Electronic Colloquium on Computational Complexity, Report No. 91 (2026)

# Non-Levin NP-Hardness of Implicit MCSP and PAC Learning under Few Assumptions

Halley Goldberg
Simon Fraser University &
University of Warwick
halley_goldberg@sfu.ca

Mandar Juvekar
Boston University
mandarj@bu.edu

Valentine Kabanets
Simon Fraser University
kabanets@sfu.ca

June 4, 2026

**Abstract**

We show that several meta-complexity problems are NP-hard under randomized polynomial-time (half-Levin) reductions, and provably cannot be NP-hard under randomized Levin reductions, under the assumptions that

- (CRYPTOGRAPHY) there exists a subexponentially-secure indistinguishability obfuscator in the sense of Barak et al. (JACM 2012), and
- (PROOF COMPLEXITY) there are no infinitely-often subexponentially-optimal propositional proof systems in the sense of Cook and Reckhow (J. Symb. Log. 1979) and Krajíček and Pudlák (J. Symb. Log. 1989).

In particular, this is shown for

1. the problem of improper full-support PAC learning of boolean functions in $\mathsf{P/poly}$, and
2. a gap-version of the Implicit Minimum Circuit Size Problem (ImpMCSP).

More precisely, for item (1), we consider the following learning problem Total-Learn: Given a circuit sampling a distribution $\mathcal{E}$ of labeled examples $(x, b) \in \{0,1\}^n \times \{0,1\}$ so that *every* $x \in \{0,1\}^n$ *is in the support of* $\mathcal{E}$, distinguish between the two cases: (a) there exists a circuit $C$ of size at most $s$ that, for all $(x, b)$ in the support of $\mathcal{E}$, $C(x) = b$, and (b) no circuit $C$ of size $\mathsf{subexp}(s)$ satisfies $C(x) = b$ with probability noticeably higher than $1/2$ over samples $(x, b)$ from $\mathcal{E}$.

Gap-ImpMCSP in item (2) is defined as follows. Given a circuit $C$ sampling a distribution $\mathcal{E}$ of labeled examples $(x, f(x)) \in \{0,1\}^n \times \{0,1\}$ for some boolean function $f$ so that every $x \in \{0,1\}^n$ is in the support of $\mathcal{E}$, distinguish between the two cases: (a) the circuit complexity of $f$ is at most $s$, or (b) no circuit of size $\mathsf{subexp}(s)$ can approximate $f$ over $\mathcal{E}$ with probability noticeably higher than $1/2$.

We also give more examples of synergy between these two assumptions from cryptography and proof complexity. In particular, we prove that together they imply $\mathsf{NP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$, and hence that subexponentially-secure one-way functions and public-key encryption exist.

Our conditional NP-hardness results complement the recent results of Hirahara and Ilango (FOCS 2025) which prove conditional NP-hardness of constant-gap MCSP under quasipolynomial-time non-Levin reductions, from seemingly much stronger assumptions.



ISSN 1433-8092

# Contents

# 1 Introduction

A major open problem in complexity theory is to determine whether the Minimum Circuit Size Problem (MCSP) is NP-hard. Recall that an instance of MCSP consists of a truth table of a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ and a size parameter $0 < s < 2^n$, and one is asked to decide if $f$ can be computed by a boolean circuit of size at most $s$. MCSP is easily seen to be in NP. It must be outside of P, if one believes in the existence of cryptographic one-way functions [KC00]. But showing that MCSP is NP-hard has defied all attempts since at least the 1970s.

A recent paper by Hirahara and Ilango [HI25] makes some exciting progress towards the proof that MCSP is NP-hard. Their main result is that (a certain gap version of) MCSP is NP-hard under quasipolynomial-time non-adaptive Turing reductions, if one makes a number of very strong (albeit plausible) assumptions from complexity theory and cryptography. There are a few natural open questions left by [HI25]: Can one reduce the number of assumptions (or eliminate them completely)? Can one improve the running time of the NP-hardness reduction to be polynomial? Can one make the reduction many-one, which is what essentially all known NP-hardness reductions for other natural NP-complete problems are?

A very interesting feature of the conditional NP-hardness reduction given in [HI25] is that it avoids a barrier for such reductions due to Mazor and Pass [MP24]. It is an empirical fact that essentially all known NP-hardness reductions for natural NP-complete problems $L$ are in fact so-called "Levin reductions": efficient many-one reductions that map every yes- or no-instance $\varphi$ of SAT to a yes- or no-instance $R(\varphi)$ of $L$ respectively, which have the additional property that one can (a) efficiently map a satisfying assignment of a yes-instance $\varphi \in \mathsf{SAT}$ to a witness for $R(\varphi) \in L$, and (b) efficiently map a witness for $R(\varphi) \in L$ back to some satisfying assignment for $\varphi$.

In [MP24], it is shown that no NP-hardness reduction for a gap version of MCSP can be a Levin reduction, if one assumes that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ and that Indistinguishability Obfuscation ($i\mathcal{O}$) [Bar+01; Bar+12] is possible; both of these assumptions are generally accepted as plausible, with [JLS26] providing strong evidence for the existence of $i\mathcal{O}$. The reduction given in [HI25] is indeed non-Levin, thanks to the use of certain cryptographic constructions within the reduction!

**Implicit MCSP and Learning.** In this paper, we introduce and study an *implicit* variant of MCSP, denoted ImpMCSP. An instance of ImpMCSP is a pair $(\mathcal{E}, s)$, where $0 < s < 2^n$ and $\mathcal{E}\colon \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^{n+1}$ is a circuit that, for some boolean function $f : \{0,1\}^n \to \{0,1\}$, outputs pairs $(x, f(x))$ and furthermore is such that, for every $x \in \{0,1\}^n$, there is some input $r \in \{0,1\}^{\mathsf{poly}(n)}$ such that $\mathcal{E}(r) = (x, f(x))$; that is, information-theoretically, $\mathcal{E}$ uniquely determines $f$. One needs to decide if the function $f$ defined by $\mathcal{E}$ is computable by a circuit of size $s$. In other words, in ImpMCSP, one is given a concise description of the truth table of some boolean function $f$: as one enumerates all inputs $r$, the circuit $\mathcal{E}(r)$ will output a list of $(x, f(x))$ pairs in a "scrambled" order of inputs $x$ (rather than the canonical lexicographic order over $x$), possibly with repetitions. This makes ImpMCSP different from the variant of *succinct* MCSP considered in the literature (see, e.g., [AHK17]), where a concise description of the truth table of $f\colon \{0,1\}^n \to \{0,1\}$ is given by a circuit that outputs the pairs $(x, f(x))$ in the natural lexicographic order of $x$.

The problem ImpMCSP is closely related to certain learning problems which we discuss next. In the Computational Gap Learning (CGL) problem [ABX08], one is given a circuit $\mathcal{E}\colon \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^{n+1}$ (viewed as a sampler for a distribution of labeled examples $(x, b)$, for $x \in \{0,1\}^n$ and $b \in \{0,1\}$) and a parameter $s$, and one needs to distinguish between the following two cases: (YES-

CASE) there is a boolean circuit $C$ of size at most $s$ that agrees with all labeled examples coming from $\mathcal{E}$ (i.e., $C(x) = b$ for all pairs $(x, b)$ in the image of $\mathcal{E}$), and (NO-CASE) no boolean circuit of size at most $s'$ (which could even be larger than $s$) agrees with the labels $b$ for significantly more than the $1/2$ fraction of all pairs $(x, b)$ sampled by $\mathcal{E}$.

There are two important differences between ImpMCSP and CGL. First, in an instance $\mathcal{E}$ of CGL, there is no requirement that for every $x \in \{0,1\}^n$ there is a random string $r$ such that $\mathcal{E}(r) = (x, b)$ for some $b \in \{0,1\}$; that is, the distribution over pairs $(x, b)$ defined by $\mathcal{E}$ need not have full support over its first component $x$. Secondly, a no-instance $\mathcal{E}$ of CGL need not define a boolean *function*: there may be some random strings $r \neq r'$ such that $\mathcal{E}(r) = (x, 0)$ and $\mathcal{E}(r') = (x, 1)$ for some $x \in \{0,1\}^n$ (i.e., the labels $b$ need not be consistent for the same input $x$).

