Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 95 (2020)

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

Mikito Nanashima
Tokyo Institute of Technology, Japan
https://nanashima.github.io
nanashima.m.aa@is.c.titech.ac.jp

Abstract

Constructing one-way functions based on NP-hardness is a central challenge in theoretical computer science. Unfortunately, Akavia et al. [2] presented strong evidence that a nonadaptive black-box (BB) reduction is insufficient to solve this challenge. However, should we give up such a central proof technique even for an intermediate step?

In this paper, we turn our eyes from standard cryptographic primitives to weaker cryptographic primitives allowed to take auxiliary-input and continue to explore the capability of nonadaptive BB reductions to base auxiliary-input primitives on NP-hardness. Specifically, we prove the followings:

- 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).

These theorems extend our knowledge on nonadaptive BB reductions out of the current worst-to-average framework. The first result provides new evidence that nonadaptive BB reductions are insufficient to base AIPRG on NP-hardness. The second result also yields a weaker but still surprising consequence of nonadaptive BB reductions, i.e., a one-way function based on NP-hardness. In fact, the second result is interpreted in the following two opposite ways. Pessimistically, it shows that basing AIOWF or AIHSG on NP-hardness 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 provides conceptually stronger limitations than the currently known limitations on nonadaptive BB reductions. Optimistically, it offers a new hope: breakthrough construction of auxiliary-input primitives might also provide construction 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.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Auxiliary-input cryptographic primitives, nonadaptive black-box reductions

Funding *Mikito Nanashima*: JST, ACT-X Grant Number JPMJAX190M, Japan.

1 Introduction

How can we translate computational hardness into cryptography? This is a central question in theoretical computer science. Specifically, one of the most significant and long-standing challenges on this question is constructing fundamental cryptographic primitives such as a one-way function based on NP-hardness. At the moment, several breakthroughs seem to be required for solving this challenge, as surveyed by Impagliazzo [24].

ISSN 1433-8092

A central ingredient for solving 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 is sufficient to show many brilliant results and has played a crucial role in theoretical computer science. Therefore, it is a natural attempt to apply such a familiar proof technique even for constructing secure cryptographic primitives.

However, Akavia et al. [11] presented strong evidence that such a simple reduction is insufficient for cryptography based on NP-hardness. Generally, 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. As a corollary, their work excluded the attempt to apply nonadaptive BB reductions for cryptography based on NP-hardness under the reasonable assumption that the polynomial hierarchy does not collapse. Further, subsequent work provided stronger consequences in more specific cases of several cryptographic primitives [2, 15, 5, 16, 9, 8, 31, 20].

Then should we also give up all nonadaptive BB strategies even for an intermediate step towards cryptography? This question originally motivated our work. In this spirit, we focus on the capability of nonadaptive BB reduction for a weaker cryptographic notion, i.e., an *auxiliary-input* cryptographic primitive introduced first by Ostrovsky and Wigderson [33]. 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 must be secure depending on each adversary (instead of one specific primitive secure against all adversaries). In other words, adversaries for auxiliary-input primitives must break all primitives in the worst-case sense on auxiliary-input. This task is not directly formulated as a distributional NP-problem because the distribution is not uniquely determined in advance due to auxiliary-input. Thus, the previous work on distributional NP problem cannot be directly applied to auxiliary-input cryptography.

Herein, we present the current status of nonadaptive BB reductions to auxiliary-input cryptography. Applebaum et al. [5] observed that nonadaptive fixed-auxiliary-input BB reductions are insufficient even for auxiliary-input cryptography unless the polynomial hierarchy collapses. Their reduction is a restricted case of nonadaptive BB reduction where only one auxiliary-input is accessible. However, this restricted access to auxiliary-input seems too strict and implicitly yields a reduction from an NP-hard language to some fixed cryptographic primitive (depending on the instance). In fact, this result was shown in almost the same way to the previous result for standard cryptographic primitives in [2]. The same work and later Xiao [38] observed that generalizing their result to nonadaptive BB reductions seems hard by giving the explicit technical issue. To the best of our knowledge, we had no negative result on general nonadaptive BB reductions to base auxiliary-input cryptography on NP-hardness before this work. For more detailed reason why the previous work such as [11, 2] is not applicable for auxiliary-input primitives, refer to Section 4.

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 [3, 19]. However, such an implication requires adaptive techniques at present (e.g., [18]). Thus, the property of nonadaptive 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 nonadaptive BB reductions for auxiliary-input cryptographic primitives based on NP-hardness. 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 in [11] and to identify the inherent difficulty on constructing cryptographic primitives on NP-hardness more finely.

1.1 Our Contribution

Our main contribution is to provide 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 definition of each primitive will be presented in Section 2 with a formal description of our main results. First, we informally state the main theorem as follows.

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

- *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 provides 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. Note 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 [32]. What is surprising is that even a nonadaptive BB reduction to such a weak primitive yields a solution close to the long-standing challenge, i.e., characterization of one-way functions based on NP-hardness.

The second result is not sufficient to exclude nonadaptive BB reductions which base auxiliary-input primitives on NP-hardness, 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 of 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., [18, 23, 17]) use adaptive techniques in the security proof; for instance, construction of false entropy generators and the uniform hardcore lemma [22]. 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 [5] on hardness of learning because we need to construct AIPRG first to show the hardness of learning from the existence of AIOWF.

2 Formal Descriptions

In this section, 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 denote 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) denote a distributional problem of recognizing $L(x)$ on an instance x selected uniformly at random (for the detail, refer to Section 6.2). 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*[1] *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 the sake of 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, i.e., $|z| > n(|z|)$ 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 all auxiliary-inputs $z \in \{0,1\}^*$,*

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

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

A BB reduction for AIPRG is defined as follows. It is easily verified 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 all polynomials p, there exists a randomized polynomial-time oracle machine $R^?$ such that for all oracles $\mathcal{O}$ that $(1/p)$-distinguish G and $x \in \{0,1\}^*$, R satisfies that*

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

[1] 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 $n(|z|) \leq c' \cdot |z|^{c'}$.

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

The first main result on AIPRG is stated as follows.

▶ **Theorem 4.** *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 is easy to compute but hard to invert, and it is a fundamental primitive in the sense that most cryptographic tools do not exist without a one-way function [27, 35]. The formal definition is the following:

▶ **Definition 5** (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)[2] if f is polynomial-time computable, and there exists a polynomial p such that for all polynomial-time randomized algorithms A, there exist 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 the sake of simplicity, we may omit to write the input 1^n to A.

The auxiliary-input analog of OWF, first introduced by Ostrovsky and Wigderson [33], is defined as follows.

▶ **Definition 6** (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 all $z \in \{0,1\}^*$,*

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

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 [18]. However, we cannot directly apply Theorem 4 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 verified 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 7** (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 all oracles $\mathcal{O}$ that $(1-1/p)$-invert f and $x \in \{0,1\}^*$, R satisfies that*

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

[2] Strictly speaking, a one-way function defined in Definition 5 is usually called a "weak" one-way function, which implies the standard (strong) one-way function.

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

The second main result on AIOWF is stated as follows.

▶ **Theorem 8.** *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 $\mathsf{NP} \not\subseteq \mathsf{BPP}$ 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. [4]. For the original purpose, they considered (possibly) exponential-time computable generators. In this paper, we focus on polynomial-time computable generators as in cryptography. We define the auxiliary-input analog as follows.

