}) \\
&\quad\iff A^f(z,f_z(x^f_{[n(|z|)]});s_A) \in f_z^{-1}(f_z(x^f_{[n(|z|)]})). \qquad (1)
\end{aligned}$$

Fix $m \in \mathbb{N}$ arbitrarily. Let $r := r(m)$, $N := N(m)$ and $q := q(m)$. We divide instances on L' into three sets $B_m^{L'}, G_m^{L'}$, and $N_m^{L'}$ (which stand for bad, good, and neutral, respectively) as

$$\begin{aligned}
B_m^{L'} &:= \left\{x \in \{0,1\}^m : \Pr_{A,U_r,U_N}\left[A(g_m(x,U_r,U_N)) \notin g_m^{-1}(g_m(x,U_r,U_N))\right] > \frac{\delta(N)}{216\cdot q}\right\}, \\
G_m^{L'} &:= \left\{x \in \{0,1\}^m : \Pr_{A,U_r,U_N}\left[A(g_m(x,U_r,U_N)) \notin g_m^{-1}(g_m(x,U_r,U_N))\right] \le \frac{\delta(N)}{512\cdot q}\right\}, \\
N_m^{L'} &:= \{0,1\}^m \setminus (B_m^{L'} \cup G_m^{L'}).
\end{aligned}$$

Now fix any (not-bad) instance $x \in G_m^{L'} \cup N_m^{L'}$. We also define good and bad sets on the randomness for A. Let r_A be a polynomial such that $r_A(m)$ is the number of random bits used by A for inverting g_m. Then we define the bad and good sets of randomness of A by

$$\begin{aligned}
B_{m,x}^A &:= \left\{s_A \in \{0,1\}^{r_A(m)} : \Pr_{U_r,U_N}\left[A(g_m(x,U_r,U_N;s_A)) \notin g_m^{-1}(g_m(x,U_r,U_N))\right] > \frac{\delta(N)}{16\cdot q}\right\}, \\
G_{m,x}^A &:= \{0,1\}^{r_A(m)} \setminus B_{m,x}^A.
\end{aligned}$$

For any good random bits $s_A \in G_{m,x}^A$, we define a bad set B_{m,x,s_A}^f on auxiliary-input of f as

$$B_{m,x,s_A}^f := \left\{z \in \{0,1\}^{\le a(m)} : \Pr_{U_{n(|z|)}}\left[A^f(z,f_z(U_{n(|z|)});s_A) \notin f_z^{-1}(f_z(U_{n(|z|)}))\right] > \delta(n(|z|))\right\}.$$

Consider a function $\mathcal{O}_{m,x,s_A} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ by

$$\mathcal{O}_{m,x,s_A}(z,y) = \begin{cases} A^f(z,y;s_A) & z \in \{0,1\}^{\le a(m)} \setminus B_{m,x,s_A}^f \\ x_{z,y} & z \in B_{m,x,s_A}^f \text{ or } z \in \{0,1\}^{>a(m)}, \end{cases}$$

where $x_{z,y}$ is the lexicographically first element of $f_z^{-1}(y)$ if any, otherwise $x_{z,y} = 0$.

First we show that the above $\mathcal{O}_{m,x,s_A}(z,y)$ is indeed a $(1-\delta)$-inverting oracle for f.

Claim 1. *For any $m \in \mathbb{N}$, $x \in G_m^{L'} \cup N_m^{L'}$, and $s_A \in G_{m,x}^A$, $\mathcal{O}_{m,x,s_A}$ $(1-\delta)$-inverts f.*

Proof of Claim 1. Fix $m \in \mathbb{N}$, $x \in G_m^{L'}$, and $s_A \in G_{m,x}^A$ arbitrarily. If $z \in B_{m,x,s_A}^f \cup \{0,1\}^{>a(m)}$, by the definition of $\mathcal{O}_{m,x,s_A}$, $\mathcal{O}_{m,x,s_A}(z,y)$ must output the first inverse element of y if any.

If $z \in \{0,1\}^{\leq a(m)} \setminus B_{m,x,s_A}^f$, then we have that

$$\Pr_{x \sim \{0,1\}^{n(|z|)}}[\mathcal{O}_{m,x,s_A}(z, f_z(x)) \notin f_z^{-1}(f_z(x))] = \Pr_x[A^f(z, f_z(x); s_A) \notin f_z^{-1}(f_z(x))] \leq \delta(n(|z|)),$$

where the last inequality holds because z is not contained in B_{m,x,s_A}^f. □

Note that, by Claim 1, we have that for any $x' \in \{0,1\}^*$,

$$\Pr_{R_{L \to f}}[R_{L \to f}^{\mathcal{O}_{m,x,s_A}}(x') \neq L(x')] \leq 1/16. \tag{2}$$

If we can construct a randomized errorless heuristic scheme B for (L', U) of failure probability at most δ, then B and $R_{f \to L'}$ yield a randomized polynomial-time algorithm $(1-\delta)$-inverting f. By using $R_{L \to f}$, we have also a randomized polynomial-time algorithm for L. Since L is NP-hard, this implies $\mathsf{NP} \subseteq \mathsf{BPP}$. Therefore, the remaining part is to construct the randomized errorless heuristic scheme B.

Now we construct B by using A, $R_{L' \to L}$, and $R_{L \to f}$ as Algorithm 3.

Algorithm 3: B (a randomized errorless heuristic scheme for (L', U))

Input : $x \in \{0,1\}^m$

1 *estimate the failure probability of A*
2 let $c := 0$, $M := \frac{2^{21} \cdot q^2(m)}{\delta^2(N(m))}$;
3 **repeat** M *times* **do**
4 select $s \leftarrow \{0,1\}^{r(m)}$, $x^f \leftarrow_u \{0,1\}^{N(m)}$ and compute $y = g_m(x, s, x^f)$;
5 execute $(\bar{x}^{(1)}, \bar{x}^{(2)}, \bar{x}^{(3)}) \leftarrow A(y)$;
6 **if** $g_m(\bar{x}^{(1)}, \bar{x}^{(2)}, \bar{x}^{(3)})) \neq y$ *(fail in inverting)* **then** $c := c + 1$;
7 **if** $c > M \cdot \frac{3 \cdot \delta(N(m))}{1024 \cdot q(m)}$ **then return** $\perp$;
8 select random bits for A^f as $s_A \leftarrow_u \{0,1\}^{r_A(m)}$;
9 execute $x' \leftarrow R_{L' \to L}(x)$;
10 execute $R_{L \to f}(x')$ where for each query (z, y), answer $A^f(z, y; s_A)$;
11 if $R_{L \to f}(x')$ halts and outputs a value b, then **return** b;

We show that B is a randomized errorless heuristics for (L', U). In the subsequent argument, we use x to denote the input for B. Let $m = |x|$. By Hoeffding inequality, we can show the following claim on lines 1-7.

Claim 2. *1. If $x \in B_m^{L'}$, then $\Pr_B[B(x) = \perp] \geq 15/16$*

2. If $x \in G_m^{L'}$, then $\Pr_B[B(x) = \perp] \leq 1/16$.

Proof of Claim 2. (1) For each i-th trial in line 3, consider a Bernoulli random variable X_i which takes 1 if A fails in inverting g_m, otherwise 0. By the definition of $x \in B_m^{L'}$,

$$\mu := \mathrm{E}[X_i] > \frac{\delta(N(m))}{216 \cdot q}.$$

Therefore, we have that

$$\begin{aligned}\Pr_B[B(x) \neq \perp] &= \Pr\left[\sum_{i=1}^{M} X_i \leq M \cdot \frac{3 \cdot \delta(N(m))}{1024 \cdot q(m)}\right] \\ &\leq \Pr\left[\frac{1}{M}\sum_{i=1}^{M} X_i - \mu \leq -\frac{\delta(N(m))}{1024 \cdot q(m)}\right] \\ &\leq \exp\left(-2 \cdot \frac{\delta^2(N(m))}{2^{20} \cdot q^2(m)} \cdot M\right) = e^{-4} < \frac{1}{16},\end{aligned}$$

where the second inequality follows from the Hoeffding inequality.

(2) We use the same notation about X_i and μ. In the case where $x \in G_m^{L'}$, we have that

$$\mu := \mathrm{E}[X_i] \leq \frac{\delta(N(m))}{512 \cdot q}.$$

Thus, by using the Hoeffding inequality again,

$$\begin{aligned}\Pr_B[B(x) = \perp] &= \Pr\left[\sum_{i=1}^{M} X_i > M \cdot \frac{3 \cdot \delta(N(m))}{1024 \cdot q(m)}\right] \\ &\leq \Pr\left[\frac{1}{M}\sum_{i=1}^{M} X_i - \mu \leq \frac{\delta(N(m))}{1024 \cdot q(m)}\right] \leq \exp\left(-2 \cdot \frac{\delta^2(N(m))}{2^{20} \cdot q^2(m)} \cdot M\right) < \frac{1}{16}.\end{aligned}$$

□

By Claim 2, we can show the following claims:

Claim 3. $\Pr_{x\sim\{0,1\}^m}[x \in B_m^{L'} \cup N_m^{L'}] \leq \delta(m)$.

Claim 4. *If* $x \in G_m^{L'}$, *then* $\Pr_B[B(x) = L'(x)] \geq 3/4$.

Claim 5. *If* $x \in N_m^{L'}$, *then* $\Pr_B[B(x) \in \{L'(x), \perp\}] \geq 3/4$.

Assume that the above three claims hold. Then we can show that B is a randomized errorless heuristic scheme as follows: For the condition on errorless, by Claims 2-(1), 4, and 5, for any instance $x \in \{0,1\}^m$, $B(x) \in \{L'(x), \perp\}$ with probability at least 3/4. For the condition on the failure probability, Claims 2-(1), 4, and 5 imply that $B(x)$ outputs $\perp$ with probability at least 3/4 only if $x \in B_m^{L'} \cup N_m^{L'}$. By Claim 3, the latter event occurs with probability at most $\delta(m)$ over the uniform choice of $x \in \{0,1\}^m$.

Therefore, the remaining part is only to show Claims 3, 4, and 5.

Proof of Claim 3. By the definitions of $B_m^{L'}$ and $N_m^{L'}$,

$$B_m^{L'} \cup N_m^{L'} = \left\{x \in \{0,1\}^m : \Pr_{A,U_r,U_N}\left[A(g_m(x,U_r,U_N)) \notin g_m^{-1}(g_m(x,U_r,U_N))\right] > \frac{\delta(N)}{512 \cdot q}\right\}.$$

Remember that A satisfies

$$\Pr_{A,U_m,U_r,U_N}\left[A(g_m(x,U_r,U_N)) \notin g_m^{-1}(g_m(x,U_r,U_N))\right] \leq \delta(m) \cdot \frac{\delta(N)}{512 \cdot q}.$$

By Lemma 1, we have that $\Pr_{x\sim\{0,1\}^m}[x \in B_m^{L'} \cup N_m^{L'}] \leq \delta(m)$. □

Claims 4 and 5 are immediately implied by the following Claim 6. Therefore, first we show Claims 4 and 5 by assuming Claim 6 then we show Claim 6.

Claim 6. *If* $x \in G_m^{L'} \cup N_m^{L'}$, *then* $\Pr_B[B(x) = \neg L'(x) | B(x) \neq \bot] \leq 3/16$.