As an intermediate between CGL and ImpMCSP, we introduce the problem Total-Learn, which is exactly the same as CGL, but with the additional requirement that an instance $\mathcal{E}$ defines a distribution over $(x, b)$ with *full support* over $x$'s.

We generalize ImpMCSP, CGL, and Total-Learn to the corresponding *gap* versions, where the no-instances require circuit complexity at least $s'$, where $s'$ is noticeably larger than the circuit complexity $s$ of yes-instances. For a given function $g$, a gap-$g$ version of any of these three problems has $s' = g(s)$.

Our main result is that the subexponential-gap version of ImpMCSP is NP-hard under randomized polynomial-time many-one reductions, if one makes just two plausible and well-studied assumptions:

1. (CRYPTO) there exists $i\mathcal{O}$, and
2. (PROOF COMPLEXITY) there is no optimal propositional proof system (as defined by [CR79; KP89]);

both assumptions are taken in the subexponential regime of parameters, to be explained below.

We also argue that our many-one reduction is provably *non-Levin*, under the same two assumptions. Specifically, we show that the approach of [MP24] can be used to argue that no randomized (even subexponential-time) Levin reductions exist for proving that the polynomial-gap ImpMCSP is NP-hard.

Under the same assumptions, we also prove an analogous result for the learning problem Total-Learn: its subexponential-gap version is NP-hard under randomized polynomial-time many-one reductions, and that even its polynomial-gap version cannot be NP-hard under randomized subexponential-time Levin reductions. The conditional NP-hardness of the subexponential-gap version of Total-Learn means that improper (i.e., with arbitrary polynomial gap) full-support PAC learning of the class P/poly is NP-hard, under the same two assumptions as above. By the *full-support* PAC learning, we mean the special case of PAC learning where a given input distribution $\mathcal{E}$ of labeled examples $(x, f(x))$, for some boolean function function $f\colon \{0,1\}^n \to \{0,1\}$, must have full support over its first component, i.e., for every $x \in \{0,1\}^n$, the support of $\mathcal{E}$ contains the pair $(x, f(x))$. Our result strengthens the conditional NP-hardness of improper PAC learning of [GK25].

### 1.1 Our Results

We first discuss our assumptions.

**Indistinguishability Obfuscation.** An Indistinguishability Obfuscator ($i\mathcal{O}$) [Bar+01; Bar+12] is an efficient randomized algorithm for "obfuscating" a given boolean circuit $C$ into a functionally equivalent circuit $\tilde{C}$ so that for any two same-size, syntactically different but functionally equivalent boolean circuits $C_1$ and $C_2$, the distributions $i\mathcal{O}(C_1)$ and $i\mathcal{O}(C_2)$ are computationally indistinguishable. The computation power of adversaries that are fooled by $i\mathcal{O}$ defines the security of $i\mathcal{O}$. For example, if $i\mathcal{O}$ fools adversaries that are non-uniform boolean circuits of subexponential size, the $i\mathcal{O}$ is called subexponentially secure. There appears to be a strong consensus in the crypto community that such $i\mathcal{O}$ likely do exist, especially after the celebrated construction of $i\mathcal{O}$ from well-studied assumptions in [JLS26].

**Optimal proof systems.** The seminal work of Cook and Reckhow [CR79] introduced the general notion of a propositional proof system (PPS), as an efficient algorithm $\mathcal{L}$ that maps arbitrary inputs $\pi$ to propositional tautologies. For a given tautology $x \in \mathsf{TAUT}$, the size of a smallest string $\pi \in \{0,1\}^*$ such that $\mathcal{L}(\pi) = x$ is called the minimum $\mathcal{L}$-proof size of $x$. The main observation of [CR79] is that $\mathsf{NP} = \mathsf{coNP}$ if and only if there is a PPS $\mathcal{L}$ and a polynomial $p$ such that every $x \in \mathsf{TAUT}$ has the minimum $\mathcal{L}$-proof of size $p(|x|)$. Under the standard assumption that $\mathsf{NP} \neq \mathsf{coNP}$ (the "holy grail" of proof complexity), one immediately concludes that no such *polynomially bounded* PPS exists for $\mathsf{TAUT}$.

Cook and Reckhow [CR79] (see also [KP89]) defined a natural way to compare the power of different proof systems, via the notion of efficient simulation. Intuitively, a PPS $\mathcal{L}$ polynomially simulates a PPS $\mathcal{L}'$ if every tautology $x \in \mathsf{TAUT}$ provable in $\mathcal{L}'$ with a proof $\pi'$ also has a proof $\pi$ in $\mathcal{L}$ with $|\pi| \leq \mathsf{poly}(|\pi'|)$.

Then a polynomially *optimal PPS* is a PPS $\mathcal{L}$ that polynomially simulates every other PPS $\mathcal{L}'$ (with an actual polynomial proof-size overhead in the simulation dependent on the particular PPS $\mathcal{L}'$). It is easy to see that if $\mathsf{NP} = \mathsf{coNP}$ then by [CR79], $\mathsf{TAUT}$ has a polynomially-bounded PPS, and so a polynomially optimal PPS exists trivially.

Krajíček and Pudlák [KP89] conjectured that a polynomially optimal PPS for $\mathsf{TAUT}$ *does not* exist. As observed above, this conjecture implies that $\mathsf{NP} \neq \mathsf{coNP}$. In fact, it implies that $\mathsf{NE} \neq \mathsf{coNE}$ [KP89], and that $\mathsf{NEE} \neq \mathsf{coNEE}$ [KMT03]. We will assume the non-existence of even *subexponentially* (rather than polynomially) optimal PPS, and even for infinitely many input lengths (rather than almost everywhere). This is a natural strengthening of the classical conjecture by [KP89], also recently used in the work of Ilango [Ila25].

We can now state an informal version of our main result.

**Theorem 1** (Conditional $\mathsf{NP}$-hardness of gap $\mathsf{ImpMCSP}$, informal)**.** *Assume the following:*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then*

- *a subexponential-gap version of $\mathsf{ImpMCSP}$ is $\mathsf{NP}$-hard under randomized polynomial-time many-one reductions, but*
- *a polynomial-gap version of $\mathsf{ImpMCSP}$ cannot be $\mathsf{NP}$-hard under (subexponential-time) randomized Levin reductions.*

An analogous result holds also for the problem Total-Learn.

**Theorem 2** (Conditional NP-hardness of gap Total-Learn, informal)**.** *Assume the following:*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then*

- *a subexponential-gap version of* Total-Learn *is* NP*-hard under randomized polynomial-time many-one reductions, but*
- *a polynomial-gap version of* Total-Learn *cannot be* NP*-hard under (subexponential-time) randomized Levin reductions.*

The many-one NP-hardness reductions in the theorems above are actually half-Levin (as defined in [GK25]): these are many-one reductions $R$ from SAT to a given language $L$ such that one can also efficiently construct, for a given yes-instance $\varphi$ of SAT with a satisfying assignment $w$, an $L$-witness for $R(\varphi) \in L$, but there may be no efficient algorithm that extracts a witness for $\varphi \in \mathsf{SAT}$ from an $L$-witness $w'$ of $R(\varphi)$.

It is worth pointing out that the half-Levin reductions in Theorem 1 and Theorem 2 can be made *deterministic quasipolynomial time*. They can even be made *deterministic polynomial time*, albeit under an extra derandomization assumption that promise-BPP = promise-P.

We also remark that the existence of half-Levin reductions in Theorem 1 and Theorem 2 is *non-constructive* in the sense that we do not know how to exhibit an actual code for the Turing machine computing the corresponding reduction. Intuitively, this is due to the non-constructive nature of the "no optimal PPS" assumption, which says that for every given PPS $\mathcal{L}$ there must exist a "hard" PPS $\mathcal{L}'$, but we do not necessarily know how to build $\mathcal{L}'$ from $\mathcal{L}$.