▶ **Definition 9** (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 all (public) auxiliary-inputs $z \in \{0,1\}^*$ and (private) inputs $x \in \{0,1\}^{n(|z|)}$,*

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

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 verified that AIPRG is also AIHSG (for 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 [32]; 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 [39]. 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 verified 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 10** (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 all oracles $\mathcal{O}$ that $(1-\gamma)$-avoid 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.

The third main result on AIHSG is stated as follows.

▶ **Theorem 11.** *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* $\mathsf{NP} \not\subseteq \mathsf{BPP}$ *also implies that a one-way function exists (via an adaptive BB reduction).*

3 Discussion and Future Directions

As discussed in Section 1.1, Theorems 8 and 11 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 suggest 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) via 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 quite non-trivial, as mentioned in [5, 37]. In this paper, we also provide a simple oracle separation between AIOWF and OWF as follows. This indicates that we cannot expect any relativized technique to remove auxiliary-input from cryptographic primitives.

▶ **Theorem 12.** *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.*

Additionally, there are several barriers by other oracle separations at the intermediate levels to base OWF on NP-hardness (e.g., [39, 25]). Although such barriers on relativization are common throughout theoretical computer science (e.g., the P vs. NP problem [6]), there are only a few success stories of overcoming such barriers at present. Unfortunately, Theorems 8 and 11 do not provide 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 12 or the oracle separation presented in [25].

However, our result offers one hope. Although there seems to be several barriers towards cryptography based on NP-hardness as discussed above, the essential barrier we must overcome might be few. Theorems 8 to 12 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 difficulty in basing OWF on NP-hardness could rely on a much smaller part of tasks at an intermediate level. This conjecture seems somewhat controversial but enhances the significance of further investigation on 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

[4] Note that oracle separations do not necessarily rule out BB reductions from particular languages, not as fully BB reductions defined in [34].

that nonadaptive BB reductions for 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 [2] or HSG [20] to each auxiliary-input analog for the stronger consequence. At least the latter approach seems to require some new technique to simulate nonadaptive BB reductions by constant-round interactive proof systems, as observed in [5] and [38].

4 A First Attempt: Applying [11] and [2]

Before presenting our proof strategies, we roughly explain why the previous technique developed by Bogdanov and Trevisan [11] for the worst-to-average framework is not applicable in the case of auxiliary-input cryptography. For the sake of simplicity, we assume that there exists a nonadaptive BB reduction R from an NP-hard language L to a distributional NP-problem (L', U), and R makes queries of the same length n determined by the size of input to R. Note that if we can answer these queries by an oracle which correctly recognizes L' on average, then R must recognize L. Bogdanov and Trevisan construct an $\mathsf{AM/poly}$ protocol for recognizing L by leaving this role of the oracle to a prover, which implies that $\mathsf{coNP} \subseteq \mathsf{AM/poly}$ and the collapse of the polynomial hierarchy.

Roughly speaking, their central idea is to divide each R's query $x \in \{0,1\}^n$ into "light" and "heavy" queries according to the probability p_x that the query x is generated by R. Specifically, they determine a threshold $p(n) = \mathsf{poly}(n)$ depending on the permissible error probability for solving (L', U) on average and define a light (resp. heavy) query x as a query satisfying the condition $p_x \leq p(n)2^{-n}$ (resp. $p_x > p(n)2^{-n}$). Then, they make the prover answer (ideally) all light queries correctly, i.e., simulate the following oracle.

$$\mathcal{O}_{\mathcal{L}} = \{x \in \{0,1\}^* : x \in L' \text{ and } x \text{ is a light query}\}$$

Because the number of heavy queries is at most $2^n/p(n)$, the above oracle $\mathcal{O}_{\mathcal{L}}$ solves (L', U) with error probability at most $p(n)$. Therefore, it is enough to make a prover simulate $\mathcal{O}_{\mathcal{L}}$ for constructing an $\mathsf{AM/poly}$ protocol which recognizes L based on R. For the soundness, the verifier must accomplish the following two tasks without deceived by malicious provers: (1) distinguishing between light and heavy queries and (2) identifying the correct answer for each light query. Bogdanov and Trevisan developed such a verifier by introducing sophisticated protocols called the heavy sampling protocol and the hiding protocol.

Herein, we consider the case of auxiliary-input primitives, where each R's query takes the form of (z, x) where z denotes auxiliary-input. For the sake of simplicity, we assume that the task of breaking an auxiliary-input primitive is further reduced to an average-case deterministic problem on uniform distribution with auxiliary-input by applying the techniques in [26, 7] as in the previous work [11] and the length of instances of the average-case problem is the same as the length of auxiliary-input. We also assume that a reduction R makes queries of the form (z, x) where $|z| = |x| = n$ and n is determined by the size of input to R.

There are two possible ways to extend the above idea to the case of auxiliary-input primitives: considering auxiliary-input in queries (a) together or (b) separately. For the first approach (a), we must determine a light query as a query satisfying the condition $p_x \leq \mathsf{poly}(n)2^{-2n}$ for applying the hiding protocol. This is problematic because the number of heavy queries is possibly $2^{2n}/\mathsf{poly}(n)$, and many of them can be concentrated on one auxiliary-input. In other words, the above oracle $\mathcal{O}_{\mathcal{L}}$ does not always solve the average-case problem in the worst-case sense on auxiliary-input. On the other hand, for the second approach (b), their verifier needs information on some statistics as advice for each auxiliary-input. Because there are exponentially many possibilities on auxiliary-input, the total length of such advice is exponentially large, which is unfeasible as an $\mathsf{AM/poly}$ protocol.

The subsequent work [2] provided the method to remove the above advice in the case of standard cryptographic primitives by applying an additional property of breaking cryptographic primitives (therefore, they constructed an AM protocol for $\neg L$ instead of an $\mathsf{AM/poly}$ protocol). Unfortunately, even this method cannot be applied directly in the case of auxiliary-input cryptography. To obtain the statistics corresponding to the above advice, their protocol needs to generate query set by executing R. In our case, remember that we consider each auxiliary-input separately, so we need to simulate a conditional distribution on queries for fixed auxiliary-input. However, such distributions are not efficiently samplable in general: for example, consider the query distribution on $(h(y), y)$ where h is a collision-free hashing function. Then, a polynomial-time verifier which simulates a conditional distribution for a fixed auxiliary-input (i.e., hash value) can easily find the collision of h.

5 Proof Sketches

We give proof ideas of Theorems 4, 8, 11, and 12, and each formal proof will be given in Sections 7 to 10, respectively. Note that Theorem 11 heavily relies on Theorem 8, and Theorem 8 heavily relies on Theorem 4. Therefore, although each proof idea may look pretty simple and intuitive, our construction of OWF for Theorem 11 becomes complicated and quite non-trivial as a whole.

5.1 The Case of AIPRG: Proof Idea of Theorem 4

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

▶ **Definition 13** (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, i.e., satisfying the condition that for all sufficiently large $n \in \mathbb{N}$,*

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

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

Theorem 4 essentially follows from a nonadaptive BB security reduction from distinguishing AIPRG to avoiding HSG. Note that HSG based on AIPRG with a nonadaptive BB security reduction has been implicitly given in the study on MCSP [3, 19]. To see this explicitly, we will provide a much simpler construction of HSG based on AIPRG and a self-contained proof. 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 direct relationship between AIPRG and HSG.

First, we assume that there is 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 19). 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 [11] (formally, Fact 2), such a reduction implies the collapse of the polynomial-hierarchy.

Our construction of HSG from AIPRG is the following (formally, Lemma 18): 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 $z + n(|z|) \geq \ell(n(|z|))$ 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. It can be easily verified 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 employing an adversary A for G' as an adversary for G. Obviously, this reduction is nonadaptive. To show the correctness, assume that A γ-avoids G'. For the sake of 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, if 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.