Proof of Claim 4. If $x \in G_m^{L'}$, then

$$\begin{aligned}\Pr_B[B(x) \neq L'(x)] &= \Pr_B[B(x) = \neg L'(x) \text{ or } B(x) = \bot] \\ &= \Pr_B[B(x) = \bot] + \Pr_B[B(x) = \neg L'(x) | B(x) \neq \bot] \\ &\leq 1/16 + 3/16 = 1/4,\end{aligned}$$

where the last inequality follows from Claims 2-(2) and 6. □

Proof of Claim 5. If $x \in N_m^{L'}$, then

$$\begin{aligned}\Pr_B[B(x) \in \{L'(x), \bot\}] &= \Pr_B[B(x) = \bot] + \Pr_B[B(x) = L'(x) | A(x) \neq \bot] \\ &\geq \Pr_B[B(x) = L'(x) | B(x) \neq \bot] \geq 13/16 > 3/4,\end{aligned}$$

where the last inequality follows from Claim 6. □

Proof of Claim 6. By the assumption that $x \in G_m^{L'} \cup N_m^{L'}$, we have that

$$\Pr_{A, U_r, U_N}\left[A(g_m(x, U_r, U_N)) \notin g_m^{-1}(g_m(x, U_r, U_N))\right] \leq \frac{\delta(N)}{216 \cdot q}.$$

By Lemma 1,

$$\Pr_{s_A \sim \{0,1\}^{r_A}}[s_A \in B_{m,x}^A] \leq \frac{1}{16}. \tag{3}$$

Therefore, we assume that B succeeds in selecting a good $s_A \in G_{m,x}^A$. By Claim 1, if B could simulate $\mathcal{O}_{m,x,s_A}$ in line 10 instead of $A^f(\,\cdot\,; s_A)$, then $R_{L \to f}$ can recognize $L'(x)$ with high probability. As shown below, however, answer by $A^f(\,\cdot\,; s_A)$ is consistent with answer by $\mathcal{O}_{m,x,s_A}$ with high probability over the choice of random bits for $R_{L \to f}$.

Let (z, y) be a query generated by $R_{L \to f}$. By the definition of $\mathcal{O}_{m,x,s_A}$, $A^f(z, y; s_A)$ is inconsistent with $\mathcal{O}_{m,x,s_A}(z, y)$ only if (a) $z \in B_{m,x,s_a}^f$ or (b) $|z| > a(m)$. Since $a(m)$ is the upper bound on the length of queries by $R_{L \to f}(R_{L' \to L}(x))$, the latter case (b) never occurs.

Thus, we bound above on the probability that the event (a) occurs. For the choice of the randomness $s \in \{0,1\}^{r(m)}$ to execute $R_{L \to f}$, define a bad set $B_{m,x,s_A}^{R_{L \to f}}$ by

$$B_{m,x,s_A}^{R_{L \to f}} := \left\{ s \in \{0,1\}^{r(m)} : \Pr_{U_{N(m)}}\left[A(g_m(x, s, U_{N(m)}; s_A)) \notin g_m^{-1}(g_m(x, s, U_{N(m)}))\right] > \delta(N(m)) \right\}.$$

Since $s_A \in G_{m,x}^A$, we have that

$$\Pr_{U_{r(m)}, U_{N(m)}}\left[A(g_m(x, U_{r(m)}, U_{N(m)}; s_A)) \notin g_m^{-1}(g_m(x, U_{r(m)}, U_{N(m)}))\right] \leq \frac{\delta(N(m))}{16 \cdot q(m)}$$

By Lemma 1,

$$\Pr_{s \sim \{0,1\}^{r(m)}}\left[s \in B_{m,x,s_A}^{R_{L \to f}}\right] \leq \frac{1}{16 \cdot q(m)}.$$

We define the event E_x over the choice of random bits for $R_{L\to f}$ by

$$E_x := \left(R_{L\to f}(R_{L'\to L}(x)) \text{ makes the first query } (z,y) \text{ such that } z \in B^f_{m,x,s_a}\right).$$

Then by the definitions of Q_m and g_m,

$$\begin{aligned}
\Pr_{R_{L\to f}}[E_x] &= \Pr_{s\sim\{0,1\}^{r(m)}}[Q_m(x,s) \in B^f_{m,x,s_A}] \\
&\le \Pr_{s\sim\{0,1\}^{r(m)}}\left[z \leftarrow Q_m(x,s);\ \Pr_{U_{n(|z|)}}\left[A^f(z, f_z(U_{n(|z|)}); s_A) \notin f_z^{-1}(f_z(U_{n(|z|)}))\right] > \delta(n(|z|))\right] \\
&\le \Pr_{s\sim\{0,1\}^{r(m)}}\left[\Pr_{U_{N(m)}}\left[A(g_m(x,s,U_{N(m)}); s_A) \notin g_m^{-1}(g_m(U_{n(m)}))\right] > \delta(N(m))\right] \quad (\because (1)) \\
&= \Pr_{s\sim\{0,1\}^{r(m)}}\left[s \in B^{R_{L\to f}}_{m,x,s_A}\right] \le \frac{1}{16\cdot q(m)}.
\end{aligned}$$

Since each query distribution by $R_{L\to f}$ is identical to the first query distribution, by the union bound, we have that

$$\begin{aligned}
&\Pr_{R_{L\to f}}[\text{the event (a) occurs}] \\
&\quad = \Pr_{R_{L\to f}}[R_{L\to f}(R_{L'\to L}(x)) \text{ makes at least one query } (z,y) \text{ such that } z \in B^f_{m,x,s_a}] \\
&\quad \le q(m)\cdot \Pr_{R_{L\to f}}[E_x] \le q(m)\cdot\frac{1}{16\cdot q(m)} = \frac{1}{16}.
\end{aligned}$$

Therefore, we have that

$$\Pr_{R_{L\to f}}[R^{\mathcal{O}_{m,x,s_A}}_{L\to f}(R_{L'\to L}(x)) \neq R^{A^f(\cdot;s_A)}_{L\to f}(R_{L'\to L}(x))] \le \Pr_{R_{L\to f}}[\text{the event (a) occurs}] \le 1/16. \tag{4}$$

Since $R_{L'\to L}$ is a Karp reduction from L' to L, $L'(x) = L(R_{L'\to L}(x))$ holds. By the inequality (2),

$$\Pr_{R_{L\to f}}[R^{\mathcal{O}_{m,x,s_A}}_{L\to f}(R_{L'\to L}(x)) \neq L'(x)] \le 1/16. \tag{5}$$

By the union bound, we conclude that (under the condition that B does not output $\perp$ in line 7)

$$\begin{aligned}
\Pr_B[B(x) \neq L'(x)] &\le \Pr_{s_A}[s_A \in B^A_{m,x}] + \Pr_{s_A,R_{L\to f}}[R^{A^f(\cdot;s_A)}_{L\to f}(R_{L'\to L}(x)) \neq L'(x) | s_A \notin B^A_{m,x}] \\
&\le \Pr_{s_A}[s_A \in B^A_{m,x}] + \Pr_{s_A,R_{L\to f}}[R^{\mathcal{O}_{m,x,s_A}}_{L\to f}(R_{L'\to L}(x)) \neq L'(x) | s_A \notin B^A_{m,x}] \\
&\qquad + \Pr_{s_A,R_{L\to f}}[R^{A^f(\cdot;s_A)}_{L\to f}(R_{L'\to L}(x)) \neq R^{\mathcal{O}_{m,x,s_A}}_{L\to f}(R_{L'\to L}(x)) | s_A \notin B^A_{m,x}] \\
&\le \frac{1}{16} + \frac{1}{16} + \frac{1}{16} = \frac{3}{16}.
\end{aligned}$$

where the last inequality follows from inequalities (3), (4), and (5). □

□

8 On Basing Auxiliary-Input Hitting Set Generator

In this section, we formally show Theorem 3 based on the idea in Section 4.3.

Theorem 3. *Let p be a polynomial and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function where $\ell(n) > (1+\epsilon)\cdot n$ for some constant $\epsilon > 0$. If there exists a nonadaptive BB reduction from an NP-hard language L to $(1-1/p)$-avoiding G, then $\mathsf{NP} \nsubseteq \mathsf{BPP}$ also implies that a one-way function exists (via an adaptive BB reduction).*

Theorem 3 obviously follows from Lemma 6 and Theorem 2.

Lemma 6. *Let δ be a reciprocal of polynomial and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ be an auxiliary-input function where $\ell(n) > (1+\epsilon)\cdot n$ for some constant $\epsilon > 0$. If there exists a nonadaptive BB reduction from an NP-hard language L to $(1-\delta)$-avoiding G, then there exist another auxiliary-input function f and a reciprocal δ' of polynomial such that there exists a nonadaptive BB reduction from L to $(1-\delta')$-inverting f.*

Proof of Lemma 6. Let $\epsilon' = \epsilon/2$ and $R^?$ be the nonadaptive BB reduction from L to $(1-\delta)$-avoiding G. W.l.o.g, we can assume that $R^?$ makes $q(m)$ queries on input $x \in \{0,1\}^m$ where q is polynomial and all $q(m)$ distributions on query generated by R are identical regardless of query position by applying a random permutation before asking them.

Fix input $x \in \{0,1\}^m$ arbitrarily. Let Q_x be the distribution on the first query by $R(x)$ (which is identical to query distributions in other query positions). Let $Q_x^{(1)}$ be the distribution on auxiliary-input of Q_x.

Fix a length $a \in \mathbb{N}$ of auxiliary-input arbitrarily. Let $n := n(a)$ and $\ell := \ell(n)$. We divide possible queries into three sets H_x, L_x and M_x (which stand for heavy, light, and medium, respectively) as follows:

$$
\begin{aligned}
H_x &:= \left\{(z,y) \in \{0,1\}^n \times \{0,1\}^\ell : p_x(y|z) > \frac{64}{2^{(1+\epsilon')n}}\right\},\\
L_x &:= \left\{(z,y) \in \{0,1\}^n \times \{0,1\}^\ell : p_x(y|z) \le \frac{1}{2^{(1+\epsilon')n}}\right\},\\
M_x &:= \left(\{0,1\}^n \times \{0,1\}^\ell\right) \setminus (H_x \cup L_x),
\end{aligned}
$$

where $p_x(y|z) = \Pr[(z,y) \leftarrow Q_x | z \leftarrow Q_x^{(1)}]$.

Now we define a set $\mathcal{T}_{x,a}$ composed of all statistical tests $T : \{0,1\}^a \times \{0,1\}^\ell \to \{0,1\}$ satisfying following conditions: for any $z \in \{0,1\}^a$,

1. $y \in \mathrm{Im}(G_z) \implies T(z,y) = 0$
2. $(y \notin \mathrm{Im}(G_z) \ \wedge\ (z,y) \in H_x) \implies T(z,y) = 0$
3. $(y \notin \mathrm{Im}(G_z) \ \wedge\ (z,y) \in L_x) \implies T(z,y) = 1$

Since $\delta(n)$ is a reciprocal of polynomial, $-\log\delta(n) = O(\log n)$. Therefore, there exists $n_0 \in \mathbb{N}$ such that for any $n \ge n_0$, $n \ge \frac{1}{\epsilon'}(1 - \log\delta(n))$ holds. In the following claim, we show that each element in $\mathcal{T}_{x,a}$ avoids G_z for large enough a.