**Implications for PAC learning.** The PAC learning model of Valiant [Val84] defines a learning task for P/poly as follows. A learning algorithm is given access to an oracle providing labeled examples $(x, f(x))$, for some boolean function family $f \in \mathsf{P/poly}$ (of some polynomial circuit complexity $s(n)$), where the examples are sampled from some unknown distribution $\mathcal{D}$ over inputs $x$. The learning algorithm must output, with high probability, a boolean circuit $C$ that computes $f$ correctly with high probability over inputs $x \in \{0,1\}^n$ sampled from $\mathcal{D}$. In the *proper* setting of PAC learning, the learning algorithm must output a circuit $C$ of size at most $s(n)$. In the more relaxed *improper* setting, the learning algorithm may output a circuit $C$ with size much larger than $s(n)$ (bounded by the runtime of the learning algorithm).

It is suspected that the task of improper PAC learning of boolean functions in P/poly is NP-hard. However, as in the case of MCSP, proving NP-hardness for improper PAC learning unconditionally is currently out of reach. Goldberg and Kabanets [GK25] obtained a conditional NP-hardness result for the task of improper PAC learning of P/poly, by showing that a polynomial-gap version of CGL is NP-hard, under cryptographic assumptions (the existence of $i\mathcal{O}$). Our Theorem 2 gives an NP-hardness result for Total-Learn, which is a special case of CGL, under cryptographic and proof complexity assumptions.

**The power of cryptography and proof complexity.** We also show the following examples of synergy between our two assumptions.

**Theorem 3** (Synergy between $i\mathcal{O}$ and "no optimal PPS")**.** *Assume*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then all of the following hold:*

1. $\mathsf{NP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$,
2. $\mathsf{promise\text{-}BPP} \subseteq \mathsf{promise\text{-}TIME}[n^{\mathsf{poly}(\log n)}]$,
3. *subexponentially-secure one-way functions exist,*
4. *public-key encryption exists, and*
5. *if* $\mathsf{promise\text{-}BPP} = \mathsf{promise\text{-}P}$ *then* $\mathsf{NP} \cap \mathsf{coNP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$.

Previously, Impagliazzo et al. [IKV23] showed that the existence of (polynomially secure) $i\mathcal{O}$ implies that $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$. In contrast, our Theorem 3 gets a much stronger, almost-everywhere, circuit lower bound for the much smaller complexity class NP, using an additional assumption from proof complexity.

**Limits of strong witness encryption.** Given the results above and prior related work, one may have the impression that Indistinguishability Obfuscation is somehow *working against itself*. For example, while $i\mathcal{O}$ yields NP-hardness of problems in meta-complexity (via the closely related notion of Witness Encryption; see [HIR25; GK25]), it also yields non-NP-hardness of such problems (see [MP24]). In light of this phenomenon, we prove that certain variants of Witness Encryption are actually *incompatible* with $i\mathcal{O}$, assuming only $\mathsf{NP} \not\subseteq \mathsf{BPP}$. In particular, this holds for a strong form of Witness Encryption in which security is "extractable" [Gol+13], and ciphertexts are "verifiable". If one believes in $i\mathcal{O}$ and in $\mathsf{NP} \not\subseteq \mathsf{BPP}$, then Witness Encryption cannot have these additional features. We refer the reader to Section 6 for the details.[1]

## 1.2 Comparison to Related Work

**MCSP vs. ImpMCSP.** Our Theorem 1 is similar to the aforementioned conditional result of [HI25] that the constant-factor-gap version of MCSP is NP-hard under deterministic quasipolynomial-time Turing reductions. The assumptions used by [HI25] are the following: (1) there exist subexponentially secure non-interactive witness indistinguishable proof systems (NIWIs) for SAT, (2) coNP requires subexponential-size nondeterministic circuits for almost all input lengths, and (3) $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ requires (deterministic) circuits of close to maximum size $\delta 2^n/n$, for some constant $\delta > 0$, for almost all input lengths.

We compare these assumptions of [HI25] with the two assumptions of our Theorem 1. Assumption (1) of [HI25] is in the spirit of our cryptographic assumption about the existence of

[1] We also mention a line of work in cryptography studying incompatibilities among various forms of obfuscation [GK05; Bit+14; Gar+17].

subexponentially secure $i\mathcal{O}$; in fact, as we will show, our two assumptions together ($i\mathcal{O}$ plus non-existence of optimal PPS) imply the existence of NIWI needed by [HI25]. Assumption (2) of [HI25] is related to our proof complexity assumption, and could be viewed as a statement about the power of "propositional proof systems with advice" (as defined by [CK07]), but without the use of the notion of an optimal PPS (which does not seem to have been defined for such non-uniform PPSs). Assumption (3) of [HI25] is a very strong circuit lower bound assumption for a relatively small complexity class $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$. While we do not make any similar assumption in our Theorem 1, we note that our two assumptions together do imply an almost-everywhere subexponential circuit lower bounds for NP (and even for $\mathsf{NP} \cap \mathsf{coNP}$ under an extra derandomization assumption), as stated in Theorem 3, items (1) and (5). However, assumption (3) of [HI25] seems to be much stronger than what can be derived from our two assumptions of Theorem 1.

The result of [HI25] is about $\mathsf{GapMCSP} \in \mathsf{NP}$, while our Theorem 1 is about a potentially harder problem $\mathsf{GapImpMCSP} \in \mathsf{promise\text{-}MA}$. We exploit the structure of ImpMCSP to get conditional NP-hardness for a large (subexponential) gap version of ImpMCSP, as opposed to the small factor gap for MCSP in [HI25]. We also manage to get NP-hardness under polynomial-time many-one (half-Levin) reductions, rather than for quasipolynomial-time Turing reductions.

**Learning.** The problem CGL was introduced by [ABX08] in the context of trying to show that Valiant's PAC learning [Val84] is NP-hard. It was shown by [GK25] that the NP-hardness of CGL is equivalent to the cryptographic assumption that Witness Encryption (WE) for NP is possible (note that WE is known to be implied by the existence of $i\mathcal{O}$ [Gar+16]).

A yes-instance of CGL is a distribution $\mathcal{E}$ over labeled examples $(x, f(x))$ for some *partial* boolean function $f$, defined on the inputs $x$ in the support of $\mathcal{E}$. In contrast, a yes-instance of Total-Learn is a distribution $\mathcal{E}$ over labeled examples $(x, f(x))$ for some *total* boolean function $f$. So Theorem 2 shows conditional NP-hardness, under half-Levin reductions, for a special case of CGL. The fact that instances of Total-Learn have full support also allows us to rule out Levin NP-hardness reductions for Total-Learn, under the same assumptions, using the techniques of [MP24]; it is not known how to rule out Levin NP-hardness reductions for CGL.

The result from [GK25] about the conditional NP-hardness of CGL from the WE assumption is a starting point for our proof of Theorem 2. As in [GK25], we also get equivalence between NP-hardness of Total-Learn and the existence of certain versions of Witness Encryption. Together with the conditional impossibility result for NP-hardness Levin reductions for Total-Learn mentioned above, these results can be used to rule out certain strong forms of Witness Encryption. See Section 6 for details.

**Combining cryptography and proof complexity.** A recent paper by Ilango [Ila25] (building upon [KZ20]) showed how to combine powerful cryptographic primitives (like NIWI) with a hardness assumption from proof complexity (the non-existence of optimal PPS) to get a powerful relaxation of the classical notion of zero-knowledge proofs, where the existence of a standard zero-knowledge simulator is relaxed to the hardness of proving (in a certain formal sense) that such a simulator does not exist. This relaxed notion of zero-knowledge allows one to get prover–verifier systems for languages in NP where the verifier has perfect soundness (i.e., rejects all incorrect statements), and certain logical consequences of the existence of a zero-knowledge simulator hold, even though a zero-knowledge simulator does *not* actually exist!