5.2 The Case of AIOWF: Proof Idea of Theorem 8

In this section, we omit all arguments about the success probabilities of adversaries to focus on the proof idea. First, we introduce several reductions as elements of a standard OWF. Let $R_{L\to f}$ denote the nonadaptive BB reduction from L to inverting f in the assumption. By the construction of PRG from OWF (e.g., [18]), 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 5.1, 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. if $R_{L\to f}$ outputs $b \in \{0,1\}$, then output the same decision b.

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

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 the following 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 $\mathsf{NP} \not\subseteq \mathsf{BPP}$. For contradiction, we assume that there exists an adversary A that inverts g. Remember that g simulates a distribution on queries produced by $R_{L\to f}$ 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. This is a little technical part, and we will present further detail in Section 8. 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 polynomial-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 5.1. 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 4. Thus, finding such a simple construction of AIPRG is one direction for excluding a nonadaptive BB reduction to base AIOWF on NP-hardness, as mentioned in Section 3.

5.3 The Case of AIHSG: Proof Idea of Theorem 11

Our goal is to simulate an avoiding oracle for a nonadaptive BB reduction by another protocol in some restricted complexity class, in our case, BPP. The key idea for this is to classify each query generated by the nonadaptive BB reduction into "light" and "heavy" queries as in [11]. A similar technique was also applied in the previous work for HSG [15, 20]. Thus, we first review the previous case of HSG and then explain the difference to our case of AIHSG.

The Case of Hitting Set Generator (Previous work)

Let $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ denote a generator with $\ell(n) \geq (1 + \Omega(1)) \cdot n$ and $R^?$ denote a nonadaptive BB reduction from an NP-language L to avoiding G. W.l.o.g., we can assume that marginal distributions on each query by R are identical regardless of each query position by applying a random permutation on query positions before asking them to oracle. Thus, for each input $x \in \{0,1\}^n$ to R, one marginal distribution Q_x on R's 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.

We simulate the avoiding oracle for G by using the classification of queries as follows. 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 $\mathcal{O}'$ denote the induced oracle by the above simulating procedure. Note that the probability that $\mathcal{O}'(y)$ outputs 0 is exponentially small because the fraction of light queries is $\tilde{\Theta}(2^n)/2^{\ell(n)} \leq 2^{-\Omega(n)}$. Thus, $\mathcal{O}'$ satisfies the condition on the probability of outputting 1. However, $\mathcal{O}'$ is not avoiding oracle for G, because there is possibly a query y such that y is heavy but contained in $\mathrm{Im}G$. In this case, $\mathcal{O}'(y)$ outputs 1 even for $y \in \mathrm{Im}G$ and fails to avoid G.

The key observation to overcome this issue is the following:

($\star$) For each length $\ell(n)$ of query (i.e., 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 $\mathrm{Im}G$ (we refer to it as a "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 [15] presented a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm by approximation of counting in [30], and Hirahara and Watanabe [20] presented an $\mathsf{AM} \cap \mathsf{coAM}$ algorithm by generalizing the protocol in [12].

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

We move on to our case of AIHSG. Let $G = \{G_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{\ell(n(|z|))}\}_{z\in\{0,1\}^*}$ denote an auxiliary-input generator with $\ell(n) \geq (1 + \Omega(1)) \cdot n$ and $R^?$ denote a nonadaptive BB reduction from an NP-language L to avoiding G. We can also assume that all marginal query distributions of $R^?(x)$ are identical to Q_x regardless of query position.

To extend 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 overcome this, we need to consider each case of auxiliary-input z separately. Therefore, we change the definitions of "light" and "heavy" queries depending on auxiliary-input. Let $p_x(z)$ denote 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 the auxiliary-input in the query is z. In fact, this modification will work well even for AIHSG (for the formal argument, refer to Section 9).

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 yield any nontrivial result, not as the work in [15][5]. Even for the simulation in $\mathsf{AM} \cap \mathsf{coAM}$ in [20], there are 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 developed in [1]. Moreover, we cannot possibly even sample the conditional distribution efficiently for fixed auxiliary-input, as discussed in Section 4.

Our idea is to adopt universal extrapolation in [26]. 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 a polynomial-time computable f and a given $y = f(x)$ where $x \in \{0,1\}^n$ is selected at random. In fact, the universal extrapolation holds even for an auxiliary-input function, and a similar technique was also used in [33]. By using the universal extrapolation for each circuit which generates query and auxiliary-input, we have a good approximation of

[5] Their work concerned the original aim of HSG, i.e., derandomization (e.g., [29]). For this purpose, they considered (possibly) exponential-time computable HSG G, where 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.

$p_x(y|z)$ 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 11, 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 8, R also yields a one-way function under the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

5.4 Oracle Separation between OWF and AIOWF: Proof Idea of Theorem 12

To show Theorem 12, we employ 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 in [28], 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., [39]). 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 all auxiliary-inputs 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 (standard) 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 $\mathcal{O}_\mathcal{F}$ denote this 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 32). Thus, AIOWF still exists relative to $\mathcal{O}_\mathcal{F}$.

On the other hand, we 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 verify 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 33). Thus, we have the oracle separation between AIOWF and OWF.

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

6 Preliminaries

For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. For two strings $x, y \in \{0,1\}^*$, let $\langle x, y \rangle \in \{0,1\}^*$ denote 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 such 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 (i.e., $0 \le x_{\mathbb{N}} \le 2^n - 1$).

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 $e(S(n))(= O(S(n) \log S(n)))$.

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 [21].

▶ **Fact 1** (Hoeffding inequality). *For real values $a, b \in \mathbb{R}$, let $X_1, \ldots, X_m$ denote 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:*

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

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

▶ **Lemma 14.** *Let X and Y denote (possibly correlated) random variables on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let E denote 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*

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

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

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:

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

◀

6.1 Universal Extrapolation

We formally introduce the key ingredient for our proof, i.e., the universal extrapolation.

▶ **Lemma 15** (Universal extrapolation [26]). *Let $\epsilon \in (0,1]$ and $\delta : \mathbb{N} \to (0,1]$ denote 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 $s(n)$,*

$$\Pr_{Ext_{s(n)}, x \sim \{0,1\}^n}\left[Ext_{s(n)}(C, C(x)) \in [p_C(x), 2^{(3+\epsilon)} p_C(x)]\right] \geq 1 - \delta(n),$$

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 from the original work [26]. Although the proof sketch for the original statement was presented in [26], the author could not unfortunately find the full version of the proof. To show the correctness and nonadaptiveness explicitly, we will also present the full proof of Lemma 15 in Appendix A based on the original proof sketch.

6.2 Average-case Complexity

We introduce the basics of average-case complexity. For further details, refer to the survey [10].

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 16** (Errorless heuristic scheme). *Let (L, D) denote 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 the condition that for any $n \in \mathbb{N}$,*

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

▶ **Definition 17** (Randomized errorless heuristic scheme). *Let (L, D) denote 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 the condition that for any $n \in \mathbb{N}$,*

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

The previous work [11] ruled out nonadaptive BB reductions from an NP-hard problem to a distributional NP problem, usually called worst-case to average-case reductions, unless the polynomial hierarchy collapses at the third level.

▶ **Fact 2** ([11]). *For any polynomial p, language L, and 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 $\mathsf{PH} = \Sigma_3^p$.*

7 On Basing Auxiliary-Input Pseudorandom Generator on NP-hardness

In this section, we formally rule out nonadaptive BB reductions from an NP-hard problem to distinguishing AIPRG based on Section 5.1. Let us state the main theorem again.