Claim 7. *For any $x \in \{0,1\}^m$ and $a \in \mathbb{N}$, if $n(a) \ge n_0$, then any $T \in \mathcal{T}_{x,a}$ $(1-\delta)$-avoids G_z for any $z \in \{0,1\}^a$*

Proof. Fix $T \in \mathcal{T}_{x,a}$ arbitrarily. Since T satisfies the condition 1, we have that $T(z,y) = 0$ for any $z \in \{0,1\}^a$ and $y \in \mathrm{Im}(G_z)$. Thus, it is enough to show that

$$\Pr_{y \sim \{0,1\}^{\ell(n(a))}}[T(z,y) = 0] \leq \delta(n(a)).$$

Since T also satisfies the condition 3,

$$\begin{aligned}
\Pr_{y \sim \{0,1\}^{\ell}}[T(z,y) = 0] &\leq \Pr_{y \sim \{0,1\}^{\ell}}[y \in \mathrm{Im}(G_z) \ \vee \ (y,z) \notin L_x] \\
&\leq \Pr_{y \sim \{0,1\}^{\ell}}[y \in \mathrm{Im}(G_z)] + \Pr_{y \sim \{0,1\}^{\ell}}[(y,z) \in H_x \cup M_x] \\
&\leq \frac{2^n}{2^\ell} + \Pr_{y \sim \{0,1\}^{\ell}}[(y,z) \in H_x \cup M_x].
\end{aligned}$$

Notice that if

$$\left|\left\{y \in \{0,1\}^\ell : p_x(y|z) > 2^{-(1+\epsilon')n}\right\}\right| > 2^{(1+\epsilon')n},$$

then,

$$1 = \sum_{y \in \{0,1\}^\ell} p_x(y|z) \geq \sum_{\substack{y \in \{0,1\}^\ell : \\ (y,z) \in H_x \cup M_x}} p_x(y|z) > 2^{-(1+\epsilon')n} \cdot 2^{(1+\epsilon')n} = 1.$$

Hence, we have that

$$\left|\left\{y \in \{0,1\}^\ell : (y,z) \in H_x \cup M_x\right\}\right| = \left|\left\{y \in \{0,1\}^\ell : p_x(y|z) > 2^{-(1+\epsilon')n}\right\}\right| \leq 2^{(1+\epsilon')n}.$$

Therefore,

$$\begin{aligned}
\Pr_{y \sim \{0,1\}^{\ell}}[T(z,y) = 0] &\leq \frac{2^n}{2^\ell} + \Pr_{y \sim \{0,1\}^{\ell}}[(y,z) \in H_x \cup M_x] \\
&\leq \frac{2^n}{2^\ell} + \frac{2^{(1+\epsilon')n}}{2^\ell} \\
&\leq \frac{2^n(1+2^{\epsilon' n})}{2^{(1+2\epsilon')n}} \leq \frac{2^{\epsilon' n+1}}{2^{2\epsilon' n}} = \frac{2}{2^{\epsilon' n}} \leq \delta(n(a)). \quad (\because n(a) \geq n_0)
\end{aligned}$$

□

For $x \in \{0,1\}^m$ and a family of statistical tests $\{T_a\}_{a \in \mathbb{N}}$ where $T_a \in \mathcal{T}_{x,a}$, we define a function $\mathcal{O}_{\{T_a\}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ by

$$\mathcal{O}_{\{T_a\}}(z,y) = \begin{cases} 1 & \text{if } |y| \neq \ell(n(|z|)) \\ \mathbb{1}\{y \in \mathrm{Im}(G_z)\} & \text{if } |y| = \ell(n(|z|)) \ \wedge \ (2^{\epsilon' n(|z|)} \leq 2^{10} \cdot q(m) \ \vee \ n(|z|) < n_0) \\ T_{|z|}(z,y) & \text{otherwise.} \end{cases}$$

We define sets $\mathcal{T}_x$ and $\mathcal{T}$ of functions by $\mathcal{T}_x := \left\{\mathcal{O}_{\{T_a\}_{a \in \mathbb{N}}} : T_a \in \mathcal{T}_{x,a}\right\}$ and $\mathcal{T} = \bigcup_{x \in \{0,1\}^*} \mathcal{T}_x$. Then Claim 7 implies that any $\mathcal{O} \in \mathcal{T}$ $(1-\delta)$-inverts G and thus, for any $x \in \{0,1\}^*$,

$$\Pr_R[R^{\mathcal{O}}(x) \neq L(x)] \leq 1/3. \tag{6}$$

We can assume that $R^?(m)$ uses at most $r(m)$ random bits for any $m \in \mathbb{N}$ to create its $q(m)$ queries, where $r(\cdot)$ is a polynomial. Let $s(\cdot)$ be a polynomial satisfying that for any $m \in \mathbb{N}$ and

$x \in \{0,1\}^m$, the first query by $R^?(x)$ is generated by an $s(r(m))$-size circuit which takes $r(m)$ random bits as input.

We construct a randomized polynomial-time algorithm A for L as Algorithm 4 by using the universal extrapolation algorithm $Ext_{s(n)}$ with $\epsilon = 1$ and $\delta(r(m)) = \frac{1}{32 \cdot q(m)}$ in Lemma 2. Remark that A uses $Ext_{s(n)}$ nonadaptively (in line 3). Since $Ext_{s(n)}$ uses an inverting oracle for a certain auxiliary-input function f, this yields a nonadaptive BB reduction from L to inverting f.

Algorithm 4: A (a randomized algorithm for L)

Input : $x \in \{0,1\}^m$

1 execute $R^?(x)$ and make $q(m)$ queries $(z_1, y_1), \ldots, (z_{q(m)}, y_{q(m)})$;
2 embed x to $R^?$ and create $s(r(m))$-size circuits $C_x(r)$ and $C_x^{(1)}(r)$ generating the first query and the auxiliary-input in the first query of $R^?(x; r)$, respectively;
3 execute $\tilde{p}_i \leftarrow Ext_{s(n)}(C_x, (z_i, y_i))$ and $\tilde{p}'_i \leftarrow Ext_{s(n)}(C_x^{(1)}, z_i)$ for each $i \in [q(m)]$;
4 **for** $i := 1$ **to** $q(m)$ **do**
5 let $n_i := n(|z_i|)$;
6 *answer the i-th query (z_i, y_i) as follows:*
7 **if** $\exists j < i$ *such that* $(z_j, y_j) = (z_i, y_i)$ **then** return the same answer as the j-th query;
8 **else if** $n_i < n_0$ *or* $2^{\epsilon' n_i} \leq 2^{10} \cdot q(m)$ **then** find the answer by brute-force search and return it (note that the latter condition implies $2^{n_i} \leq (2^{10} \cdot q(m))^{1/\epsilon'} \leq \mathsf{poly}(m)$);
9 **else if** $\frac{\tilde{p}_i}{\tilde{p}'_i} \leq \frac{8}{2^{(1+\epsilon')n_i}}$ **then** return 1;
10 **else** return 0;
11 **if** $R^?(x)$ *halts and outputs* $b \in \{0,1\}$ **then return** the same value b;

We will show that A indeed solves L. It is not hard to see that A is polynomial-time computable and executes $Ext_{s(n)}$ $2q(m)$ times for the input of size m. Since the failure probability of each execution is at most $\frac{1}{32 \cdot q(m)}$, the probability that at least one of the executions fails is at most $1/16$.

We assume that all executions of $Ext_{s(n)}$ will not fail. For $x \in \{0,1\}^*$ and $a \in \mathbb{N}$, we define a set $\mathcal{T}'_{x,a}$ composed of all statistical tests $T' : \{0,1\}^a \times \{0,1\}^{\ell(n(a))} \to \{0,1\}$ satisfying the followings: for any $z \in \{0,1\}^a$,

i. $(z, y) \in H_x \Longrightarrow T'(z, y) = 0$;

ii. $(z, y) \in L_x \Longrightarrow T'(z, y) = 1$;

For a family of statistical tests $\{T'_a\}_{a \in \mathbb{N}}$ where $T'_a \in \mathcal{T}'_{x,a}$, we define a function $\mathcal{O}_{\{T'_a\}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ in the same way as $\mathcal{T}_{x,a}$. We also define the sets $\mathcal{T}'_x$ and $\mathcal{T}'$ of functions by $\mathcal{T}'_x := \left\{\mathcal{O}_{\{T'_a\}_{a \in \mathbb{N}}} : T'_a \in \mathcal{T}'_{x,a}\right\}$ and $\mathcal{T}' = \bigcup_{x \in \{0,1\}^*} \mathcal{T}'_x$.

By the correctness of $Ext_{s(n)}$, we have that for each $i \in [q(m)]$,

$$\frac{1}{8} \cdot p_x(y_i|z_i) \leq \frac{\frac{1}{2} \cdot \Pr[(z_i, y_i) \leftarrow Q_x]}{4 \cdot \Pr[z_i \leftarrow Q_x^{(1)}]} \leq \frac{\tilde{p}_i}{\tilde{p}'_i} \leq \frac{4 \cdot \Pr[(z_i, y_i) \leftarrow Q_x]}{\frac{1}{2} \cdot \Pr[z_i \leftarrow Q_x^{(1)}]} \leq 8 \cdot p_x(y_i|z_i).$$

Therefore,

$$(y_i, z_i) \in L_x \Longrightarrow \frac{\tilde{p}_i}{\tilde{p}'_i} \leq 8 \cdot p_x(y_i|z_i) \leq \frac{8}{2^{(1+\epsilon')n}},$$

and

$$(y_i, z_i) \in H_x \Longrightarrow \frac{\tilde{p}_i}{\tilde{p}'_i} \geq \frac{1}{8} \cdot p_x(y_i|z_i) > \frac{8}{2^{(1+\epsilon')n}}.$$

Hence, for any $x \in \{0,1\}^*$, $A(x)$ answers each query of $R^?(x)$ by some oracle $\mathcal{O}'$ in $\mathcal{T}'_x$ unless $Ext_{s(n)}$ will not fail. Thus, if the values of $\mathcal{O}'$ is consistent with some $\mathcal{O} \in \mathcal{T}$, then $A(x)$ can correctly simulate $(1-\delta)$-inverting oracle for G. This motivates us to show the following claim.

Claim 8. *For any $m \in \mathbb{N}$, $x \in \{0,1\}^m$, and $\mathcal{O}' \in \mathcal{T}'_x$, there exists $\mathcal{O} \in \mathcal{T}_x$ such that*

$$\Pr_R\left[R^{\mathcal{O}'}(x) \neq R^{\mathcal{O}}(x)\right] \leq \frac{1}{16}.$$

First, we assume that Claim 8 holds. Notice that if the following three events occur on the execution of $A(x)$, then $A(x)$ outputs $L(x)$ correctly:

1. $Ext_{s(n)}$ does not fail, that is, A simulates some oracle $\mathcal{O}' \in \mathcal{T}'_x$;
2. $R^{\mathcal{O}'}(x) = R^{\mathcal{O}}(x)$ where $\mathcal{O} \in \mathcal{T}_x$ is the oracle in Claim 8;
3. $R^{\mathcal{O}}(x) = L(x)$;

By Claim 8 and inequality (6), the probability that each of events 1–3 does not occur is at most $1/16$, $1/16$, and $1/3$, respectively. Therefore, for any $x \in \{0,1\}^*$,

$$\Pr_A[A(x) \neq L(x)] \leq \frac{1}{16} + \frac{1}{16} + \frac{1}{3} = \frac{11}{24} < \frac{1}{2}.$$

Thus, the remaining part is to show Claim 8.

Proof of Claim 8. Fix $x \in \{0,1\}^m$ and $\mathcal{O}' \in \mathcal{T}'_x$ arbitrarily. By the definition of $\mathcal{T}'_x$, there exists a family of statistical tests $\{T'_a\}_{a\in\mathbb{N}}$ where $T'_a \in \mathcal{T}'_{x,a}$ such that $\mathcal{O}' \equiv \mathcal{O}'_{\{T'_a\}}$.

For each $a \in \mathbb{N}$, we define a statistical test $T_a : \{0,1\}^a \times \{0,1\}^{\ell(n(a))} \to \{0,1\}$ by

$$T_a(z,y) = \begin{cases} 0 & \text{if } y \in \mathrm{Im}(G_z) \\ T'_a(z,y) & \text{otherwise.} \end{cases}$$

We also define $\mathcal{O} := \mathcal{O}_{\{T_a\}}$. It is easily checked that $T_a \in \mathcal{T}_{x,a}$. Thus, $\mathcal{O} \in \mathcal{T}_x$.