We use the insights from [Ila25] to prove our main results. For our cryptographic assumption,

instead of the existence of NIWIs, we use the existence of $i\mathcal{O}$. As mentioned above, we can show that $i\mathcal{O}$ plus the non-existence of optimal PPS imply the existence of NIWI (as well as derandomization of promise-BPP; see Theorem 3), which then allows us to use the constructions of [Ila25]. Rather than using the main theorem from [Ila25] as a "black box", we build upon the ideas from [Ila25] to obtain a simpler proof framework that is sufficient for our purposes.

### 1.3 Our Techniques

**Proof sketch of Theorem 3.** To illustrate how we use $i\mathcal{O}$ plus "no optimal PPS", we first sketch the proof of item (1) of Theorem 3, i.e., that these two assumptions imply $\mathsf{NP} \not\subset \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$.

First, we use the result of [GK25] that $i\mathcal{O}$ implies a deterministic polynomial-time many-one reduction from SAT to a subexponential-gap CGL. The reduction has the following property: For $\varphi \in \mathsf{SAT}$, the output of the reduction is a yes-instance $\mathcal{E}_\varphi \colon \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^n \times \{0,1\}$ of CGL such that the outputs $(x,b)$ of $\mathcal{E}_\varphi$ are consistent with a partial boolean function: there are no $r \neq r' \in \{0,1\}^{\mathsf{poly}(n)}$ such that $\mathcal{E}_\varphi(r) = (x,0)$ and $\mathcal{E}_\varphi(r') = (x,1)$ for some $x \in \{0,1\}^n$. This implies that if there exist $r \neq r'$ and $x$ such that $\mathcal{E}_\varphi(r) = (x,0)$ and $\mathcal{E}_\varphi(r') = (x,1)$, then we know that $\varphi$ must be unsatisfiable! Also note the size of such a proof of unsatisfiability of $\varphi$ is $|r| + |r'| \leq \mathsf{poly}(n)$, and it can be verified in some fixed deterministic time $\mathsf{poly}(n)$ by running the polynomial-time reduction to compute $\mathcal{E}_\varphi$, and then checking that $\mathcal{E}_\varphi(r) = (x,0)$ and $\mathcal{E}_\varphi(r') = (x,1)$, for some $x$.

Define the following PPS $\mathcal{L}$:

- an input $\pi$ is interpreted as $(\varphi, r, r')$, with $|r| = |r'| \leq \mathsf{poly}(|\varphi|)$,
- $\mathcal{E}_\varphi(r) = (x,0)$ and $\mathcal{E}_\varphi(r') = (x,1)$, for some $x$, then output $\varphi$; otherwise, output some trivial unsatisfiable formula.

The soundness of this proof system $\mathcal{L}$ follows from the correctness of the reduction from SAT to CGL, i.e., from the assumption that $i\mathcal{O}$ exists.

By our proof complexity assumption, we get that there must exist a PPS $\mathcal{L}'$ such that our PPS $\mathcal{L}$ cannot subexponentially simulate $\mathcal{L}'$, even for infinitely many input lengths. It was argued by [KP89] that for any such pair of $\mathcal{L}$ and $\mathcal{L}'$, there exists a uniform polynomial-time algorithm $Gen$ that, on input $1^n$, generates an unsatisfiable formula $\psi_n$ (of length $n$) so that $\mathcal{L}$ cannot prove the unsatisfiability of $\psi_n$, with subexponential-size proofs, for all but finitely many $n \geq 1$.

It follows that for almost all such $\psi_n$, the distribution $\mathcal{E}_{\psi_n}$ defines a valid partial boolean function $f$ over the inputs $x$ in the support of $\mathcal{E}_{\psi_n}$. Since each $\psi_n \notin \mathsf{SAT}$, we get by the correctness of the reduction that this partial boolean function $f$ must have subexponential circuit complexity $2^{n^\delta}$, for some $\delta > 0$, for almost all input lengths. We can make each such $f$ into a total function $h$ by assigning the value 0 to all inputs $x$ outside the support of $\mathcal{E}_{\psi_n}$.

Observe that this hard boolean function $h$ is computable in NP as follows:

> On input $x \in \{0,1\}^n$, compute the input length $m$ such that $\mathcal{E}_{\psi_m} \colon \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^n \times \{0,1\}$. Run $Gen(1^m)$ to compute $\psi_m$. Finally, nondeterministically guess a string $r \in \{0,1\}^{\mathsf{poly}(n)}$, and accept if $\mathcal{E}_{\psi_m}(r) = (x,1)$.

This concludes the proof sketch; for more details, see Lemma 29 below.[2]

[2]We note that our argument to construct a hard boolean function from an NP-hardness reduction is similar to

Items (2)–(4) of Theorem 3 follow by known results from the literature: hardness-randomness tradeoffs [NW94; Bab+93; IW97; Uma03] for (2), [Kom+22] for (3), and [SW14] for (4). Finally, item (5) is proved along the lines of item (1), but using Theorem 2 instead of [GK25] as a starting point; see Corollary 42 for the proof.

**Proof sketch of Theorem 2.** Next, we sketch the main ideas in the proof of Theorem 2. We begin by exhibiting a reduction from SAT to Total-Learn under assumptions of $i\mathcal{O}$ and no optimal PPS. To simplify the proof sketch, we make an additional assumption that promise-BPP $\subseteq$ promise-P, which will yield a deterministic reduction. Again, the starting point here is the reduction of [GK25] discussed above, mapping instances $\varphi$ of SAT to instances $\mathcal{E}_\varphi$ of CGL. Because we would like a reduction to Total-Learn, we want to modify the outputs $\mathcal{E}_\varphi$ to have a full domain: namely, for every $x'$ of a given length, there exists $b \in \{0,1\}$ such that $\mathcal{E}_\varphi$ outputs $(x', b)$ with non-zero probability. To that end, we will employ some tools from cryptography. By Theorem 3, under our assumptions of subexponentially secure $i\mathcal{O}$ and non-existence of subexponentially optimal PPS, we have a subexponentially secure one-way function. Combined with constructions of [SZ25b] and [BP15], this yields a subexponentially secure non-interactive witness-indistinguishable proof system (NIWI) for any language in NP. Moreover, promise-BPP $\subseteq$ promise-P implies that the NIWI verifier can be made deterministic.

Notice that the domain of the CGL instance $\mathcal{E}_\varphi$ corresponds to an NP language

$$L_{dom} = \{(\varphi, x) \mid \exists\, b, r \text{ such that } \mathcal{E}_\varphi(r) = (x, b)\}.$$

We will use a NIWI for $L_{dom}$ to augment $\mathcal{E}_\varphi$ to be total. Specifically, a subexponentially secure NIWI for $L_{dom}$ consists of a randomized prover $P$ and a deterministic verifier $V$ satisfying

- **(Perfect Completeness)** for any $(\varphi, x) \in L_{dom}$ with witness $(b, r)$,

$$\Pr_P[V(\varphi, x; P(\varphi, x; b, r)) = 1] = 1;$$

- **(Perfect Soundness)** for any $(\varphi, x) \notin L_{dom}$ and any string $\pi$, $V(\varphi, x; \pi) = 0$; and
- **(Witness Indistinguishability)** for any $(\varphi, x) \in L_{dom}$ with witnesses $(b, r)$ and $(b', r')$, $P(\varphi, x; b, r)$ is indistinguishable from $P(\varphi, x; b', r')$ by circuits of subexponential size.[3]

We then define a modified distribution $\widehat{\mathcal{E}}_{\varphi,(P,V)}$ as follows, letting $\alpha(n)$ be some "tiny" probability $1/2^{n^{\Omega(1)}}$.

1. With probability $1 - \alpha(n)$, sample $(x, b) \sim \mathcal{E}_\varphi$ with randomness $r$.
   (a) With probability $1 - \alpha(n)$, sample $\pi \sim P(\varphi, x; b, r)$ and output $(x, \pi; b)$.
   (b) With probability $\alpha(n)$, sample $\pi$ uniformly at random. If $V(\varphi, x; \pi) = 1$, output $(x, \pi; b)$. Otherwise, output $(x, \pi; 0)$.

the way Kabanets and Cai [KC00] constructed a hard boolean function from an assumed deterministic many-one (natural) reduction from SAT to MCSP. They used arbitrary (canonical) unsatisfiable formulas as inputs to the reduction, whereas we use P-uniform "hard" unsatisfiable formulas coming from the "no optimal PPS" assumption.