▶ **Theorem 4.** *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 present the nonadaptive BB reduction from distinguishing AIPRG to avoiding HSG.

▶ **Lemma 18.** *Let G denote 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 condition: 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 $b_i \in \{0,1\}$ denote 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 verified 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 ←_u [m];
let x^(k) := y_[n(a)];
for i = 1 to m do
    if i < k then select σ_i ←_u {0,1};
    else if i = k then σ_k = y_{n(a)+1};
    else execute G_z(x^(i−1)) →_{n(a),1} (x^(i), σ_i);
end
query b ← O(σ_1 ∘ ··· ∘ σ_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 (e.g., refer to [13]), 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 implies that $R^{\mathcal{O}}$ $1/\gamma$-distinguishes G for any $\mathcal{O}$ $1/\gamma'$-avoiding G'. ◀

Lemma 18 also implies a nonadaptive BB reduction from distinguishing AIPRG to a distributional NP problem.

▶ **Lemma 19.** *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. We define polynomials γ' and m as $\gamma'(n) = \frac{\delta(2n)}{\delta(2n)-2}$ and $m(n) = 2n$. By Lemma 18 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 $L := \mathrm{Im}G' = \{G'(x) : x \in \{0,1\}^*\}$. Because G' is polynomial-time computable, $L \in \mathsf{NP}$ holds.

Since δ is polynomial, there exists $n_0 \in \mathbb{N}$ such that $2^{n/2} \geq \delta(n)$ 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 $1/\delta$)

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

We will 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 with R_2, we can also construct 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 Definition 13. Therefore, we consider only the case where $2n \geq n_0$.

We assume that the given oracle $\mathcal{O}$ is an errorless heuristic scheme of failure probability at most $1/\delta$, then $\mathcal{O}$ must satisfy the following conditions:

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

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

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

◀

Lemma 19 and Fact 2 immediately imply Theorem 4 as follows.

Proof of Theorem 4. By Lemma 19, 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. ◀

8 On Basing Auxiliary-Input One-Way Function on NP-hardness

In this section, we formally show Theorem 8 based on the idea in Section 5.2.

▶ **Theorem 8.** *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 19 in Section 7.

▶ **Lemma 20.** *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', $R^{\mathcal{O}}$ $(1-\delta)$-inverts f.*

Proof. The lemma follows from Lemma 19 and the construction of auxiliary-input pseudorandom generator based on an auxiliary-input (weak) one-way function (e.g., [18]). ◀

▶ Note 21. The reader may wonder whether the Goldreich-Levin theorem [14] is sufficient to show Lemma 20. In fact, it seems not directly applicable in our case because such a technique does not eliminate the worst-case condition on auxiliary-input.

Now we present the full proof of Theorem 8.

Proof of Theorem 8. Let $R_{L\to f}$ denote 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 all marginal distributions on query

[6] By more careful simulation technique for HSG in [20], we can improve the consequence up to the collapse of the polynomial hierarchy at the second level.

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 20, 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. Because 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., we assume that $|R_{L'\to L}(x)| \le p(|x|)$ 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 $x \in \{0,1\}^m$, notice that 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 $\mathsf{NP} \subseteq \mathsf{BPP}$. This immediately implies Theorem 8.

For the sake of simplicity, we consider that g_m takes as input a triple of length m, $r(m)$, and $N(m) := n(a(m))$, respectively. We 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

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

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

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

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 $B^{L'}_m, G^{L'}_m$, and $N^{L'}_m$ (which stand for bad, good, and neutral, respectively) as

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

For 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

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

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

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

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

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

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 22. 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 22. Fix $m \in \mathbb{N}$, $x \in G_m^{L'}$, and $s_A \in G_{m,x}^A$ arbitrarily. By the definition of $\mathcal{O}_{m,x,s_A}$, if $z \in B_{m,x,s_A}^f \cup \{0,1\}^{>a(m)}$, then $\mathcal{O}_{m,x,s_A}(z,y)$ 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 22, 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 NP ⊆ BPP. Therefore, the remaining part is to construct the randomized errorless heuristic scheme B for (L', U).

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

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 the probability that B outputs $\perp$.

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 $c := 0$, $M := \frac{2^{21} \cdot q(m)^2}{\delta(N(m))^2}$;
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;

▷ Claim 23. **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 23. (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

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

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,

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

◁

By Claim 23, we can show the following claims:

▷ Claim 24. $\Pr_{x\sim\{0,1\}^m}[x \in B_m^{L'} \cup N_m^{L'}] \leq \delta(m)$.

▷ Claim 25. If $x \in G_m^{L'}$, then $\Pr_B[B(x) = L'(x)] \geq 3/4$.

▷ Claim 26. 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, Claims 23-(1), 25, and 26 imply that 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 23-(1), 25, and 26 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 24, 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 24, 25, and 26.

Proof of Claim 24. 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 the following expression:

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

By Lemma 14, we have that $\Pr_{x\sim\{0,1\}^m}[x \in B_m^{L'} \cup N_m^{L'}] \leq \delta(m)$. ◁

Claims 25 and 26 are immediately derived from the following Claim 27. Therefore, we first show Claims 25 and 26 by assuming Claim 27 then we show Claim 27.

▷ Claim 27. If $x \in G_m^{L'} \cup N_m^{L'}$, then $\Pr_B[B(x) = \neg L'(x)] \leq 3/16$.

Proof of Claim 25. If $x \in G_m^{L'}$, then

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

where the inequality follows from Claims 23-(2) and 27. ◁

Proof of Claim 26. If $x \in N_m^{L'}$, then

$$\Pr_B[B(x) \in \{L'(x), \perp\}] = 1 - \Pr_B[B(x) = \neg L'(x)] \geq 13/16 > 3/4,$$

where the first inequality follows from Claim 27. ◁

Proof of Claim 27. 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 14, the following expression holds:

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

Thus, we assume that B succeeds in selecting a good $s_A \in G^A_{m,x}$. By Claim 22, 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. In the following, we show that answer by $A^f(\,\cdot\,;s_A)$ is indeed consistent with answer by $\mathcal{O}_{m,x,s_A}$ with high probability over the choice of random bits for $R_{L\to f}$ to provide the upper bound on the probability that B outputs $\neg L'(x)$.

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) $z \in B^f_{m,x,s_a}$ 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 present the upper bound on the probability that the event (a) occurs. For a randomness $s \in \{0,1\}^{r(m)}$ to execute $R_{L\to f}$, define a bad set $B^{R_{L\to f}}_{m,x,s_A}$ by

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

Since $s_A \in G^A_{m,x}$, we have that

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

By Lemma 14,

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

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,

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

Because 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

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

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

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

By the union bound, we conclude that

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

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

◀

9 On Basing Auxiliary-Input Hitting Set Generator on NP-hardness

In this section, we formally show Theorem 11 based on the idea in Section 5.3.

▶ **Theorem 11.** *Let p denote a polynomial and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ denote 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 $\not\subseteq$ BPP *also implies that a one-way function exists (via an adaptive BB reduction).*

Theorem 11 obviously follows from Lemma 28 and Theorem 8.