We have that

$$\Pr_R\left[R^{\mathcal{O}'}(x) \neq R^{\mathcal{O}}(x)\right] \leq \Pr_R\left[R^?(x) \text{ queries } (z,y) \text{ such that } \mathcal{O}(z,y) \neq \mathcal{O}'(z,y)\right].$$

Thus, we will bound the latter probability above by $1/16$.

$$\begin{aligned}
&\mathcal{O}(z,y) \neq \mathcal{O}'(z,y) \\
&\quad\Longrightarrow 2^{\epsilon' n(|z|)} > 2^{10} \cdot q(m) \text{ and } T_{|z|}(z,y) \neq T'_{|z|}(z,y) \quad (\because \text{definitions of } \mathcal{O} \text{ and } \mathcal{O}') \\
&\quad\Longleftrightarrow 2^{\epsilon' n(|z|)} > 2^{10} \cdot q(m) \text{ and } y \in \mathrm{Im}(G_z) \text{ and } T'_{|z|}(z,y) = 1 \quad (\because \text{definitions of } T_a \text{ and } T'_a) \\
&\quad\Longrightarrow 2^{\epsilon' n(|z|)} > 2^{10} \cdot q(m) \text{ and } y \in \mathrm{Im}(G_z) \text{ and } (z,y) \notin H_x \quad (\because \text{definition of } \mathcal{T}'_x).
\end{aligned}$$

For each position $j \in [q(m)]$,

$$
\begin{aligned}
&\Pr_R\left[R^?(x) \text{ queries } (z,y) \text{ such that } \mathcal{O}(z,y) \neq \mathcal{O}'(z,y) \text{ at the } j\text{-th query}\right] \\
&= \Pr_{(z,y)\leftarrow Q_x}\left[\mathcal{O}(z,y) \neq \mathcal{O}'(z,y)\right] \\
&\leq \Pr_{(z,y)\leftarrow Q_x}\left[2^{\epsilon' n(|z|)} > 2^{10}\cdot q(m) \text{ and } y \in \mathrm{Im}(G_z) \text{ and } (z,y) \notin H_x\right] \\
&= \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} \Pr_{(z',y)\leftarrow Q_x}\left[y \in \mathrm{Im}(G_{z'}) \text{ and } (z',y) \in L_x \cup M_x | z' = z\right] \cdot \Pr_{z'\leftarrow Q_x^{(1)}}[z' = z] \\
&\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} \sum_{\substack{y\in\mathrm{Im}(G_z):\\ (z,y)\in L_x\cup M_x}} \Pr_{(z',y')\leftarrow Q_x}\left[y' = y | z' = z\right] \cdot \Pr_{z'\leftarrow Q_x^{(1)}}[z' = z] \\
&= \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} \sum_{\substack{y\in\mathrm{Im}(G_z):\\ (z,y)\in L_x\cup M_x}} p_x(y|z) \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] \\
&\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} \sum_{\substack{y\in\mathrm{Im}(G_z):\\ (z,y)\in L_x\cup M_x}} \frac{64}{2^{(1+\epsilon')n(|z|)}} \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] \quad (\because (z,y) \in L_x \cup M_x) \\
&\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} |\mathrm{Im}(G_z)| \cdot \frac{64}{2^{(1+\epsilon')n(|z|)}} \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] \\
&\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} \frac{64}{2^{\epsilon' n(|z|)}} \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] \quad (\because |\mathrm{Im}(G_z)| \leq 2^{n(|z|)}) \\
&\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{10}\cdot q(m)}} \frac{1}{16\cdot q(m)} \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] \quad (\because 2^{\epsilon' n(|z|)} > 2^{10}\cdot q(m)) \\
&\leq \frac{1}{16\cdot q(m)} \cdot \sum_{z\in\{0,1\}^*} \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] = \frac{1}{16\cdot q(m)}.
\end{aligned}
$$

By union bound, we conclude that

$$
\Pr_R\left[R^?(x) \text{ queries } (z,y) \text{ such that } \mathcal{O}(z,y) \neq \mathcal{O}'(z,y)\right] \leq \frac{q(m)}{16\cdot q(m)} = \frac{1}{16}.
$$

□

□

9 Oracle Separation between AIOWF and OWF

To show Theorem 4, we introduce the auxiliary-input analog of the random function oracle.

Definition 12 (Auxiliary-input embedded random function)**.** *Let r, r' be random values independently selected according to the uniform distribution over $[0,1]$. For each $n \in \mathbb{N}$, we define a target*

auxiliary-input $z_n \in \{0,1\}^n$ *by letting* i*-th bit of* z_n *be* $(\frac{n(n-1)}{2}+i)$*-th bit of the binary representation of* r*. We define an auxiliary-input embedded random function* $\mathcal{F} = \{\mathcal{F}_z : \{0,1\}^{|z|} \to \{0,1\}^{|z|}\}_{z\in\mathbb{N}}$ *by*

$$\mathcal{F}_z(x)_i = \begin{cases} \big((2\cdot(2^n-2+x_{\mathbb{N}})+1)\cdot 2^i\big)\text{-th bit of the binary representation of } r' & \text{if } z = z_n \\ 0 & \text{if } z \neq z_n. \end{cases}$$

Note that the position $(2\cdot(2^n-2+x_{\mathbb{N}})+1)\cdot 2^i$ is uniquely determined by $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and $i \in [n]$ because any integer is uniquely expressed as $(2m_1+1)\cdot 2^{m_2}$ for $m_1, m_2 \in \mathbb{N}\cup\{0\}$.

On access to $\mathcal{F}$ as an oracle, we assume that $\mathcal{F}(x)$ returns a default value 0 for invalid input x which does not take the form of $\langle z, x'\rangle$ with $|z| = |x'|$. Note that we express a random choice of (r, r') over $[0,1]^2$ as a random choice of an auxiliary-input embedded random function $\mathcal{F}$.

For each choice of auxiliary-input embedded random function $\mathcal{F}$, we define an oracle $\mathcal{O}_{\mathcal{F}} : \{0,1\}^* \to \{0,1\}^*$ by

$$\mathcal{O}_{\mathcal{F}}(x) = \begin{cases} 0 & \text{if } |x| = 1 \\ \mathcal{F}(x') & \text{if } x = 0\circ x' \\ \mathsf{TQBF}(x') & \text{if } x = 1\circ x'. \end{cases}$$

For AIOWF, we can show the following lemma, which shows the intractability of inverting an auxiliary-input embedded random function.

Lemma 7. *With probability* 1 *over the choice of an auxiliary-input embedded random function* $\mathcal{F}$*, all randomized polynomial-time oracle machines* $A^?$ *and* $c \in \mathbb{N}$ *satisfy that for any sufficiently large* $n \in \mathbb{N}$*,*

$$\Pr_{A,U_n}[A^{\mathcal{O}_{\mathcal{F}}}(z_n, \mathcal{F}_{z_n}(U_n)) \in \mathcal{F}_{z_n}^{-1}(\mathcal{F}_{z_n}(U_n))] < n^{-c},$$

where $z_n \in \{0,1\}^n$ *is a target auxiliary-input of* $\mathcal{F}$*.*

The above lemma is shown essentially by the same argument by Rudich (1988:Section 6). For completeness, we will give the full proof of Lemma 7 later.

On the other hand, we can show the following lemma for OWF.

Lemma 8. *With probability* 1 *over the choice of an auxiliary-input embedded random function* $\mathcal{F}$*, for any polynomial-time oracle machine* $F^?$*, there exists a polynomial-time oracle machine* $A^?$ *such that for sufficiently large* $n \in \mathbb{N}$*,*

$$\Pr_{U_n}\left[A^{\mathcal{O}_{\mathcal{F}}}(F^{\mathcal{O}_{\mathcal{F}}}(U_n)) \in {F^{\mathcal{O}_{\mathcal{F}}}}^{-1}(F^{\mathcal{O}_{\mathcal{F}}}(U_n))\right] \geq 1 - \frac{2}{n}.$$

First we give the proof of Theorem 4 by assuming Lemmas 7 and 8.

Proof of Theorem 4. With probability 1 over the choice of an auxiliary-input random embedded function $\mathcal{F}$, $\mathcal{O}_{\mathcal{F}}$ satisfies both of conditions in Lemmas 7 and 8. Now we fix such an oracle $\mathcal{O}_{\mathcal{F}}$.

First we define an auxiliary-input function $f^{\mathcal{O}_{\mathcal{F}}} = \{f_z^{\mathcal{O}_{\mathcal{F}}} : \{0,1\}^{|z|} \to \{0,1\}^{|z|}\}_{z\in\{0,1\}^*}$ by

$$f_z^{\mathcal{O}_{\mathcal{F}}}(x) = \mathcal{O}_{\mathcal{F}}(0\circ\langle z, x\rangle)\,(= \mathcal{F}_z(x))\,.$$

Lemma 8 shows that any polynomial-time oracle machine $A^{\mathcal{O}_{\mathcal{F}}}$ cannot invert $\{f_{z_n}^{\mathcal{O}_{\mathcal{F}}}\}_{n\in\mathbb{N}}$ with non-negligible probability. Thus, $f^{\mathcal{O}_{\mathcal{F}}}$ is indeed an auxiliary-input one-way function relative to $\mathcal{O}_{\mathcal{F}}$.

On the other hand, we show that there is no one-way function relative to $\mathcal{O}_{\mathcal{F}}$. For contradiction, assume that there exists a one-way function $f'^{\mathcal{O}_{\mathcal{F}}}$ and let $F^{\mathcal{O}_{\mathcal{F}}}$ be an oracle machine which computes $f'^{\mathcal{O}_{\mathcal{F}}}$ in polynomial-time. By Lemma 8, even for this F, there exists a polynomial-time adversary $A^{\mathcal{O}_{\mathcal{F}}}$ which $(1-2/n)$-inverts $f'^{\mathcal{O}_{\mathcal{F}}}$. This contradicts the assumption that $f'^{\mathcal{O}_{\mathcal{F}}}$ is one-way. Thus, there is no one-way function relative to $\mathcal{O}_{\mathcal{F}}$. □

Now we give the proof of Lemma 7.

Proof of Lemma 7. In this proof, we use the notation poly to denote a certain polynomial. For simplicity, we regard the oracle access to $\mathcal{O}_{\mathcal{F}}$ as oracle access to two oracles $\mathcal{F}$ and TQBF.

Fix a polynomial-time oracle machine $A^?$ arbitrarily. For any $n \in \mathbb{N}$, $y \in \{0,1\}^n$, and random tape $s \in \{0,1\}^{\mathsf{poly}(n)}$ for $A^?(z,y)$, define a bad event B_y that there are at least $n+1$ elements $x \in \{0,1\}^n$ satisfying $\mathcal{F}_{z_n}(x) = y$ over the choice of $\mathcal{F}$. Then by simple calculation, $\Pr_{\mathcal{F}}[B_y] \leq C \cdot 2^{-n}$ for some constant C (as shown in the last of the proof).