[3] Actually, from the assumptions of Theorem 2, we only get a NIWI with a randomized verifier. For simplicity, we assume here that it can be derandomized. This is true under the additional assumption that promise-BPP = promise-P.

2. With probability $\alpha(n)$, sample $(x, \pi)$ uniformly at random. If $V(\varphi, x; \pi) = 0$, output $(x, \pi; 0)$. Otherwise, take a sample from $\mathcal{E}_\varphi$ and augment using $P$ as in 1(a).

First observe that $\widehat{\mathcal{E}}_{\varphi,(P,V)}$ satisfies the promise of Total-Learn: for any pair $(x, \pi)$, there exists $b \in \{0, 1\}$ such that $(x, \pi; b)$ is in the support of $\widehat{\mathcal{E}}_{\varphi,(P,V)}$. This is easily verified by considering three cases: (1) $x \in L_{dom}$ and $\pi$ is a possible output of $P$, (2) $x \in L_{dom}$ and $\pi$ is not a possible output of $P$, and (3) $x \notin L_{dom}$. In all cases, there is a choice of randomness yielding the output $(x, \pi)$

Next, observe that the completeness of the reduction still holds. Indeed, recall that if $\varphi \in \mathsf{SAT}$, then $\mathcal{E}_\varphi$ is a yes-instance of CGL. That is, there is a circuit $C$ of size at most $\mathsf{poly}(n)$ such that for all $(x, b)$ in the support of $\mathcal{E}_\varphi$, $C(x) = b$. To see that $\widehat{\mathcal{E}}_{\varphi,(P,V)}$ is a yes-instance of Total-Learn, define a $\mathsf{poly}(n)$-size circuit $C'$ as follows.

$$C'(x, \pi) = \begin{cases} C(x) & \text{if } V(\varphi, x; \pi) = 1 \\ 0 & \text{otherwise} \end{cases}$$

It is easily verified that $C'$ agrees with all the labels output by $\widehat{\mathcal{E}}_{\varphi,(P,V)}$.

Before discussing soundness, we need to be more specific about the NIWI we use above. In particular, for some sequence of formulas $\Psi = \{\psi_\lambda\}_{\lambda \in \mathbb{N}}$, we will use a NIWI $(P[\Psi], V[\Psi])$ as defined in [Ila25], based on the "OR proof" construction of [FLS90]. Roughly, $(P[\Psi], V[\Psi])$ is a NIWI for the language

$$L' = \{z \mid z \in L_{dom} \text{ or } \psi_\lambda \in \mathsf{SAT}\},$$

where $\lambda = \mathsf{poly}(|z|)$. Interpreted as a NIWI for $L_{dom}$, it is easy to see that $(P[\Psi], V[\Psi])$ has perfect completeness. Moreover, if all the $\psi_\lambda$ are unsatisfiable, then it also has perfect soundness. Lastly, $(P[\Psi], V[\Psi])$ has the property that if some $\psi_\lambda$ is satisfiable, then $P[\Psi]$ has a non-uniform *zero-knowledge simulator*[4] at input length corresponding to $\lambda$. In particular, this follows from the witness-indistinguishability property of $(P[\Psi], V[\Psi])$: the simulator can simply be $P[\Psi]$ with a witness for $\psi_\lambda$ hard-wired (see Lemma 20 for more details). Taken in the contrapositive, the latter implies that any "short proof" that a simulator does not exist would also be a short proof of $\psi_\lambda \in \mathsf{UNSAT}$.

Suppose toward a contradiction that there exist $\varphi \in \{0, 1\}^n \setminus \mathsf{SAT}$ and a subexponential-size circuit $D$ such that

$$\Pr_{(x,\pi;b)\sim\widehat{\mathcal{E}}_{\varphi,(P,V)}}[D(x, \pi) = b] \geq \frac{1}{2} + \frac{1}{2^{n^{o(1)}}}.$$

The key idea is that, if a zero-knowledge simulator $S$ exists, then $D'(x) := D(x, S(\varphi, x))$ is a subexponential-size circuit such that

$$\Pr_{(x;b)\sim\mathcal{E}_\varphi}[D'(x) = b] \geq \frac{1}{2} + \frac{1}{2^{n^{o(1)}}}. \tag{1}$$

Indeed, the distribution $\widehat{\mathcal{E}}_{\varphi,(P,V)}$ is $O(\alpha(n))$-close to the distribution of case 1(a) in its construction. And, by the definition of a simulator, the distribution of case 1(a) is computationally indistinguishable from the distribution $(x, S(\varphi, x); b)$, where $(x, b) \sim \mathcal{E}_\varphi$.

[4]Roughly, a simulator on a yes-instance produces a distribution of strings that looks like the distribution of proofs coming from the actual prover, even though the simulator does not have a witness that the prover has.

However, if Equation (1) held, then it would contradict the correctness of our reduction to CGL on a no-instance of SAT: we know that for $\varphi \notin \mathsf{SAT}$, there is no subexponential-size circuit with inverse subexponential advantage on $\mathcal{E}_\varphi$. Overall, an unsatisfiable $\varphi$ together with a circuit $D$ as above (ie. with non-trivial success probability on $\widehat{\mathcal{E}}_{\varphi,(P,V)}$) would constitute a "short proof" that a simulator for $(P[\Psi], V[\Psi])$ does not exist. Together with the latter property of $(P[\Psi], V[\Psi])$ mentioned above, such a pair $(\varphi, D)$ would then constitute a short proof that the corresponding $\psi_\lambda$ is unsatisfiable in some appropriate PPS $\mathcal{L}$. This proof has size at most $2^{\mathsf{poly}(n)}$, including a transcript of a brute-force search showing that $\varphi \notin \mathsf{SAT}$.

We may now invoke our assumption that no subexponentially optimal PPS exists. As mentioned earlier, by the work of Krajícek and Pudlák [KP89], this is known to imply that for every PPS $\mathcal{L}$, for some constant $c \in \mathbb{N}$, there exists a P-uniform sequence $\Psi^* = \{\psi^*_\lambda\}_{\lambda\in\mathbb{N}}$ of unsatisfiable formulas such that for almost all $\lambda \in \mathbb{N}$, the statement "$\psi^*_\lambda$ is unsatisfiable" does not have an $\mathcal{L}$-proof of length less than $2^{\lambda^{1/c}}$. Our final reduction will be as described above, using the NIWI $(P[\Psi^*], V[\Psi^*])$ for $L_{dom}$. Setting $\lambda(n) \gg n^c$ a sufficiently large polynomial, the proof of length $2^{\mathsf{poly}(n)}$ given by $(\varphi, D)$ violating soundness would contradict the fact that $\psi^*_\lambda$ does not have short $\mathcal{L}$-proofs. We conclude that the reduction is sound.

For the second part of Theorem 2, which states that Total-Learn cannot be NP-hard under a subexponential-time Levin reduction, the proof essentially follows from techniques of Mazor and Pass [MP24]. In particular, they show that a "gap" version of the Minimum Circuit Size Problem is not NP-hard, assuming subexponentially-secure $i\mathcal{O}$ and one-way functions. A crucial idea in that proof is that *any* two witnesses $C, C'$ for a given yes-instance of GapMCSP are circuits computing exactly the same function. For this reason, these circuits satisfy the condition of $i\mathcal{O}$, so the obfuscations of $C$ and $C'$ are guaranteed to be computationally indistinguishable. Notice that the same is true of a yes-instance of Total-Learn: if $\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1$ and $\Pr_{(x,b)\sim\mathcal{E}}[C'(x) = b] = 1$, and if $\mathcal{E}$ has full support over its first component as promised by Total-Learn, then $C$ and $C'$ compute the same function. We also make use of the fact that, by Theorem 3, we have subexponentially-secure one-way functions. We refer the reader to Section 4 for more details.