▶ **Lemma 28.** *Let δ denote a reciprocal of polynomial and $G := \{G_z : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{z\in\{0,1\}^*}$ denote 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 28. Let $\epsilon' = \epsilon/2$ and $R^?$ denote 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 $q(m)$ marginal 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 denote the distribution on the first query by $R(x)$ (which is identical to query distributions in other query positions). Let $Q_x^{(1)}$ denote 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:

$$\begin{aligned} H_x &:= \left\{(z,y) \in \{0,1\}^a \times \{0,1\}^\ell : p_x(y|z) > \frac{256}{2^{(1+\epsilon')n}}\right\}, \\ L_x &:= \left\{(z,y) \in \{0,1\}^a \times \{0,1\}^\ell : p_x(y|z) \leq \frac{1}{2^{(1+\epsilon')n}}\right\}, \\ M_x &:= \left(\{0,1\}^a \times \{0,1\}^\ell\right) \setminus (H_x \cup L_x), \end{aligned}$$

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 the following conditions: for any $z \in \{0,1\}^a$,

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

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

▷ Claim 29. For any $x \in \{0,1\}^m$ and $a \in \mathbb{N}$, if $n(a) \geq n_0$, 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. Because 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

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

Because T also satisfies the condition 3,

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

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,

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

Hence, we have that

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

Therefore,

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

◁

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

$$\mathcal{O}_{\{T_a\}}(z,y) = \begin{cases} 1 & \text{if } |y| \neq \ell(n(|z|)) \\ \mathbb{1}\{y \in \mathrm{Im}(G_z)\} & \text{if } |y| = \ell(n(|z|)) \wedge (2^{\epsilon' n(|z|)} \leq 2^{12} \cdot q(m) \vee n(|z|) < n_0) \\ T_{|z|}(z,y) & \text{otherwise.} \end{cases}$$

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 29 implies that any $\mathcal{O} \in \mathcal{T}$ $(1-\delta)$-inverts G and 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 generate its $q(m)$ queries, where $r(\cdot)$ is a polynomial. Let $s(\cdot)$ denote 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 15. 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 $(z_1,y_1),\dots,(z_{q(m)},y_{q(m)})$;
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* $2^{\epsilon' n_i} \leq 2^{12}\cdot q(m)$ **then** find the answer by brute-force search and return it (note that the latter condition implies that $2^{n_i} \leq (2^{12}\cdot q(m))^{1/\epsilon'} \leq \mathsf{poly}(m)$);
9   **else if** $\frac{\tilde{p}_i}{\tilde{p}'_i} \leq \frac{16}{2^{(1+\epsilon')n_i}}$ **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 verify 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 following conditions: 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 $i \in [q(m)]$,

$$\frac{1}{16} \cdot p_x(y_i|z_i) \le \frac{\Pr[(z_i, y_i) \leftarrow Q_x]}{16 \cdot \Pr[z_i \leftarrow Q_x^{(1)}]} \le \frac{\tilde{p}_i}{\tilde{p}'_i} \le \frac{16 \cdot \Pr[(z_i, y_i) \leftarrow Q_x]}{\Pr[z_i \leftarrow Q_x^{(1)}]} \le 16 \cdot p_x(y_i|z_i).$$

Therefore, we have that

$$(z_i, y_i) \in L_x \Longrightarrow \frac{\tilde{p}_i}{\tilde{p}'_i} \le 16 \cdot p_x(y_i|z_i) \le \frac{16}{2^{(1+\epsilon')n}},$$

and

$$(z_i, y_i) \in H_x \Longrightarrow \frac{\tilde{p}_i}{\tilde{p}'_i} \ge \frac{1}{16} \cdot p_x(y_i|z_i) > \frac{16}{2^{(1+\epsilon')n}}.$$

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 a $(1-\delta)$-inverting oracle for G. This motivates us to show the following claim.

▷ Claim 30. 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

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

First, we assume that Claim 30 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 30;
3. $R^{\mathcal{O}}(x) = L(x)$;
By Claim 30 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 $x \in \{0,1\}^*$,

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

Thus, the remaining part is to show Claim 30.

Proof of Claim 30. 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 verified that $T_a \in \mathcal{T}_{x,a}$. Therefore, $\mathcal{O} \in \mathcal{T}_x$ holds.

We have that

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

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

$$\begin{aligned}
&\mathcal{O}(z,y) \neq \mathcal{O}'(z,y)\\
&\quad\Longrightarrow 2^{\epsilon' n(|z|)} > 2^{12} \cdot q(m) \text{ and } T_{|z|}(z,y) \neq T'_{|z|}(z,y) \quad (\because \text{definitions of } \mathcal{O} \text{ and } \mathcal{O}')\\
&\quad\Longleftrightarrow 2^{\epsilon' n(|z|)} > 2^{12} \cdot q(m) \text{ and } y \in \mathrm{Im}(G_z) \text{ and } T'_{|z|}(z,y) = 1 \quad (\because \text{definitions of } T_a \text{ and } T'_a)\\
&\quad\Longrightarrow 2^{\epsilon' n(|z|)} > 2^{12} \cdot q(m) \text{ and } y \in \mathrm{Im}(G_z) \text{ and } (z,y) \notin H_x \quad (\because \text{definition of } \mathcal{T}'_x).
\end{aligned}$$

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]\\
&\quad= \Pr_{(z,y)\leftarrow Q_x}\left[\mathcal{O}(z,y) \neq \mathcal{O}'(z,y)\right]\\
&\quad\leq \Pr_{(z,y)\leftarrow Q_x}\left[2^{\epsilon' n(|z|)} > 2^{12} \cdot q(m) \text{ and } y \in \mathrm{Im}(G_z) \text{ and } (z,y) \notin H_x\right]\\
&\quad= \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\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]\\
&\quad= \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\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}[y' = y | z' = z] \cdot \Pr_{z'\leftarrow Q_x^{(1)}}[z' = z]\\
&\quad= \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\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)}]\\
&\quad\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\cdot q(m)}} \sum_{\substack{y\in\mathrm{Im}(G_z):\\ (z,y)\in L_x\cup M_x}} \frac{256}{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)\\
&\quad\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\cdot q(m)}} |\mathrm{Im}(G_z)| \cdot \frac{256}{2^{(1+\epsilon')n(|z|)}} \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}]\\
&\quad\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\cdot q(m)}} \frac{256}{2^{\epsilon' n(|z|)}} \cdot \Pr_{Q_x^{(1)}}[z \leftarrow Q_x^{(1)}] \quad (\because |\mathrm{Im}(G_z)| \leq 2^{n(|z|)})\\
&\quad\leq \sum_{\substack{z\in\{0,1\}^*:\\ 2^{\epsilon' n(|z|)}>2^{12}\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))\\
&\quad\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

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

◁

◀

10 Oracle Separation between AIOWF and OWF

To show Theorem 12, we introduce the auxiliary-input analog of the random function oracle.

▶ **Definition 31** (Auxiliary-input embedded random function). *Let r, r' denote 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 $|z| = |x'|$. For convenience, 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 an 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 32.** *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 $z_n \in \{0,1\}^n$ denotes a target auxiliary-input of $\mathcal{F}$.

The above lemma is shown essentially by the same argument by [36, Section 6]. For completeness, we will give the full proof of Lemma 32 later.

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

▶ **Lemma 33.** *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 present the proof of Theorem 12 by assuming Lemmas 32 and 33.

Proof of Theorem 12. 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 32 and 33. 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 33 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 33, 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 present the proof of Lemma 32.

Proof of Lemma 32. In this proof, we use the notation poly to denote a certain polynomial. For the sake of simplicity, we regard the oracle access to $\mathcal{O}_\mathcal{F}$ as access to two oracles $\mathcal{F}$ and TQBF.

We fix a polynomial-time oracle machine $A^?$ arbitrarily. For any $n \in \mathbb{N}$ and $y \in \{0,1\}^n$, we 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 a simple calculation, $\Pr_\mathcal{F}[B_y] \le C \cdot 2^{-n}$ for some constant C (as shown in the last of the proof).