Let $q(n)$ be a polynomial of query complexity of $A^?$. Now we consider the random choice of $\mathcal{F}$ under the condition of $\neg B_y$. Since $\{\mathcal{F}_{z_n}\}_{n\in\mathbb{N}}$ is a random function selected independently of TQBF, the access to TQBF will not reveal any information about $\mathcal{F}$. Thus, for any i-th query where $i \in [q(n)]$, $A^{\mathsf{TQBF},\mathcal{F}}$ queries unknown inverse element x satisfying $\mathcal{F}_{z_n}(x) = y$ to $\mathcal{F}_{z_n}$ with probability at most

$$\frac{|\{x : \mathcal{F}_{z_n}(x) = y\}|}{2^n - (i-1)} \leq \frac{n}{2^n - q(n) + 1}.$$

By union bound, the probability that $A^{\mathsf{TQBF},\mathcal{F}}$ queries at least one (z_n, x) satisfying $\mathcal{F}_{z_n}(x) = y$ is at most $\frac{n \cdot q(n)}{2^n - q(n) + 1}$. In this case, the probability that $A^{\mathsf{TQBF},\mathcal{F}}(z_n, y)$ outputs x satisfying $\mathcal{F}_{z_n}(x) = y$ is at most $\frac{n}{2^n - q(n)}$. Therefore, for any $y \in \{0,1\}^n$ and $s \in \{0,1\}^{\mathsf{poly}(n)}$,

$$\Pr_{\mathcal{F}}[\mathcal{F}_{z_n}(A^{\mathcal{O}_{\mathcal{F}}}(z_n, y; s)) = y | \neg B_y] \leq \frac{n \cdot q(n)}{2^n - q(n) + 1} + \frac{n}{2^n - q(n)} \leq \frac{\mathsf{poly}(n)}{2^n}.$$

Therefore, we have that

$$\begin{aligned}
&\Pr_{\mathcal{F}, U_n, A}[A^{\mathcal{O}_{\mathcal{F}}}(z_n, \mathcal{F}_{z_n}(U_n)) \in \mathcal{F}_{z_n}^{-1}(\mathcal{F}_{z_n}(U_n))] \\
&\quad \leq \Pr_{\mathcal{F}, y \leftarrow \mathcal{F}_{z_n}(U_n)}[B_y] + \Pr_{\mathcal{F}, A, y \leftarrow \mathcal{F}_{z_n}(U_n)}[A^{\mathcal{O}_{\mathcal{F}}}(z_n, y) \in \mathcal{F}_{z_n}^{-1}(y) | \neg B_y] \leq \frac{C}{2^n} + \frac{\mathsf{poly}(n)}{2^n} \leq \frac{\mathsf{poly}(n)}{2^n}.
\end{aligned}$$

For any sufficiently large $n \in \mathbb{N}$, the above probability is less than $n^{-2} \cdot 2^{-n/2}$. By Lemma 1,

$$\Pr_{\mathcal{F}}\left[\Pr_{U_n, A}[A^{\mathcal{O}_{\mathcal{F}}}(z_n, \mathcal{F}_{z_n}(U_n)) \in \mathcal{F}_{z_n}^{-1}(\mathcal{F}_{z_n}(U_n))] \geq 2^{-\frac{n}{2}}\right] \leq \frac{1}{n^2},$$

for any sufficiently large n.

For each n, let E_n be the above event that $\Pr_{U_n, A}[A^{\mathcal{O}_{\mathcal{F}}}(z_n, \mathcal{F}_{z_n}(U_n)) \in \mathcal{F}_{z_n}^{-1}(\mathcal{F}_{z_n}(U_n))] \geq 2^{-\frac{n}{2}}$. Since the above inequality implies $\sum_{n\in\mathbb{N}} \Pr_{\mathcal{F}}[E_n] < \infty$, by the Borel-Cantelli lemma, the events E_n occur for infinitely many n with probability 0 over the choice of $\mathcal{F}$.

Recall that the above argument holds for any polynomial-time randomized oracle machine A. For each A, we ignore the measure zero of oracles where the the events E_n occur for infinitely many n. Since polynomial-time oracle machines are countable, we have ignored measure zero of oracles in total. Thus, the remaining measure one of oracles satisfies that for all polynomial-time oracle machines A, the events E_n occur only finitely often. In other words, with probability 1 over the choice of $\mathcal{F}$, all randomized polynomial-time oracle machines $A^?$ satisfy that for any sufficiently large $n \in \mathbb{N}$,

$$\Pr_{A, U_n}[A^{\mathcal{O}_{\mathcal{F}}}(z_n, \mathcal{F}_{z_n}(U_n)) \in \mathcal{F}_{z_n}^{-1}(\mathcal{F}_{z_n}(U_n))] < 2^{-\frac{n}{2}}.$$

This directly implies Lemma 7.

Therefore, the remaining part is to show that $\Pr_{\mathcal{F}}[B_y] \leq C \cdot 2^{-n}$. This bound holds by the following simple calculation (the reader may skip this part because it is not so essential):

$$\begin{aligned}\Pr_{\mathcal{F}}[B_y] &= \sum_{i=n+1}^{2^n} \binom{2^n}{i} (2^{-n})^i (1-2^{-n})^{2^n-i} \\ &\leq 2^n \cdot \binom{2^n}{n+1} (2^{-n})^{n+1} (1-2^{-n})^{2^n-n-1} \\ &\leq 2^n \cdot \frac{2^n \cdot (2^n-1) \cdot \cdots \cdot (2^n-n)}{(n+1)!} (2^{-n})^{n+1} \leq \frac{2^n}{(n+1)!} \leq \frac{4}{2^n}.\end{aligned}$$

□

To prove Lemma 8, we first show the following key lemma.

Lemma 9. *With probability* 1 *over the choice of an auxiliary-input embedded random function* $\mathcal{F}$*, all (possibly inefficient) oracle machines* $A^?$ *and* $c \in \mathbb{N}$ *satisfy that for any sufficiently large* $n \in \mathbb{N}$*,*

$$\Pr_{U_n}\left[A^{\mathcal{F}}(U_n) \text{ accesses } \mathcal{F}_{z_m} \text{ for } m \geq m_c \text{ within } c \cdot n^c \text{ queries to } \mathcal{F}_z \text{ with } |z| \geq m_c\right] < \frac{1}{n},$$

where $m_c = \lceil \log c \cdot n^{3+c} \rceil$ *and* $z_m \in \{0,1\}^m$ *is a target auxiliary-input of* $\mathcal{F}$ *for each* $m \in \mathbb{N}$*.*

Proof. Fix an oracle machine A and $c \in \mathbb{N}$ arbitrarily. Since any value of $\mathcal{F}_z$ with $|z| < m_c$ has no information about the target auxiliary-inputs z_m where $m \geq m_c$, we ignore the query access to $\mathcal{F}_z$ with $|z| < m_c$ by A in the following argument.

For any $m \in \mathbb{N}$ and $z \in \{0,1\}^m$, the target auxiliary-input z_m corresponds to z with probability exactly 2^{-m} over the choice of $\mathcal{F}$. For any $x \in \{0,1\}^n$, under the condition that A does not access to the target auxiliary-input, the queries generated by $A^{\mathcal{F}}(x)$ are determined independent of the choice of $\mathcal{F}$. We call such queries "typical" queries. Then we have that for any $x \in \{0,1\}^n$,

$$\begin{aligned}&\Pr_{\mathcal{F}}\left[A^{\mathcal{F}}(x) \text{ accesses } \mathcal{F}_{z_m} \text{ for } m \geq m_c \text{ within } cn^c \text{ queries}\right] \\ &\quad = \Pr_{\mathcal{F}}\left[\exists z_m \text{ where } m \geq m_c \text{ such that } z_m \text{ corresponds to one of the first } cn^c \text{ typical queries}\right] \\ &\quad \leq \frac{cn^c}{2^{m_c}} = \frac{cn^c}{cn^{3+c}} = \frac{1}{n^3}.\end{aligned}$$

Therefore,

$$\Pr_{\mathcal{F},U_n}\left[A^{\mathcal{F}}(U_n) \text{ accesses } \mathcal{F}_{z_m} \text{ for } m \geq m_c \text{ within } cn^c \text{ queries}\right] \leq \frac{1}{n^3},$$

and by Lemma 1,

$$\Pr_{\mathcal{F}}\left[\Pr_{U_n}\left[A^{\mathcal{F}}(U_n) \text{ accesses } \mathcal{F}_{z_m} \text{ for } m \geq m_c \text{ within } cn^c \text{ queries}\right] \geq \frac{1}{n}\right] \leq \frac{1}{n^2}.$$

For each n, let E_n be the above event that

$$\Pr_{U_n}\left[A^{\mathcal{F}}(U_n) \text{ accesses } \mathcal{F}_{z_m} \text{ for } m \geq m_c \text{ within } cn^c \text{ queries}\right] \geq \frac{1}{n}.$$

Since the above inequality implies $\sum_{n\in\mathbb{N}} \Pr_{\mathcal{F}}[E_n] < \infty$, by the Borel-Cantelli lemma, the events E_n occur for infinitely many n with probability 0 over the choice of $\mathcal{F}$.

Note that the above argument holds for any tuple (A, n) of an oracle machine and an integer. Now we ignore the measure 0 of oracles which satisfies the above condition for any (A, n). Since such tuples (A, n) are countable, by the same argument in the proof of Lemma 7, we have that with probability 1 over the choice of $\mathcal{F}$, for all oracle machines $A^?$ and $c \in \mathbb{N}$, the events E_n do not occur for sufficiently large n. This is equivalent to the statement in Lemma 9. □

Finally, we give the proof of Lemma 8.

Proof of Lemma 8. Fix a polynomial-time oracle machine $F^?$ arbitrarily, and assume that F asks at most $c \cdot n^c$ queries on any input of length n. Let $m_c := \lceil \log c \cdot n^{3+c} \rceil$. We can also assume that the output length of $F^?(x)$ is exactly $p(|x|)$ by zero-padding, where p is a polynomial. Note that zero-padding does not change the one-wayness and the query complexity of $F^?$.

Let $r(n) = \sum_{i=1}^{m_c-1} i \cdot 2^{2i}$ $(\leq 2^{O(m_c)} = \mathsf{poly}(n))$. Now we define an auxiliary-input function $f : \{0,1\}^{r(n)} \times \{0,1\}^n \to \{0,1\}^{p(n)}$ (where we regard the first input as an auxiliary-input, that is, $f_z(x) = f(z, x)$) by the deterministic procedure in Algorithm 5.

Algorithm 5: Procedure for computing f

Input : $z \in \{0,1\}^{r(n)}$ and $x \in \{0,1\}^n$

executes $F^?(x)$ where the answer for each query $y \in \{0,1\}^*$ is simulated as follows:
  if $|y| = 1$ **then** answer 0;
  else if $y = 1 \circ x'$ **then** answer $\mathsf{TQBF}(x')$;
  else if $y = 0 \circ \langle z', x' \rangle$ *and* $|z'| = |x'|$ **then**
    if $|z'| \geq m_c$ **then** answer 0;
    else
      let $m = |z'|$ and $k = \left(\sum_{i=1}^{m-1} i 2^{2i}\right) + m(2^m(z'_{\mathbb{N}} - 1) + x'_{\mathbb{N}} - 1)$;
      answer $z_{k+1} \circ \ldots \circ z_{k+m}$;
  else answer 0 (in this else case, the query y is invalid for $\mathcal{O}_{\mathcal{F}}$);
return the same value to $F^?(x)$ in the above simulation;

It is easily checked that f is polynomial-time computable with access to TQBF. Thus, f is also computable using only polynomial-size space (but not in polynomial-time) without any oracle access. Therefore, following problem is contained in PSPACE:

Input: $z \in \{0,1\}^{r(n)}, y \in \{0,1\}^n, s, t \in \{0,1\}^n$.
Goal: determine whether there exists $x \in \{0,1\}^n$ such that $f_z(x) = y$ and $s_{\mathbb{N}} \leq x_{\mathbb{N}} \leq t_{\mathbb{N}}$.

Therefore, by applying the binary search, there exists a polynomial-time oracle machine $I^?$ such that $I^{\mathsf{TQBF}}(z, y)$ outputs lexicographically first inverse element of $f_z(y)$ if any, otherwise $0^{|y|}$.

Now we construct the inverter A for $F^?$ as Algorithm 6. For simplicity, we identify access to $\mathcal{O}_{\mathcal{F}}$ with access to $\mathcal{F}$ and TQBF and may use both of notations interchangeably.

For each choice of $\mathcal{F}$, we use the notation $z_{\mathcal{F}}$ to denote the binary string constructed in lines 1–5 of A. Notice that $|z_{\mathcal{F}}| = r(n)$ and $z_{\mathcal{F}}$ consists of truth tables of $\mathcal{F}_z$ for each $z \in \{0,1\}^{\leq m_c-1}$.