**Proof sketch of Theorem 1.** Proving the first item of Theorem 1, for the case of ImpMCSP, proceeds in two stages. The first stage is to obtain a reduction from SAT to a version of the problem for *partial* functions, denoted ImpMCSP$^*$. Here, it is required that $\mathcal{E}$ defines a (partial) boolean function $f_\mathcal{E}$: namely, there is no string $x$ such that both $(x, 0)$ and $(x, 1)$ are in the support of $\mathcal{E}$, regardless of whether $\mathcal{E}$ is a yes-instance or no-instance of the problem. Unlike the case of (total) ImpMCSP, $f_\mathcal{E}$ is not required to have a full domain.

The proof builds on a number of ideas from the sketches given above. In particular, let $R$ be a reduction mapping SAT-instances $\varphi$ to CGL-instances $\mathcal{E}_\varphi$, as given by [GK25]. Our task at this stage is to ensure that $\mathcal{E}_\varphi$ is consistent with a function, even in the "no" case. As in the sketch of Theorem 3, we observe that if there existed $x$ and $r \neq r'$ such that $\mathcal{E}_\varphi(r) = (x, 0)$ and $\mathcal{E}_\varphi(r) = (x, 1)$, then $(x, r, r')$ would constitute a short proof that $\varphi$ is unsatisfiable.

The next idea is to apply the reduction $R$ to a disjunction $\varphi \vee \psi_{\lambda(|\varphi|)}$ for some fixed sequence $\Psi$ of unsatisfiable formulas $\psi_\lambda$. If $\varphi$ is satisfiable, then the disjunction is satisfiable, and correctness of the reductions holds. On the other hand, since $\psi_\lambda$ is unsatisfiable, the soundness of the reduction still holds with respect to $\varphi$. Crucially, $(x, r, r')$ as above would now also constitute a short proof that $\psi_\lambda$ is unsatisfiable. We then invoke [KP89] to obtain a P-uniform sequence $\Psi^*$ of $\psi^*_\lambda \in \mathsf{UNSAT}$ without short proofs of unsatisfiability in an appropriate proof system. It follows that $\mathcal{E}_{\varphi\vee\psi_\lambda}$ is a

valid instance of ImpMCSP$^*$.

The second stage is to transform an instance of ImpMCSP$^*$ into an instance of ImpMCSP. To do so, we rely on the techniques discussed above for transforming an instance of CGL into an instance of Total-Learn. An important observation here is that the augmentation of $\mathcal{E}$ to be total never introduces inconsistencies in the resulting function.

Lastly, since NP-hardness of ImpMCSP implies NP-hardness of Total-Learn, we obtain the second item of Theorem 1 by techniques from [MP24]. This completes the sketch of the proof.

### 1.4 Further Related Work

The NP-hardness of variants of MCSP is the subject of an active line of work. In addition to the aforementioned recent results of [HI25] showing conditional hardness of MCSP, unconditional NP-hardness has been demonstrated for several variants. This includes the partial-function version "MCSP$^*$" [Hir22], a multi-output version [ILO20], an oracle version [Ila20], and a variant where circuits are allowed access to a random oracle [Ila23]. The NP-hardness of variants of MCSP under cryptographic assumptions has been studied in [HIR25]. Barriers to proving NP-hardness have also been studied extensively [HP15; MW17; AHK17; AH19; SS20; RS22; MP24]. Work on (non) NP-hardness of variants of MCSP has also led to unconditional lower bounds in cryptography [Lu+24].

Similarly, the existence of indistinguishability obfuscation is an active subject of study in cryptography. Since its introduction in [Bar+12], a number of constructions of $i\mathcal{O}$ have proven secure under a variety of assumptions ([Gar+16; Gen+15; Lin16; WW21], to name just a few). As mentioned before, the breakthrough work of [JLS26] shows that $i\mathcal{O}$ exists under relatively well-understood assumptions. Variants of $i\mathcal{O}$ have also been studied for models of computation other than standard circuits, including Turing machines [KLW15; Lin+16] and certain circuit models of interest in quantum computing [Can+24]. It has been shown that if NP $\not\subseteq$ io-P/poly, then the existence of (even a weaker form of) $i\mathcal{O}$ is sufficient to imply the existence of one-way functions [Kom+22]. Furthermore, $i\mathcal{O}$ and one-way functions are sufficient to construct a large fraction of "standard cryptography" (see [SW14] and subsequent work).

A number of results showing the NP-hardness of PAC learning for various concept classes are known (e.g., [PV88; Ale+01; Hir22; KST23; GK25]). In particular, [GK25] (building on [Hir22]) show that a constant-gap version of CGL is unconditionally NP-hard.

The question of the (non-)existence of optimal propositional proof systems (as defined by [CR79; KP89]) has been studied quite extensively; see, e.g., [KM98; Buh+00; Sad02; KMT03; BKM09; BKM11; Kra13; CFM14; Gla+14; PS19; Kha22; Kha24]. Propositional proof complexity has had a number of interesting applications in cryptography recently [JJ22; Jin+24; Jin+25; MDS25; Ila25].

**Remainder of the paper.** After stating the basics in Section 2, we prove our main theorems (Theorem 1, Theorem 2, and part of Theorem 3) in Section 3. In Section 4, we prove the conditional impossibility of NP-hardness Levin reductions for Total-Learn and ImpMCSP. We prove Theorem 3 in Section 5. In Section 6, we give equivalences between NP-hardness of Total-Learn and certain variants of Witness Encryption, and derive some limitations on strong variants of Witness Encryption. We state some open questions in Section 7. In Appendix A, we show an unconditional coNP-hardness result for a certain strong version of Total-Learn.

# 2 Preliminaries

## 2.1 Basics

A *polynomial* function is a function $p(n) = n^c$ where $c \geq 1$ is a constant. A *subexponential* function is any $p(n)$ such that, for all constants $c \geq 1$, it holds that $p(n) \leq 2^{n^{1/c}}$. When we say that a runtime (or some other growth rate) is polynomial, we mean that there is some constant $c \geq 1$ such that the runtime on inputs of length $n$ is at most $n^c$. When we say a runtime (growth rate) is subexponential, we mean that the runtime is at most $2^{n^{1/c}}$ for *every* $c \geq 1$.

**Definition 4** (Promise Problems)**.** Let $R_{\mathcal{Y}}, R_{\overline{\mathcal{N}}} \subseteq \{0,1\}^* \times \{0,1\}^*$ be relations such that $R_{\mathcal{Y}} \subseteq R_{\overline{\mathcal{N}}}$. The *promise problem associated with* $(R_{\mathcal{Y}}, R_{\overline{\mathcal{N}}})$ is a decision problem $(\mathcal{Y}, \mathcal{N})$ where one must, given an input $x$, decide whether

1. **(Yes-case)** $x \in \mathcal{Y} := \{z \in \{0,1\}^* \mid \exists\, w\ (z,w) \in R_{\mathcal{Y}}\}$, or
2. **(No-case)** $x \in \mathcal{N} := \{z \in \{0,1\}^* \mid \forall\, w\ (z,w) \notin R_{\overline{\mathcal{N}}}\}$.

The three original ways of defining NP-completeness [Coo71; Kar72; Lev73] differed in the type of reduction used. Some of our results consider the power of Levin's (more restrictive) reductions versus the (more permissive) "many-one" reductions considered by Karp. We formally define Levin reductions as well as the intermediate notion of "half-Levin" reductions (defined by [GK25]) for promise problems next.

**Definition 5** (Levin and Half-Levin Reductions for Promise Problems)**.** Consider promise problems $(\mathcal{Y}^1, \mathcal{N}^1)$ and $(\mathcal{Y}^2, \mathcal{N}^2)$ associated with pairs of relations $(R^1_{\mathcal{Y}}, R^1_{\overline{\mathcal{N}}})$ and $(R^2_{\mathcal{Y}}, R^2_{\overline{\mathcal{N}}})$ respectively.