Let $q(n)$ denote 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

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

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 $s \in \{0,1\}^{\mathsf{poly}(n)}$,

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

Therefore, we have that

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

For any sufficiently large $n \in \mathbb{N}$, the above probability is less than $n^{-2} \cdot 2^{-n/2}$. By Lemma 14,

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

for any sufficiently large n.

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

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. Because 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 32.

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):

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

◀

To prove Lemma 33, we first show the following key lemma.

▶ **Lemma 34.** *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 $z_m \in \{0,1\}^m$ denotes 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. Because 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 refer to such queries as "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 14,

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

For each n, let E_n denote the above event that

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

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

Note that the above argument holds for any tuple (A, c) of an oracle machine and an integer. Now we ignore the measure 0 of oracles which satisfies the above condition for all (A, c). Because such tuples (A, c) are countable, by the same argument in the proof of Lemma 32, 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 34. ◀

Finally, we present the proof of Lemma 33.

Proof of Lemma 33. Fix a polynomial-time oracle machine $F^?$ arbitrarily, and we 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 $r(n) = \sum_{i=1}^{m_c-1} i \cdot 2^{2i}$ $(\leq 2^{O(m_c)} = \mathsf{poly}(n))$. 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, i.e., $f_z(x) = f(z, x)$) 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 $k = \left(\sum_{i=1}^{m-1} i2^{2i}\right) + m(2^m(z'_{\mathbb{N}} - 1) + x'_{\mathbb{N}} - 1)$;
    - answer $z_{k+1} \circ \ldots \circ z_{k+m}$;
- **else** answer 0 (in this 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 verified 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, the following problem is contained in PSPACE:

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

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^n.

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

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

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

```
1 set z_F to empty string;
2 for i = 1 to m_c − 1 do
3     foreach z ∈ {0,1}^i do
4         foreach x =∈ {0,1}^i do
5             z_F := z_F ∘ F_z(x);
6 execute x ← I^TQBF(z_F, y) and return x;
```

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 $z \in \{0,1\}^{\leq m_c - 1}$.

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

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

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

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

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 $y \in \{0,1\}^{p(n)}$, $x_{\mathcal{F},y} := A^{\mathcal{O}_{\mathcal{F}}}(y)$ is uniquely determined and contained in $\{0,1\}^n$. Thus, the above (8) implies 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. $F^{\mathcal{O}_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x))) = f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x)))$; and
c. $f_{z_{\mathcal{F}}}(A^{\mathcal{O}_{\mathcal{F}}}(f_{z_{\mathcal{F}}}(x))) = f_{z_{\mathcal{F}}}(x)$,

then it is easily verified 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,

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

Thus, we provide the upper bound on the three probabilities in the right-hand side.

For the third probability, by the equation (7), we have that 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 probabilities, we will apply Lemma 34. By the expressions (8) and (9), we have that

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

and

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

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 34, 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 33 by ignoring the measure 0 of oracles not satisfying the condition (10) for all $F^?$ and applying the same argument in the proof of Lemma 32. ◀

Acknowledgments

The author thanks Toshiya Itoh and the anonymous reviewers for many helpful comments.

References

1 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.

2 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.

3 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.

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

5 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.

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

7 S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. *Journal of Computer and System Sciences*, 44(2):193–219, 1992. doi:https://doi.org/10.1016/0022-0000(92)90019-F.

8 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.

9 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.

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

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

12 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.

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

14 O. Goldreich and L. Levin. A Hard-Core Predicate for All One-Way Functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 25—-32, New York, NY, USA, 1989. ACM.

15 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.

16 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.

17 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.

18 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.

19 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.

20 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.

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

22 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.

23 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.

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

25 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.

26 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*, FOCS'90, pages 812–821, 1990.

27 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.

28 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.
29 R. Impagliazzo and A. Wigderson. Randomness vs Time: Derandomization under a Uniform Assumption. *Journal of Computer and System Sciences*, 63(4):672 – 688, 2001.
30 M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.
31 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.
32 M. Nanashima. Extending Learnability to Auxiliary-Input Cryptographic Primitives and Meta-PAC Learning. In *Proceedings of the 33rd Conference on Learning Theory, COLT'20*, volume 125, pages 2998–3029. PMLR, 09–12 Jul 2020.
33 R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems*, ISTCS'93, pages 3–17, June 1993.
34 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.
35 J. Rompel. One-way Functions Are Necessary and Sufficient for Secure Signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, 1990.
36 S. Rudich. *Limits on the Provable Consequences of One-way Functions*. PhD thesis, EECS Department, University of California, Berkeley, Dec 1988.
37 R. Santhanam. Pseudorandomness and the Minimum Circuit Size Problem. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*, volume 151 of *LIPIcs*, pages 68:1–68:26, 2020.
38 D. Xiao. *New Perspectives on the Complexity of Computational Learning, and Other Problems in Theoretical Computer Science*. PhD thesis, Princeton University, 2009.
39 D. Xiao. On basing ZK $\neq$ BPP on the hardness of PAC learning. In *Proceedings of the 24th Conference on Computational Complexity*, CCC'09, pages 304–315, 2009.

A Universal Extrapolation

A.1 Basic Tools

We introduce additional basic tools to show Lemma 15.

▶ **Fact 3** (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 $y_1, y_2 \in \{0,1\}^n$,*

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

▶ **Lemma 35.** *For every events $E_1, \dots, E_n$,*

$$\Pr\left[\text{exactly } 1 \text{ or } 2 \text{ events in } E_1, \dots, E_n \text{ occur}\right] \geq \sum_{i=1}^{n} \Pr[E_i] - \frac{1}{2} \sum_{i \neq j} \Pr[E_i \cap E_j].$$

Proof. For each index set $I \subseteq [n]$, we introduce the following notations:

$$E_I^{\cap} := \bigcap_{i \in I} E_i \quad \text{and} \quad E_I := \left(\bigcap_{i \in I} E_i\right) \cap \left(\bigcap_{i \in [n] \setminus I} \neg E_i\right).$$

By the inclusion-exclusion principle, we can show the lemma as follows:

$$
\begin{aligned}
&\Pr\left[\text{exactly 1 or 2 events in } E_1,\ldots,E_n \text{ occur}\right]\\
&\quad= \Pr\left[E_1\cup\cdots\cup E_n\right] - \sum_{I:3\le|I|\le n}\Pr[E_I]\\
&\quad= \sum_{i=1}^{n}\Pr[E_i] - \frac{1}{2}\sum_{i\neq j}\Pr[E_i\cap E_j] + \sum_{I:3\le|I|\le n}(-1)^{|I|-1}\Pr[E_I^{\cap}] - \sum_{I:3\le|I|\le n}\Pr[E_I]\\
&\quad= \sum_{i=1}^{n}\Pr[E_i] - \frac{1}{2}\sum_{i\neq j}\Pr[E_i\cap E_j] + \sum_{I:3\le|I|\le n}(-1)^{|I|-1}\sum_{J:I\subseteq J\subseteq[n]}\Pr[E_J] - \sum_{I:3\le|I|\le n}\Pr[E_I]\\
&\quad= \sum_{i=1}^{n}\Pr[E_i] - \frac{1}{2}\sum_{i\neq j}\Pr[E_i\cap E_j] + \sum_{I:3\le|I|\le n}\left(\sum_{\substack{S:S\subseteq I\\|S|\ge 3}}(-1)^{|S|-1} - 1\right)\Pr[E_I]\\
&\quad= \sum_{i=1}^{n}\Pr[E_i] - \frac{1}{2}\sum_{i\neq j}\Pr[E_i\cap E_j] + \sum_{I:3\le|I|\le n}\left(\sum_{i=3}^{|I|}(-1)^{i-1}\binom{|I|}{i} - 1\right)\Pr[E_I]\\
&\quad= \sum_{i=1}^{n}\Pr[E_i] - \frac{1}{2}\sum_{i\neq j}\Pr[E_i\cap E_j] + \sum_{I:3\le|I|\le n}\frac{|I|(|I|-3)}{2}\Pr[E_I]\\
&\quad\ge \sum_{i=1}^{n}\Pr[E_i] - \frac{1}{2}\sum_{i\neq j}\Pr[E_i\cap E_j].
\end{aligned}
$$

◀

A.2 Proof of Lemma 15

For $a,b\in\{0,1\}^n$, let $h_{a,b}:\{0,1\}^n\to\{0,1\}^n$ denote a universal hash function as in Fact 3.