Now we show that with probability 1 over the choice of $\mathcal{F}$,

$$\Pr_{U_n}\left[A^{\mathcal{O}_{\mathcal{F}}}(F^{\mathcal{O}_{\mathcal{F}}}(U_n)) \notin F^{\mathcal{O}_{\mathcal{F}}{}^{-1}}(F^{\mathcal{O}_{\mathcal{F}}}(U_n))\right] \leq \frac{2}{n}.$$

Algorithm 6: A (an inverting algorithm for $F^?$)

Input : $y \in \{0,1\}^n$
Oracle : TQBF, an auxiliary-input embedded random function $\mathcal{F}$ (equivalently, $\mathcal{O}_{\mathcal{F}}$)

set $z_{\mathcal{F}}$ to empty string;
for $i = 1$ **to** $m_c - 1$ **do**
  for $z = 0^i$ **to** $1^i \in \{0,1\}^i$ **do**
    for $x = 0^i$ **to** $1^i \in \{0,1\}^i$ **do**
      $z_{\mathcal{F}} := z_{\mathcal{F}} \circ \mathcal{F}_z(x)$;
execute $x \leftarrow I^{\mathsf{TQBF}}(z_{\mathcal{F}}, y)$ and **return** x;

For any choice of $\mathcal{F}$, $n \in \mathbb{N}$, and $x \in \{0,1\}^n$, the property of I implies that

$$f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x))) = f_{z_{\mathcal{F}}}(x), \tag{7}$$

and by the definition of f, we have that

$$F^{\mathcal{O}_{\mathcal{F}}}(x) \text{ does not access to any } \mathcal{F}_{z_m} \text{ with } m \geq m_c \Longrightarrow f_{z_{\mathcal{F}}}(x) = F^{\mathcal{O}_{\mathcal{F}}}(x). \tag{8}$$

For any $y \in \{0,1\}^n$, $x_{\mathcal{F},y} := A^{\mathcal{O}_{\mathcal{F}}}(y)$ is uniquely determined and in $\{0,1\}^n$, thus we also have that

$$F^{\mathcal{O}_{\mathcal{F}}}(x_{\mathcal{F},y}) \text{ does not access to any } \mathcal{F}_{z_m} \text{ with } m \geq m_c \Longrightarrow f_{z_{\mathcal{F}}}(x_{\mathcal{F},y}) = F^{\mathcal{O}_{\mathcal{F}}}(x_{\mathcal{F},y}). \tag{9}$$

If $\mathcal{F}$ and $x \in \{0,1\}^n$ satisfy the following three conditions

a. $f_{z_{\mathcal{F}}}(x) = F^{\mathcal{O}_{\mathcal{F}}}(x)$;

b. $F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x))) = f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x)))$;

c. $f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x))) = f_{z_{\mathcal{F}}}(x)$,

then it is easily checked that $F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(F^{\mathcal{O}_{\mathcal{F}}}(x))) = F^{\mathcal{O}_{\mathcal{F}}}(x)$. Hence, by union bound,

$$\begin{aligned}\Pr_{U_n}\left[A^{\mathcal{O}_{\mathcal{F}}}(F^{\mathcal{O}_{\mathcal{F}}}(U_n)) \notin {F^{\mathcal{O}_{\mathcal{F}}}}^{-1}(F^{\mathcal{O}_{\mathcal{F}}}(U_n))\right] \\ \leq \Pr_{U_n}\left[f_{z_{\mathcal{F}}}(U_n) \neq F^{\mathcal{O}_{\mathcal{F}}}(U_n)\right] + \Pr_{U_n}\left[F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n))) \neq f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n)))\right] \\ + \Pr_{U_n}\left[f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n))) \neq f_{z_{\mathcal{F}}}(U_n)\right].\end{aligned}$$

Thus, we bound above the three probabilities in the right-hand side.

For the third probability, by the equation (7), for any choice of $\mathcal{F}$,

$$\Pr_{U_n}\left[f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n))) \neq f_{z_{\mathcal{F}}}(U_n)\right] = 0.$$

For the first and second probability, we will apply Lemma 9. By the conditions (8) and (9),

$$\Pr_{U_n}\left[f_{z_{\mathcal{F}}}(U_n) \neq F^{\mathcal{O}_{\mathcal{F}}}(U_n)\right] \leq \Pr_{U_n}\left[F^{\mathcal{O}_{\mathcal{F}}}(U_n) \text{ accesses to some } \mathcal{F}_{z_m} \text{ with } m \geq m_c\right],$$

and

$$\Pr_{U_n}\left[F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n))) \neq f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n)))\right] \leq \Pr_{U_n}\left[F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(U_n))) \text{ accesses to any } \mathcal{F}_{z_m} \text{ with } m \geq m_c\right],$$

where we use the fact that A does not access to $\mathcal{F}_z$ with $|z| \geq m_c$ for the second inequality.

Notice that both executions of $F^{\mathcal{O}_{\mathcal{F}}}(x)$ and $F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x)))$ are implemented by exponential-time oracle Turing machines given access to $\mathcal{F}$ which make at most $c \cdot n^c$ queries to $\mathcal{F}_z$ with $|z| > m_c$. Therefore, by Lemma 9, with probability 1 over the choice of $\mathcal{F}$, both of probabilities is bounded above by $1/n$, and

$$\Pr_{U_n}\left[A^{\mathcal{O}_{\mathcal{F}}}(F^{\mathcal{O}_{\mathcal{F}}}(U_n)) \notin F^{\mathcal{O}_{\mathcal{F}}{}^{-1}}(F^{\mathcal{O}_{\mathcal{F}}}(U_n))\right] \leq \frac{1}{n} + \frac{1}{n} + 0 = \frac{2}{n}. \tag{10}$$

The above argument holds for any polynomial-time oracle machine $F^?$. Since polynomial-time oracle machines are countable, we have Lemma 8 by ignoring the measure 0 of oracles not satisfying the condition (10) for each $F^?$ and applying the same argument in the proof of Lemma 7. □

Acknowledgments

This work was supported by JST, ACT-X Grant Number JPMJAX190M, Japan.

References

W. Aiello and J. Håstad. Perfect zero-knowledge languages can be recognized in two rounds. In *28th Annual Symposium on Foundations of Computer Science*, pages 439–448, 1987.

A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On Basing One-Way Functions on NP-Hardness. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, STOC 06, pages 701–710, New York, NY, USA, 2006. ACM.

E. Allender and S. Hirahara. New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems. *TOCT*, 11(4):27:1–27:27, 2019.

A. Andreev, A. Clementi, and J. Rolim. A New General Derandomization Method. *J. ACM*, 45(1): 179–213, January 1998.

B. Applebaum, B. Barak, and D. Xiao. On Basing Lower-Bounds for Learning on Worst-Case Assumptions. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '08, pages 211–220, 2008.

T. Baker, J. Gill, and R. Solovay. Relativizations of the $\mathcal{P} =? \mathcal{NP}$ Question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

A. Bogdanov and C. Brzuska. On Basing Size-Verifiable One-Way Functions on NP-Hardness. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 1–6, 2015.

A. Bogdanov and C. Lee. Limits of Provable Security for Homomorphic Encryption. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 111–128, 2013.

A. Bogdanov and L. Trevisan. On Worst-Case to Average-Case Reductions for NP Problems. *SIAM J. Comput.*, 36(4):1119–1159, December 2006a.

A. Bogdanov and L. Trevisan. Average-Case Complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1):1–106, 2006b.

J. Feigenbaum and L. Fortnow. On the random-self-reducibility of complete sets. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 124–132, 1991.

O. Goldreich. *Foundations of Cryptography: Volume 1.* Cambridge University Press, New York, NY, USA, 2006. ISBN 0521035368.

D. Gutfreund and S. Vadhan. Limitations of Hardness vs. Randomness under Uniform Reductions. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. APPROX 2008, RANDOM 2008*, volume 5171 of *LNCS*, pages 469–482, 2008.

I. Haitner, M. Mahmoody, and D. Xiao. A New Sampling Protocol and Applications to Basing Cryptographic Primitives on the Hardness of NP. In *IEEE 25th Annual Conference on Computational Complexity*, pages 76–87, 2010.

I. Haitner, O. Reingold, and S. Vadhan. Efficiency Improvements in Constructing Pseudorandom Generators from One-Way Functions. *SIAM Journal on Computing*, 42(3):1405–1430, 2013.

J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from Any One-way Function. *SIAM J. Comput.*, 28(4):1364–1396, March 1999.

S. Hirahara. Non-Black-Box Worst-Case to Average-Case Reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 247–258, 2018.

S. Hirahara and O. Watanabe. On Nonadaptive Reductions to the Set of Random Strings and Its Dense Subsets. In *Complexity and Approximation - In Memory of Ker-I Ko*, pages 67–79, 2020.

W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

T. Holenstein. Key Agreement from Weak Bit Agreement. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC 05, pages 664–673, New York, NY, USA, 2005. ACM.

T. Holenstein. Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness. In *Theory of Cryptography*, pages 443–461, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147, 1995.

R. Impagliazzo. Relativized Separations of Worst-Case and Average-Case Complexities for NP. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 104–114, 2011.

R. Impagliazzo and L. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, volume 2, pages 812–821, 1990.

R. Impagliazzo and M. Luby. One-way Functions Are Essential for Complexity Based Cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 230–235, 1989.

R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC 89, pages 44–61, New York, NY, USA, 1989. ACM.

R. Impagliazzo and A. Wigderson. Randomness vs Time: Derandomization under a Uniform Assumption. *Journal of Computer and System Sciences*, 63(4):672 – 688, 2001.

M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

T. Liu and V. Vaikuntanathan. On Basing Private Information Retrieval on NP-Hardness. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 372–386, 2016.

M. Nanashima. Extending Learnability to Auxiliary-Input Cryptographic Primitives and Meta-PAC Learning. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT'20)*, volume 125. PMLR, 09–12 Jul 2020.

R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *The 2nd Israel Symposium on Theory and Computing Systems*, pages 3–17, June 1993.

O. Reingold, L. Trevisan, and S. Vadhan. Notions of Reducibility between Cryptographic Primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 1–20, 2004.

J. Rompel. One-way Functions Are Necessary and Sufficient for Secure Signatures. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, pages 387–394, 1990.

S. Rudich. *Limits on the Provable Consequences of One-way Functions*. PhD thesis, EECS Department, University of California, Berkeley, Dec 1988.

D. Xiao. On basing ZK $\neq$ BPP on the hardness of PAC learning. In *In Proc. CCC '09*, pages 304–315, 2009a.

D. Xiao. *New Perspectives on the Complexity of Computational Learning, and Other Problems in Theoretical Computer Science*. PhD thesis, Princeton University, 2009b.

A Universal Extrapolation

A.1 Basic Tools

We introduce additional basic tools to show Lemma 2.

Fact 3 (Inclusion-exclusion principle)**.** *For every events $E_1, \ldots, E_n$,*

$$\Pr\left[\bigcup_{i=1}^{n} E_i\right] \geq \sum_{i=1}^{n} \Pr[E_i] - \frac{1}{2}\sum_{i \neq j} \Pr[E_i \cap E_j].$$

Fact 4 (Universal hash function)**.** *For $n \in \mathbb{N}$ and $a, b \in \mathbb{F}_{2^n}$, define a function $h_{a,b} : \{0,1\}^{\leq n} \to \{0,1\}^n$ by $h_{a,b}(x) = a \cdot x + b$ where the input x is interpreted as $x \circ 0^{n-|x|} \in \mathbb{F}_{2^n}$. Then we have that for any $k \in [n]$, $x_1, x_2 \in \{0,1\}^k$ with $x_1 \neq x_2$, and $y_1, y_2 \in \{0,1\}^n$,*

$$\Pr_{a,b}[h_{a,b}(x_1) = y_1] = 2^{-n} \text{ and } \Pr_{a,b}[h_{a,b}(x_1) = y_1 \wedge h_{a,b}(x_2) = y_2] = 2^{-2n}.$$

A.2 Proof of Lemma 2

For $a, b \in \{0,1\}^n$, let $h_{a,b} : \{0,1\}^n \to \{0,1\}^n$ be a universal hash function as in Fact 4.