For a time-bound $t : \mathbb{N} \to \mathbb{N}$, a *deterministic $t$-time Levin reduction* from $(\mathcal{Y}^1, \mathcal{N}^1)$ to $(\mathcal{Y}^2, \mathcal{N}^2)$ is a triplet of deterministic algorithms $(f, g, h)$ such that:

1. **(Many-one)** $f$ maps instances of the first promise problem to instances of the second: for every $x \in \mathcal{Y}^1$, $f(x) \in \mathcal{Y}^2$, and for every $x \in \mathcal{N}^1$, $f(x) \in \mathcal{N}^2$.
2. **(Witness mapping)** $g$ maps yes-witnesses for the first problem to yes-witnesses for the second: for every $(x,w) \in R^1_{\mathcal{Y}}$, $(f(x), g(x,w)) \in R^2_{\mathcal{Y}}$.
3. **(Witness recovery)** $h$ maps "not no"-witnesses for instances of the second problem produced by $f$ back to "not no"-witnesses for the corresponding instances of the first: if $(f(x), w) \in R^2_{\overline{\mathcal{N}}}$ then $(x, h(x,w)) \in R^1_{\overline{\mathcal{N}}}$.

Moreover, $f, g$, and $h$ all run in time at most $t(|x|)$ on the inputs described above.

A *deterministic $t$-time half-Levin reduction* from $(\mathcal{Y}^1, \mathcal{N}^1)$ to $(\mathcal{Y}^2, \mathcal{N}^2)$ is a pair $(f, g)$ running in time $t(|x|)$ and satisfying only the first two conditions above.

**Definition 6** (Randomized Levin and Half-Levin Reductions for Promise Problems)**.** With notation as in Definition 5, a *randomized $t$-time Levin reduction with $\varepsilon$-error* from $(\mathcal{Y}^1, \mathcal{N}^1)$ to $(\mathcal{Y}^2, \mathcal{N}^2)$ is a triplet of randomized algorithms $(f, g, h)$ running in time at most $t(|x|)$ such that:

1. **(Yes-instances)** For every $x \in \mathcal{Y}^1$, with probability at least $1 - \varepsilon$ over the choice of randomness $r_1$, the following hold:
   (a) **(Witness mapping)** $(f(x; r_1), g(x, w; r_1)) \in R^2_{\mathcal{Y}}$, and

(b) **(Witness recovery)** for every $w'$ such that $(f(x;r_1),w') \in R^2_{\mathcal{N}}$, it holds that

$$\Pr_{r_2}[(x,h(x,w';r_1,r_2)) \in R^1_{\mathcal{N}}] \geq 1-\varepsilon.$$

2. **(No-instances)** For every $x \in \mathcal{N}^1$, it holds that $\Pr_{r_1}[f(x;r_1) \in \mathcal{N}^2] \geq 1-\varepsilon$.

A *randomized $t$-time half-Levin reduction with $\varepsilon$-error* is defined analogously, requiring only conditions 1(a) and 2 above.[5]

If $\varepsilon$ is not specified, we assume it is some small constant, say $\varepsilon = 1/100$.

We say that a (randomized) reduction $f$ is *honest* if, for some constant $\delta > 0$, for all inputs $x \in \{0,1\}^*$ (and all choices of internal randomness for $f$), $|f(x)| \geq |x|^\delta$.

## 2.2 Proof complexity

We denote by $\mathsf{TAUT}$ the standard $\mathsf{coNP}$-complete language of propositional tautologies. We will usually not distinguish between tautologies and unsatisfiable propositional formulas.

The following definition of a propositional proof system (PPS) is from the work of Cook and Reckhow [CR79]. The original definition is more general as it applies to any given language $L$, rather than just $\mathsf{TAUT}$.

**Definition 7** (Cook–Reckhow Proof System [CR79])**.** A *propositional proof system (PPS)* is a polynomial-time algorithm $\mathcal{L}\colon \{0,1\}^* \to \mathsf{TAUT}$. For $x \in \mathsf{TAUT}$, the minimum $\mathcal{L}$-proof size of $x$, denoted $\mathcal{L}\text{-}\mathsf{size}(x)$, is

$$\min\{|\pi| \in \{0,1\}^* \mid \mathcal{L}(\pi) = x\};$$

it is $\infty$ if there is no $\pi \in \{0,1\}^*$ such that $\mathcal{L}(\pi) = x$.

**Definition 8** (PPS Simulation [CR79; KP89][6])**.** A PPS $\mathcal{L}$ *polynomially (resp. subexponentially) simulates* a PPS $\mathcal{L}'$ if there is a polynomial $p(n) = n^c$ for some constant $c \geq 1$ (resp. for all constants $c \geq 1$ and $p(n) = 2^{n^{1/c}}$), such that we have, for all $x \in \mathsf{TAUT}$,

$$\mathcal{L}\text{-}\mathsf{size}(x) \leq p(\mathcal{L}'\text{-}\mathsf{size}(x)).$$

We say that a PPS $\mathcal{L}$ *polynomially (resp. subexponentially) infinitely-often simulates* a PPS $\mathcal{L}'$ if there is a polynomial $p(n) = n^c$ for some constant $c \geq 1$ (resp. for all constants $c \geq 1$ and $p(n) = 2^{n^{1/c}}$), so that there are *infinitely many* $n \geq 1$, such that for all $x \in \mathsf{TAUT}$ with $|x| = n$,

$$\mathcal{L}\text{-}\mathsf{size}(x) \leq p(\mathcal{L}'\text{-}\mathsf{size}(x)).$$

[5] Note that condition 1(a) implies that for every $x \in \mathcal{Y}^1$, with probability at least $1-\varepsilon$ over the randomness $r_1$, it holds that $f(x;r_1) \in \mathcal{Y}^2$.

[6] The original paper by Cook and Reckhow [CR79] defined the *uniform* version of simulation of a PPS $\mathcal{L}'$ by a PPS $\mathcal{L}$, sometimes called *p-simulation*, where there is a polynomial-time function $f\colon \{0,1\}^* \to \{0,1\}^*$ such that for every $\mathcal{L}'$-proof $\pi$ of some tautology $x$, $f(\pi)$ is an $\mathcal{L}'$-proof of $x$. Here we follow Krajíček and Pudlák [KP89] who consider the *nonuniform* version of PPS simulation where the existence of a short $\mathcal{L}'$-proof of $x$ implies the existence of a somewhat short $\mathcal{L}$-proof of $x$, but there is no assumption that an $\mathcal{L}$-proof of $x$ can be efficiently uniformly constructed from an $\mathcal{L}'$-proof of $x$.

**Definition 9** (Optimal PPS [KP89])**.** A PPS $\mathcal{L}$ is *polynomially (resp. subexponentially) optimal* if it polynomially (resp. subexponentially) simulates every PPS $\mathcal{L}'$.

We say a PPS $\mathcal{L}$ is *polynomially (resp. subexponentially) infinitely-often optimal* if it polynomially (resp. subexponentially) infinitely-often simulates every PPS $\mathcal{L}'$.

Krajíček and Pudlák [KP89] conjectured that there is no polynomially optimal PPS. This conjecture is a strengthening of the conjecture that $\mathsf{NP} \neq \mathsf{coNP}$. In fact, assuming no polynomially optimal PPS exists, one can show that $\mathsf{NE} \neq \mathsf{coNE}$ [KP89], and that $\mathsf{NEE} \neq \mathsf{coNEE}$ [KMT03].

The following result of [KP89] shows that the assumption that no optimal PPS exists is equivalent to a natural "constructive" version, where for every candidate optimal PPS $\mathcal{L}$, one can efficiently generate a sequence of tautologies that are hard for $\mathcal{L}$.[7] While the original proof in [KP89] was about the case of a polynomially optimal PPS, it naturally extends to the case of a subexponentially optimal PPS as well. Below, by a $\mathsf{P}$-uniform family of formulas $\{\psi_n\}$ (of size $n$ each), we mean that there is a deterministic polynomial-time algorithm $G$ such that, for every $n \geq 1$, $G(1^n)$ outputs $\psi_n$.