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

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

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 $e(s(n))\le a$ and compute f_z as above, we can construct a general auxiliary-input function f.

For the sake of 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 $\delta'(n)=\frac{1}{2n}\cdot(1-2^{-\frac{\epsilon}{4}})^2\cdot\frac{\delta(n)}{12(n+1)}$. We construct the universal extrapolation algorithm $Ext_{s(n)}$ which accesses a (possibly randomized) $(1-\delta')$-inverting oracle for f_z nonadaptively as Algorithm 7. This immediately implies that if there exists no auxiliary-input one-way 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 $\epsilon' := \frac{1}{16}(2^{-\epsilon/2} - 2^{-\epsilon})$ 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 $c[\tilde{i}] \geq \frac{M}{16}(2^{-\epsilon/2} + 2^{-\epsilon})$;
return $2^{-\tilde{i}}$;

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

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

For any $i \in \{0, \ldots, n\}$, we have that

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

otherwise the equation (11) does not hold because each $i \in \{0, \ldots, n\}$ is selected with probability $2^{-\lceil \log n \rceil} \geq 1/2n$.

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

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

It is easily verified that $2^{n-t_y^*} = p_y \cdot 2^n = |X_y|$. For the sake of 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 36. For any $i < t^* - (3 + \epsilon)$,

$$\Pr\left[c[i] \geq \frac{M}{16}(2^{-\epsilon/2} + 2^{-\epsilon})\right] \leq \frac{\delta(n)}{3n},$$

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

▷ Claim 37. 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 for any input $y \in G_C$,

$$\Pr\left[c[t] \leq \frac{M}{16}(2^{-\epsilon/2} + 2^{-\epsilon})\right] \leq \frac{\delta(n)}{3},$$

where the latter probability depends on the choice of $a_{i,j}$,$b_{i,j}$, 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 36 and the union bound, $\tilde{i} \geq t^* - (3+\epsilon)$ with probability at least $1 - n \cdot \delta(n)/3n = 1 - \delta(n)/3$. By Claim 37, $\tilde{i} \leq t$ $(\leq t^*)$ with probability at least $1 - \delta(n)/3$. Thus, with probability at least $1 - \delta(n)$, $t^* - (3+\epsilon) \leq \tilde{i} \leq t^*$ holds. This implies Lemma 15 because $Ext_{s(n)}$ outputs $2^{-\tilde{i}}$ and

$$p = 2^{-t^*} \leq 2^{-\tilde{i}} \leq 2^{-t^*+(3+\epsilon)} = 2^{(3+\epsilon)} \cdot p.$$

Therefore, the remaining part is to show Claims 36 and 37.

Proof of Claim 36. We assume that $i < t^* - (3+\epsilon)$. Fix $j \in [M]$ arbitrarily. For the sake of 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, $h_{a,b}(r) \in X$), 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

$$\Pr_{a,b}[\exists\, r \in \{0,1\}^i \text{ such that } h_{a,b}(r) \in X] \leq \frac{1}{8} \cdot \frac{1}{2^\epsilon}. \tag{13}$$

This is because if we define Bernoulli random variables E_j as

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

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

$$\mu := \mathrm{E}[E_j] = \Pr_{a,b,\mathcal{I}}[E_j] \leq \Pr_{a,b}[\exists\, r \in \{0,1\}^i \text{ such that } h_{a,b}(r) \in X] \leq \frac{1}{8} \cdot \frac{1}{2^\epsilon},$$

and by the Hoeffding inequality,

$$\begin{aligned}
\Pr\left[c[i] \geq \frac{M}{16}(2^{-\epsilon/2} + 2^{-\epsilon})\right] &= \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j \geq \frac{1}{8} \cdot \frac{1}{2^\epsilon} + \epsilon'\right] \\
&\leq \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j - \mu \geq \epsilon'\right] \\
&\leq \exp\left(-2M\epsilon'^2\right) \\
&\leq \exp\left(-2\frac{\ln 3n - \ln \delta(n)}{2\epsilon'^2}\epsilon'^2\right) = \frac{\delta(n)}{3n}.
\end{aligned}$$

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

$$\begin{aligned}\Pr_{a,b}[\exists\, r \in \{0,1\}^i \text{ such that } h_{a,b}(r) \in X] &= \Pr_{a,b}[\exists\, (r,x) \in \{0,1\}^i \times X \text{ such that } h_{a,b}(r) = x] \\ &\le \sum_{(r,x)} \Pr_{a,b}[h_{a,b}(r) = x] \\ &= 2^i \cdot |X| \cdot 2^{-n} \\ &\le 2^{t^*-(3+\epsilon)} \cdot 2^{n-t^*} \cdot 2^{-n} = 2^{-(3+\epsilon)}\end{aligned}$$

◁

Proof of Claim 37. For $i = t$, we use the same notation E_j and μ as in the proof of Claim 36. 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 for any $y \in C(\{0,1\}^n) \setminus B_C$,

$$\Pr[E_j] \ge \frac{1}{8} \cdot \frac{1}{2^{\epsilon/2}}, \tag{14}$$

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

$$\begin{aligned}\Pr\left[c[t] \le \frac{M}{16}(2^{-\epsilon/2} + 2^{-\epsilon})\right] &= \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j \ge \frac{1}{8} \cdot \frac{1}{2^{\epsilon/2}} - \epsilon'\right] \\ &\le \Pr\left[\frac{1}{M}\sum_{j=1}^{M} E_j - \mu \le -\epsilon'\right] \\ &\le \exp\left(-2M\epsilon'^2\right) \quad (\because \text{ Hoeffding inequality}) \\ &\le \exp\left(-2\frac{\ln 3n - \ln\delta(n)}{2\epsilon'^2}\epsilon'^2\right) = \frac{\delta(n)}{3n} \le \frac{\delta(n)}{3}.\end{aligned}$$

Therefore, Claim 37 holds by letting $G_C := C(\{0,1\}^n) \setminus B_C$.