We define an auxiliary-input function $f : \{0,1\}^{e(s(n))} \times \{0,1\}^{3n+\lceil \log n \rceil} \to \{0,1\}^{2n+s(n)+\lceil \log n \rceil}$ by

$$f_z(a \circ b \circ r \circ i) := f(z, a \circ b \circ r \circ i) = \begin{cases} a \circ b \circ y \circ 0^{s(n)-|y|} \circ i & (\text{if } i_{\mathbb{N}} \leq n) \\ 0^{2n+s(n)+\lceil \log n \rceil} & (\text{otherwise}), \end{cases}$$

where $a, b, r \in \{0,1\}^n$, $i \in \{0,1\}^{\lceil \log n \rceil}$, C_z is an interpretation of z as a circuit, and $y = C_z(h_{a,b}(r_{[i_{\mathbb{N}}]}))$.

Note that the above f_z is defined only in the case where $|z| = \{0,1\}^{e(s(n))}$ for some $n \in \mathbb{N}$. Even in the general case where $z \in \{0,1\}^a$, by truncating z to the length $e(s(n))$ for maximum $n \in \mathbb{N}$ satisfying $e(s(n)) \leq a$ and compute f_z as above, we have a general auxiliary-input function f.

For simplicity, we consider that f_z takes four inputs as $f_z(a, b, r, i)$ in the above definition and n is the security parameter instead of $3n + \lceil \log n \rceil$. Obviously, f_z is polynomial-time computable.

Let $\delta'(n) = \frac{1}{2n} \cdot (1 - 2^{-\frac{\epsilon}{4}})^2 \cdot \frac{\delta(n)}{12}$. We construct the universal extrapolation algorithm $Ext_{s(n)}$ which accesses (possibly randomized) $(1-\delta')$-inverting oracle for f_z nonadaptively as Algorithm 7. This immediately implies that if there exists no auxiliary-input function, then $Ext_{s(n)}$ is implemented by a randomized polynomial-time algorithm.

Algorithm 7: $Ext_{s(n)}$ (a universal extrapolation algorithm for $s(n)$-size circuits)

Input : an n-input circuit $C \in \{0,1\}^{e(s(n))}$, $y \in \{0,1\}^{\leq s(n)}$
Oracle : $\mathcal{I}$ (which $(1-\delta')$-inverts $\{f_z\}_{z \in \{0,1\}^*}$)

prepare queries
  zero-pad $y := y \circ 0^{s(n)-|y|}$, and let $\epsilon' := \frac{1}{4}(2^{-\epsilon/2} - 2^{-\epsilon})$ and $M := \lceil \frac{\ln 3n - \ln \delta(n)}{2\epsilon'^2} \rceil$;
  for $i := 1$ **to** n **do**
    for $j := 1$ **to** M **do**
      select $a_{i,j}, b_{i,j} \leftarrow_u \{0,1\}^n$;
query $(a'_{i,j}, b'_{i,j}, r_{i,j}, k_{i,j}) \leftarrow \mathcal{I}(C, a_{i,j} \circ b_{i,j} \circ y \circ i)$ for each $(i,j) \in [n] \times [M]$;
let $c[1], \ldots, c[n] := 0$;
foreach $(i,j) \in [n] \times [M]$ **do**
  if $(a'_{i,j}, b'_{i,j}, k_{i,j}) = (a_{i,j}, b_{i,j}, i)$ *and* $y = C(h_{a_{i,j},b_{i,j}}(r_{i,j[i]}))$ **then** $c[i] := c[i] + 1$;
find $\min \tilde{i} \in [n]$ satisfying that $c[\tilde{i}] \geq \frac{M}{4}(2^{-\epsilon/2} + 2^{-\epsilon})$;
return $2^{-\tilde{i}}$;

Assume that the given $\mathcal{I}$ is $(1-\delta')$-inverts f, that is, for any $n \in \mathbb{N}$ and $\langle C \rangle \in \{0,1\}^{e(s(n))}$,

$$\Pr_{a,b,r,i,\mathcal{I}}[\mathcal{I}(C, f_C(a,b,r,i)) \notin f_C^{-1}(f_C(a,b,r,i))] \leq \frac{1}{2n} \cdot (1 - 2^{-\frac{\epsilon}{4}})^2 \cdot \frac{\delta(n)}{12}. \tag{11}$$

For any $i \in [n]$, we have that

$$\Pr_{a,b,r,\mathcal{I}}[\mathcal{I}(C, f_C(a,b,r,i)) \notin f_C^{-1}(f_C(a,b,r,i))] \leq (1-2^{-\frac{\epsilon}{4}})^2 \cdot \frac{\delta(n)}{12}, \tag{12}$$

otherwise the equation (11) does not hold because each $i \in [n]$ is selected with probability $2^{-\lceil \log n \rceil} \geq 1/2n$.

Fix input $n \in \mathbb{N}$, $\langle C \rangle \in \{0,1\}^{e(s(n))}$, and $y \in C(\{0,1\}^n)$ arbitrarily. Let $y := y \circ 0^{s(n)-|y|}$. We introduce some notations as follows:

$$\begin{aligned} p_y &= \Pr_{U_n}[C(U_n) = y], \\ X_y &= \{x \in \{0,1\}^n : C(x) = y\}, \\ t^*_y &= -\log p_y \quad \text{(information of } y\text{)}, \\ t_y &= \lceil t^*_y \rceil. \end{aligned}$$

Note that $2^{n-t^*_y} = p_y \cdot 2^n = |X_y|$. For simplicity, we may omit to write the index y in the above notations when we consider arbitrary $y \in C(\{0,1\}^n)$.

First we assume that the following claims on the execution of $Ext_{s(n)}$:

Claim 9. *For any $i < t^* - (1+\epsilon)$,*

$$\Pr\left[c[i] \geq \frac{M}{4}(2^{-\epsilon/2} + 2^{-\epsilon})\right] \leq \frac{\delta(n)}{3n},$$

where the probability depends on the choice of y,$a_{i,j}$,$b_{i,j}$, and the executions of $\mathcal{I}$.

Claim 10. *For any n-input circuit C of size at most $s(n)$, there exists a good set G_C such that*

$$\Pr_{U_n}[C(U_n) \notin G_C] \leq \frac{\delta(n)}{3}$$

and

$$\Pr\left[c[t] \leq \frac{M}{4}(2^{-\epsilon/2} + 2^{-\epsilon}) \middle| y \in G_C\right] \leq \frac{\delta(n)}{3},$$

where the latter probability depends on the choice of y,$a_{i,j}$,$b_{i,j}$, and the executions of $\mathcal{I}$.

Then we can show the correctness of $Ext_{s(n)}$ as follows: assume that the input y is in G_C (which occurs with probability at least $1 - \delta(n)/3$). By Claim 9 and union bound, $\tilde{i} \geq t^* - (1+\epsilon)$ with probability at least $1 - n \cdot \delta(n)/3n = 1 - \delta(n)/3$. By Claim 10, $\tilde{i} \leq t^* + 1$ with probability at least $1 - \delta(n)/3$. Thus, with probability at least $1 - \delta(n)$, $t^* - (1+\epsilon) \leq \tilde{i} \leq t^* + 1$ holds. This implies Lemma 2 because $Ext_{s(n)}$ outputs $2^{-\tilde{i}}$ and

$$\frac{1}{2} \cdot p \leq 2^{-t^*-1} \leq 2^{-\tilde{i}} \leq 2^{-t^*+(1+\epsilon)} \leq 2^{(1+\epsilon)} \cdot p.$$

Therefore, the remaining part is to show Claims 9 and 10.

Proof of Claim 9. Assume that $i < t^* - (1+\epsilon)$. Fix $j \in [M]$ arbitrarily. For simplicity, we write $(a_{i,j}, b_{i,j})$ as (a,b).

On the choice of (a,b) (in line 5), if there exists no $r \in \{0,1\}^i$ satisfying $C(h_{a,b}(r)) = y$ (equivalently, $h_{a,b} \in X$), then there exists no inverse element of $(C, a_{i,j} \circ b_{i,j} \circ y \circ i)$ and $\mathcal{I}$ must fail in inverting f_C. In this case, $c[i]$ does not increase.

Therefore, it is enough to show that

$$\Pr_{a,b}[\exists\, r \in \{0,1\}^i \text{ such that } h_{a,b}(r) \in X] \le \frac{1}{2} \cdot \frac{1}{2^\epsilon}. \tag{13}$$

This is because if we define Bernoulli random variables

$$E_j := \mathbb{1}\{c[i] \text{ is incremented in line 9 for } (i,j)\},$$

then for each $j \in [M]$,

$$\mu := \mathrm{E}[E_j] = \Pr_{a,b,\mathcal{I}}[E_j] \le \Pr_{a,b}[\exists\, r \in \{0,1\}^i \text{ such that } h_{a,b}(r) \in X] \le \frac{1}{2} \cdot \frac{1}{2^\epsilon},$$

and by the Hoeffding inequality,

$$\begin{aligned}
\Pr\left[c[i] \ge \frac{M}{4}(2^{-\epsilon/2} + 2^{-\epsilon})\right] &= \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j \ge \frac{1}{2} \cdot \frac{1}{2^\epsilon} + \epsilon'\right] \\
&\le \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j - \mu \ge \epsilon'\right] \\
&\le \exp\left(-2M\epsilon'^2\right) \\
&\le \exp\left(-2\frac{\ln 3n - \ln \delta(n)}{2\epsilon'^2}\epsilon'^2\right) = \frac{\delta(n)}{3n}.
\end{aligned}$$

For inequality (13), by union bound, we have that

$$\begin{aligned}
\Pr_{a,b}[\exists\, r \in \{0,1\}^i \text{ such that } h_{a,b}(r) \in X] &= \Pr_{a,b}[\exists\, (r,x) \in \{0,1\}^i \times X \text{ such that } h_{a,b}(r) = x] \\
&\le \sum_{(r,x)} \Pr_{a,b}[h_{a,b}(r) = x] \\
&= 2^i \cdot |X| \cdot 2^{-n} \\
&\le 2^{t^* - (1+\epsilon)} \cdot 2^{n - t^*} \cdot 2^{-n} = 2^{-(1+\epsilon)}
\end{aligned}$$

□

Proof of Claim 10. Assume $i = t$. We use the same notation E_j and μ as in the proof of Claim 9. Then it is enough to show that there exists $B_C \subseteq C(\{0,1\}^n)$ such that

$$\Pr_{U_n}[C(U_n) \in B_C] \le \frac{\delta(n)}{3},$$

and under the condition that $y \in C(\{0,1\}^n) \setminus B_C$,

$$\Pr[E_j] \ge \frac{1}{2} \cdot \frac{1}{2^{\epsilon/2}}, \tag{14}$$

for each $j \in [M]$. If the above holds, then under the condition that $y \in C(\{0,1\}^n) \setminus B_C$,

$$\begin{aligned}
\Pr\left[c[i] \leq \frac{M}{4}(2^{-\epsilon/2} + 2^{-\epsilon})\right] &= \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j \geq \frac{1}{2} \cdot \frac{1}{2^{\epsilon/2}} - \epsilon'\right] \\
&\leq \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j - \mu \leq -\epsilon'\right] \\
&\leq \exp\left(-2M\epsilon'^2\right) \quad (\because \text{ the Hoeffding inequality}) \\
&\leq \exp\left(-2\frac{\ln 3n - \ln \delta(n)}{2\epsilon'^2}\epsilon'^2\right) = \frac{\delta(n)}{3n} \leq \frac{\delta(n)}{3}.
\end{aligned}$$

Therefore, by letting $G_C := C(\{0,1\}^n) \setminus B_C$, we have Claim 10.

For inequality (14), first we show the following inequality:

$$\Pr_{a,b}[\exists r \in \{0,1\}^t \text{ such that } h_{a,b}(r) \in X] \geq 1/2. \tag{15}$$

Let $R = \{r \circ 0^{n-t} | r \in \{0,1\}^r\} \subseteq \{0,1\}^n$. For $r \in R$, define an event E_r over the choice of (a,b) by $E_r := (h_{a,b}(r) \in X \text{ holds})$. Then we have that

$$\Pr_{a,b}[\exists r \in \{0,1\}^t \text{ such that } h_{a,b}(r) \in X] = \Pr\left[\bigcup_{r \in R} E_r\right].$$

For any $n \in \mathbb{N}$ and $\epsilon \in [0,1)$, we define a value $(n+\epsilon)_r$ by

$$(n+\epsilon)_r = \begin{cases} n & \epsilon < 1/2 \\ n+1 & \epsilon \geq 1/2. \end{cases}$$

Let R' be a set of the first $(2^{t^*})_r$ elements of $\{0,1\}^n$ as $R' := \{0^n, 10^{n-1}, 010^{n-2}, 110^{n-2}, 0010^{n-3}, \ldots\}$. Since $(2^{t^*})_r \leq \lceil 2^{t^*} \rceil \leq 2^{\lceil t^* \rceil} = 2^t$, $R' \subseteq R$.

By the inclusion-exclusion principle (Fact 3),

$$\Pr\left[\bigcup_{r \in R} E_r\right] \geq \Pr\left[\bigcup_{r \in R'} E_r\right] \geq \sum_{r \in R'} \Pr[E_r] - \frac{1}{2} \sum_{\substack{r \neq r' \\ r,r' \in R'}} \Pr[E_r \cap E_{r'}].$$

Thus, we evaluate each term in the right-hand side.

For each $r \in R'$,

$$\begin{aligned}
\Pr[E_r] = \Pr_{a,b}[h_{a,b}(r) \in X] &= \sum_{x \in X} \Pr_{a,b}[h_{a,b}(r) = x] \\
&= |X| \cdot 2^{-n} = 2^{n-t^*} \cdot 2^{-n} = 2^{-t^*}.
\end{aligned}$$

For each $r, r' \in R'$ with $r \neq r'$,

$$\begin{aligned}
\Pr[E_r \cap E_{r'}] &= \Pr_{a,b}[\exists x, x' \in X \text{ such that } h_{a,b}(r) = x \wedge h_{a,b}(r') = x'] \\
&\leq \sum_{x,x' \in X} \Pr_{a,b}[h_{a,b}(r) = x \wedge h_{a,b}(r') = x'] \\
&= |X|^2 \cdot 2^{-2n} = 2^{2(n-t^*)} \cdot 2^{-2n} = 2^{-2t^*}.
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}\Pr\left[\bigcup_{r\in R} E_r\right] &\geq \sum_{r\in R'} \Pr[E_r] - \frac{1}{2}\sum_{\substack{r\neq r'\\ r,r'\in R'}} \Pr[E_r \cap E_{r'}]\\ &\geq |R'|\cdot 2^{-t^*} - \frac{1}{2}\cdot |R'|(|R'|-1)\cdot 2^{-2t^*}\\ &= (2^{t^*})_r \cdot 2^{-t^*} - \frac{1}{2}\cdot (2^{t^*})_r((2^{t^*})_r - 1)\cdot 2^{-2t^*} \geq 1/2,\end{aligned}$$

where the last inequality holds by the following inequality: for any $x \in \mathbb{R}$ with $x \geq 1$,

$$(x)_r \cdot x^{-1} - \frac{1}{2}\cdot (x)_r((x)_r - 1)\cdot x^{-2} \geq 1/2. \tag{16}$$

The above inequality (16) is shown by the following simple calculation: let $\epsilon := (x_r) - x$, then $\epsilon \in (-1/2, 1/2]$ and

$$\begin{aligned}(\text{LHS of (16)}) &= (x+\epsilon)\cdot x^{-1} - \frac{1}{2}\cdot (x+\epsilon)(x+\epsilon-1)\cdot x^{-2}\\ &= 1 + \frac{\epsilon}{x} - \frac{1}{2}\left(1 + \frac{2\epsilon-1}{x} + \frac{\epsilon(\epsilon-1)}{x^2}\right)\\ &= \frac{1}{2} + \frac{1}{2}\cdot\frac{x - \epsilon(\epsilon-1)}{x^2} \geq \frac{1}{2},\end{aligned}$$

where the last inequality holds because $\epsilon(\epsilon-1) < 1\ (\leq x)$ for any $\epsilon \in (-1/2, 1/2]$. Thus, the inequality (15) holds.

Now we introduce subsets of $\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^{\leq s(n)}$ as follows:

$$\begin{aligned}V &= \{(a,b,y) : \exists r \in \{0,1\}^{t_y} \text{ such that } C(h_{a,b}(r)) = y\},\\ V_y &= \{(a,b,y) \in V\} \quad \text{for any } y \in C(\{0,1\}^n),\\ F &= \left\{(a,b,y) : \Pr_{\mathcal{I}}\left[\mathcal{I}(C, a\circ b\circ y\circ 0^{s(n)-|y|}\circ t_y) \text{ fails in inverting}\right] \geq 1 - 2^{-\frac{\epsilon}{4}}\right\}.\end{aligned}$$

We also define the subset B_C of $C(\{0,1\}^n)$ by

$$B_C = \left\{y \in C(\{0,1\}^n) : \Pr_{a,b,r\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r))) \in F \cap V_y] \geq (1-2^{-\frac{\epsilon}{4}}) \Pr_{a,b,r\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r))) \in V_y]\right\}.$$

First, we bound the probability that $C(x) \in B_C$ over the choice of x above by $\delta(n)/3$.

By Lemma 1 and the upper bound (12) on the failure probability of $\mathcal{I}$,

$$\begin{aligned}&\Pr_{a,b,r}[(a,b,C(h_{a,b}(r))) \in F]\\ &\leq \frac{\Pr_{a,b,r,\mathcal{I}}\left[y = C(h_{a,b}(r)); \mathcal{I}(C, a\circ b\circ y\circ 0^{s(n)-|y|}\circ t_y) \text{ fails in inverting}\right]}{1-2^{-\frac{\epsilon}{4}}}\\ &\leq \frac{\Pr_{a,b,r,\mathcal{I}}\left[\mathcal{I}(C, f_C(a,b,r,t_{C(h_{a,b}(r))})) \notin f_C^{-1}(f_C(a,b,r,t_{C(h_{a,b}(r))}))\right]}{1-2^{-\frac{\epsilon}{4}}}\\ &\leq \frac{(1-2^{-\frac{\epsilon}{4}})^2\cdot\frac{\delta(n)}{12}}{1-2^{-\frac{\epsilon}{4}}} = (1-2^{-\frac{\epsilon}{4}})\cdot\frac{\delta(n)}{12} \qquad (\because \text{ (12)})\end{aligned}$$

By inequality (15), for any $y \in C(\{0,1\}^n)$,

$$\Pr_{a,b,r\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r))) \in V_y] \geq \frac{1}{2}\cdot\frac{1}{2^{t_y}}.$$

Since $t_y^* > t_y - 1$, for any $y \in C(\{0,1\}^n)$,

$$\Pr[C(U_n) = y] = 2^{-t_y^*} \leq 2\cdot 2^{-t_y} = 4\cdot \Pr_{a,b,r\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r))) \in V_y]$$

Thus, we have that

$$\begin{aligned}
\Pr_{U_n}[C(U_n)\in B_C] &= \sum_{y\in B_C}\Pr_{U_n}[C(U_n)=y]\\
&\leq \sum_{y\in B_C} 4\cdot \Pr_{a,b,r\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r)))\in V_y]\\
&\leq \sum_{y\in B_C} 4\cdot(1-2^{-\frac{\epsilon}{4}})^{-1}\cdot \Pr_{a,b,y\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r)))\in F\cap V_y] \quad (\because y\in B_C)\\
&= 4\cdot(1-2^{-\frac{\epsilon}{4}})^{-1}\cdot \Pr_{a,b,y\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r)))\in F\cap(\cup_{y\in B_C}V_y)]\\
&\leq 4\cdot(1-2^{-\frac{\epsilon}{4}})^{-1}\cdot \Pr_{a,b,y\sim\{0,1\}^n}[(a,b,C(h_{a,b}(r)))\in F]\\
&\leq 4\cdot(1-2^{-\frac{\epsilon}{4}})^{-1}\cdot(1-2^{-\frac{\epsilon}{4}})\cdot\frac{\delta(n)}{12} = \frac{\delta(n)}{3}.
\end{aligned}$$

Hence, the remaining part is to show inequality (14).

Fix $y \in C(\{0,1\}^n)\setminus B_C$ arbitrarily. Then we have that

$$\begin{aligned}
&\Pr_{a,b,r}[(a,b,C(h_{a,b}(r)))\in V_y\setminus F]\\
&= \Pr_{a,b,r}[(a,b,C(h_{a,b}(r)))\in V_y] - \Pr_{a,b,r}[(a,b,C(h_{a,b}(r)))\in F\cap V_y]\\
&\geq \Pr_{a,b,r}[(a,b,C(h_{a,b}(r)))\in V_y] - (1-2^{-\frac{\epsilon}{4}})\Pr_{a,b,r}[(a,b,C(h_{a,b}(r)))\in V_y] \quad (\because y\notin B_C)\\
&= 2^{-\frac{\epsilon}{4}}\cdot\Pr_{a,b,r}[(a,b,C(h_{a,b}(r)))\in V_y]. \qquad (17)
\end{aligned}$$

Inequality (14) is shown as follows: for each $j\in[M]$,

$$\begin{aligned}
\Pr[E_j] &= \Pr_{a,b,\mathcal{I}}[\mathcal{I}(C,a\circ b\circ y\circ t_y) \text{ succeeds in inverting}]\\
&\geq \Pr_{a,b,\mathcal{I}}[\mathcal{I}(C,a\circ b\circ y\circ t_y) \text{ succeeds in inverting}\wedge (a,b,y)\in V_y\setminus F]\\
&= \Pr_{a,b}[(a,b,y)\in V_y]\cdot\Pr_{a,b}[(a,b,y)\in V_y\setminus F|(a,b,y)\in V_y]\\
&\qquad\cdot\Pr_{a,b,\mathcal{I}}[\mathcal{I}(C,a\circ b\circ y\circ t_y) \text{ succeeds in inverting}|(a,b,y)\in V_y\setminus F]\\
&\geq \Pr_{a,b}[(a,b,y)\in V_y]\cdot\Pr_{a,b}[(a,b,y)\in V_y\setminus F|(a,b,y)\in V_y]\cdot\frac{1}{2^{\epsilon/4}} \quad (\because (a,b,y)\notin F)\\
&\geq \Pr_{a,b}[(a,b,y)\in V_y]\cdot\frac{1}{2^{\epsilon/4}}\cdot\frac{1}{2^{\epsilon/4}} \quad (\because (17))\\
&\geq \frac{1}{2}\cdot\frac{1}{2^{\epsilon/4}}\cdot\frac{1}{2^{\epsilon/4}} = \frac{1}{2}\cdot\frac{1}{2^{\epsilon/2}} \quad (\because (15))
\end{aligned}$$

□

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il