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

$$\Pr_{a,b}[\text{there exist exactly 1 or 2 elements } r \in \{0,1\}^t \text{ such that } h_{a,b}(r) \in X] \ge 1/4. \tag{15}$$

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

$$\begin{aligned}&\Pr_{a,b}[\text{there exist exactly 1 or 2 elements } r \in \{0,1\}^t \text{ such that } h_{a,b}(r) \in X] \\ &\qquad = \Pr\left[\text{exactly 1 or 2 events in } \{E_r : r \in R\} \text{ occur}\right].\end{aligned}$$

By Lemma 35, we have that

$$\Pr\left[\text{exactly 1 or 2 events in } \{E_r : r \in R\} \text{ occur}\right] \ge \sum_{r \in R} \Pr[E_r] - \frac{1}{2}\sum_{\substack{r \ne r' \\ r,r' \in R}} \Pr[E_r \cap E_{r'}].$$

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

For each $r \in R$,

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

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

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

Therefore, we have that

$$\Pr[\text{exactly 1 or 2 events in } \{E_r : r \in R\} \text{ occur}]$$
$$\geq \sum_{r \in R} \Pr[E_r] - \frac{1}{2} \sum_{\substack{r \neq r' \\ r,r' \in R}} \Pr[E_r \cap E_{r'}]$$
$$\geq |R| \cdot 2^{-t^*} - \frac{1}{2} \cdot |R|(|R|-1) \cdot 2^{-2t^*}$$
$$= 2^t \cdot 2^{-t^*} - \frac{1}{2} \cdot 2^t(2^t - 1) \cdot 2^{-2t^*}. \tag{16}$$

Let $\Delta := t^* - t$, then we have that $0 \leq \Delta < 1$. The right-hand side of (16) is bounded below as follows:

$$(\text{RHS of (16)}) = 2^t \cdot 2^{-\Delta - t} - \frac{1}{2} \cdot 2^t(2^t - 1) \cdot 2^{-2\Delta - 2t}$$
$$\geq 2^{-\Delta} - \frac{2^{-2\Delta}}{2} = \frac{1}{2} \cdot 2^{-\Delta}(2 - 2^{-\Delta}) > \frac{1}{2} \cdot 2^{-1}(2 - 2^{-1}) = \frac{3}{8} > \frac{1}{4}.$$

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

$$V = \{(a, b, y) : \text{there exist exactly 1 or 2 elements } r \in \{0,1\}^{t_y} \text{ such that } C(h_{a,b}(r)) = y\},$$

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

$$V_y = \{(a, b, y) \in V\}$$
$$F_y = \left\{(a, b, y) : \Pr_{\mathcal{I}}\left[\mathcal{I}(C, a \circ b \circ y \circ 0^{s(n)-|y|} \circ t_y) \text{ fails in inverting}\right] \geq 1 - 2^{-\frac{\epsilon}{4}}\right\}.$$

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

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

First, we show that $C(x) \in B_C$ holds with probability at most $\delta(n)/3$ over the choice of $x \in \{0,1\}^n$.

For each $i \in \{0, \ldots, n\}$, let $Y_i = \{y \in C(\{0,1\}^n) : t_y = i\}$. By the upper bound (12) on the failure probability of $\mathcal{I}$,

$$\begin{aligned}(1 - 2^{-\frac{\epsilon}{4}})^2 \cdot \frac{\delta(n)}{12} &\geq \sum_{i=0}^{n} \Pr_{a,b,r,\mathcal{I}}\left[y = C(h_{a,b}(r_{[i]}));\ \mathcal{I}(C, a \circ b \circ y \circ 0^{s(n)-|y|} \circ i) \text{ fails in inverting}\right] \\ &\geq \sum_{i=0}^{n} \sum_{y \in Y_i} \Pr_{a,b,r}\left[(a, b, C(h_{a,b}(r_{[t_y]})) \in F_y\right] \cdot \\ &\qquad \Pr_{a,b,r,\mathcal{I}}\left[\mathcal{I}(C, a \circ b \circ y \circ 0^{s(n)-|y|} \circ t_y) \text{ fails in inverting}\middle|(a, b, C(h_{a,b}(r_{[t_y]}))) \in F_y\right] \\ &\geq (1 - 2^{-\frac{\epsilon}{4}}) \cdot \sum_{i=0}^{n} \sum_{y \in Y_i} \Pr_{a,b,r}\left[(a, b, C(h_{a,b}(r_{[t_y]}))) \in F_y\right]\end{aligned}$$

Therefore, we have that

$$\sum_{y \in C(\{0,1\}^n)} \Pr_{a,b,r \sim \{0,1\}^n}\left[(a, b, C(h_{a,b}(r_{[t_y]}))) \in F_y\right] \leq (1 - 2^{-\frac{\epsilon}{4}}) \cdot \frac{\delta(n)}{12}.$$

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

$$\Pr_{a,b,r \sim \{0,1\}^n}[(a, b, C(h_{a,b}(r_{[t_y]}))) \in V_y] \geq \frac{1}{4} \cdot \frac{1}{2^{t_y}}.$$

Since $t_y^* \geq t_y$, we have that for any $y \in C(\{0,1\}^n)$,

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

Thus, we can show the upper bound as follows:

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

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

For any $y \in C(\{0,1\}^n) \setminus B_C$, we have that

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

Therefore,

$$\Pr_{a,b,r}\left[(a,b,C(h_{a,b}(r_{[t_y]}))) \in V_y \setminus F_y \;\middle|\; (a,b,C(h_{a,b}(r_{[t_y]}))) \in V_y\right] \geq 2^{-\frac{\epsilon}{4}}.$$

For each $a, b \in \{0,1\}^n$, there exist exactly 1 or 2 elements $r \in \{0,1\}^{t_y}$ such that $(a,b,C(h_{a,b}(r))) \in V_y$ by the definitions of V and V_y. Thus, we have also that

$$\begin{aligned}&\Pr_{a,b}\left[(a,b,y) \in V_y \setminus F_y \mid (a,b,y) \in V_y\right] \\ &\qquad \geq \frac{1}{2}\Pr_{a,b,r}\left[(a,b,C(h_{a,b}(r_{[t_y]}))) \in V_y \setminus F_y \;\middle|\; (a,b,C(h_{a,b}(r_{[t_y]}))) \in V_y\right] \geq \frac{1}{2}\cdot\frac{1}{2^{\epsilon/4}}\end{aligned} \tag{17}$$

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

$$\begin{aligned}\Pr[E_j] &= \Pr_{a,b,\mathcal{I}}[\mathcal{I}(C, a\circ b\circ y\circ t_y) \text{ succeeds in inverting}] \\ &\geq \Pr_{a,b,\mathcal{I}}[\mathcal{I}(C, a\circ b\circ y\circ t_y) \text{ succeeds in inverting} \wedge (a,b,y) \in V_y \setminus F_y] \\ &= \Pr_{a,b}[(a,b,y) \in V_y]\cdot\Pr_{a,b}[(a,b,y) \in V_y \setminus F_y | (a,b,y) \in V_y] \\ &\qquad\qquad \cdot \Pr_{a,b,\mathcal{I}}[\mathcal{I}(C, a\circ b\circ y\circ t_y) \text{ succeeds in inverting} | (a,b,y) \in V_y \setminus F_y] \\ &\geq \Pr_{a,b}[(a,b,y) \in V_y]\cdot\Pr_{a,b}[(a,b,y) \in V_y \setminus F_y | (a,b,y) \in V_y]\cdot\frac{1}{2^{\epsilon/4}} \quad (\because (a,b,y) \notin F_y) \\ &\geq \Pr_{a,b}[(a,b,y) \in V_y]\cdot\frac{1}{2}\cdot\frac{1}{2^{\epsilon/4}}\cdot\frac{1}{2^{\epsilon/4}} \quad (\because\ (17)) \\ &\geq \frac{1}{4}\cdot\frac{1}{2}\cdot\frac{1}{2^{\epsilon/4}}\cdot\frac{1}{2^{\epsilon/4}} = \frac{1}{8}\cdot\frac{1}{2^{\epsilon/2}} \quad (\because\ (15))\end{aligned}$$

◁

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