**Theorem 10** ([KP89])**.** *The following are equivalent:*

- *there is no polynomially (resp. subexponentially) optimal PPS, and*
- *for every PPS $\mathcal{L}$, there is a $\mathsf{P}$-uniform sequence of unsatisfiable formulas $\psi_n$, of size $n$, such that, for every polynomial $p(n) = n^c$ with $c \geq 1$ (resp. there exists $p(n) = 2^{n^{1/c}}$ with $c \geq 1$), there are infinitely many $n \geq 1$ where $\mathcal{L}\text{-}\mathsf{size}(\psi_n) \geq p(n)$.*

Moreover, as observed by Ilango [Ila25] the result above also naturally extends to the "infinitely often" case.

**Theorem 11** ([KP89; Ila25])**.** *The following are equivalent:*

1. *there is no polynomially (resp. subexponentially) infinitely-often optimal PPS, and*
2. *for every PPS $\mathcal{L}$, there is a $\mathsf{P}$-uniform sequence of unsatisfiable formulas $\psi_n$, of size $n$, such that, for every polynomial $p(n) = n^c$ with $c \geq 1$ (resp. there exists $p(n) = 2^{n^{1/c}}$ with $c \geq 1$), for all but finitely many $n \geq 1$, it holds that $\mathcal{L}\text{-}\mathsf{size}(\psi_n) \geq p(n)$.*

Inspired by [Ila25], we consider PPS based on certain "falsifiable properties". For a family of propositional formulas $\Psi = \{\psi_n\}_{n\geq 0}$ (with $|\psi_n| = n$), consider any $\Pi_1^{\mathsf{subexp}}$ statement $\mathcal{P}[\psi_n]$, checkable in $\mathsf{coNTIME}[\mathsf{subexp}(n)]$ when given $\psi_n$. Suppose that one can prove that, for all $n \geq 0$,

$$\psi_n \in \mathsf{SAT} \implies \mathcal{P}[\psi_n].$$

It follows that if $\mathcal{P}[\psi_n]$ is *false*, then $\psi_n$ must be *unsatisfiable*. Thus, a proof that $\mathcal{P}[\psi_n]$ is false yields a proof that $\psi_n$ is unsatisfiable. More precisely, any such property $\mathcal{P}$ yields a PPS $\mathcal{L}_{\mathcal{P}}$, where an $\mathcal{L}_{\mathcal{P}}$-proof $\pi$ of the unsatisfiability of $\psi_n$ consists of a witness of $\mathsf{subexp}(n)$-size that $\mathcal{P}[\psi_n]$ is false along with the transcript of an accepting computation of the $\mathsf{subexp}(n)$-time nondeterministic Turing machine deciding the complement of $\mathcal{P}[\psi_n]$.

As a consequence, if $\{\psi_n\}_{n\geq 0}$ is a family of tautologies that are almost everywhere "hard" for the PPS $\mathcal{L}_{\mathcal{P}}$, then it must be the case that $\mathcal{P}[\psi_n]$ holds for almost all $n \geq 1$. If, moreover, such a family

[7] This is related to the notion of a *refuter* from [Kab01] for constructive separations of complexity classes (see, e.g., [Che+24]), but adapted to the case of PPS.

of hard tautologies $\{\psi_n\}_{n\geq 0}$ can be generated efficiently uniformly (say, by assuming that there is no optimal PPS and using Theorem 11), then we get an efficient uniform construction of a family of objects that satisfy the property $\mathcal{P}$. As we shall see below, under additional cryptographic assumptions, such objects may be, for example, the truth tables of hard boolean functions (see Lemma 29). Another application is the correctness analysis of a half-Levin NP-hardness reduction for Total-Learn (see Theorem 33) and ImpMCSP (Theorem 40). For more applications, see also the original paper by Ilango [Ila25].

### 2.3 Cryptography

Throughout this work, we consider cryptographic primitives secure against *non-uniform* adversaries.

**Definition 12** (Nonuniform Computational Indistinguishability)**.** Two distributions $\mathcal{D}$ and $\mathcal{D}'$ over $\{0,1\}^*$ are $\varepsilon$-indistinguishable, denoted $\mathcal{D} \approx_\varepsilon \mathcal{D}'$, if, for every boolean circuit $A$ of size at most $s = \varepsilon^{-1}$, it holds that

$$\left| \Pr_{x\sim\mathcal{D}}[A(x)=1] - \Pr_{x\sim\mathcal{D}'}[A(x)=1] \right| < \varepsilon.$$

**Definition 13** (Indistinguishability Obfuscation [Bar+12])**.** A polynomially (resp. subexponentially) secure *indistinguishability obfuscator* is a polynomial-time randomized algorithm $i\mathcal{O}$ that, given a circuit $C : \{0,1\}^n \to \{0,1\}$ and a security parameter $1^\lambda$, outputs a functionally equivalent circuit $i\mathcal{O}(C, 1^\lambda)$ such that the following holds. For all sufficiently large $\lambda$, functionally equivalent circuits $C_1$ and $C_2$ with $|C_1| = |C_2| \leq \mathsf{poly}(\lambda)$, and inverse polynomial (resp. subexponential) function $\varepsilon(\lambda)$,

$$i\mathcal{O}(C_1, 1^\lambda) \approx_{\varepsilon(\lambda)} i\mathcal{O}(C_2, 1^\lambda).$$

**Definition 14** (Witness Encryption [Gar+13])**.** Consider a language $L \in \mathsf{NP}$ with witness relation $R_L$ and a length function $\ell : \mathbb{N} \to \mathbb{N}$. An $\ell$-length polynomially-secure (resp. subexponentially-secure) *witness encryption scheme* for $L$ consists of a pair $(\mathsf{Enc}, \mathsf{Dec})$ of uniform polynomial-time algorithms such that the following hold:

- Given an $L$-instance $x \in \{0,1\}^n$, a message bit $b \in \{0,1\}$, and randomness $r \sim \mathcal{U}_{\mathsf{poly}(n)}$, $\mathsf{Enc}(x, b; r)$ outputs a ciphertext $c \in \{0,1\}^{\ell(n)}$.
- Given an instance/witness pair $(x, w) \in R_L$ and $c \in \{0,1\}^{\ell(n)}$, $\mathsf{Dec}(x, w, c)$ outputs some $b \in \{0, 1, \perp\}$.

Moreover, Enc and Dec have the properties below.

- **Correctness:** For all sufficiently large $n \in \mathbb{N}$, any $b \in \{0,1\}$, $x \in \{0,1\}^n \cap L$, and $w$ such that $(x, w) \in R_L$,
$$\Pr_r[\mathsf{Dec}(x, w, \mathsf{Enc}(x, b; r)) = b] = 1.$$
- **Security:** For any sufficiently large $n \in \mathbb{N}$, $x \in \{0,1\}^n \setminus L$, and inverse polynomial (resp. subexponential) function $\varepsilon$,
$$(x, \mathsf{Enc}(x, 1; r)) \approx_\varepsilon (x, \mathsf{Enc}(x, 0; r)),$$
where $r \sim \mathcal{U}_{\mathsf{poly}(n)}$.

Non-interactive witness indistinguishable proofs [DN07; BOV07] ("NIWI"s) are non-interactive proof systems that satisfy a weak form of zero knowledge. Traditionally, NIWIs are allowed to have randomized provers *and* verifiers, though there are constructions of NIWIs from various assumptions [BOV07; GOS06] that have deterministic verifiers. We use the definition of non-interactive witness indistinguishable proof system (NIWI) considered in [Ila25], which insists on a deterministic verifier. We note that a deterministic verifier is without loss of generality if promise-BPP = promise-P.

**Definition 15** (Non-interactive Witness Indistinguishable Proof System (NIWI))**.** A NIWI proof system is a pair $(P, V)$ of uniform polynomial-time algorithms, where the prover $P$ is randomized, and the verifier $V$ is deterministic, and such that the following hold:

- **Completeness:** for all satisfiable formulas $\varphi$ and $w$ such that $\varphi(w) = 1$, and all security parameters $\lambda$,
$$\Pr_P[V(\varphi, P(\varphi, w, 1^\lambda), 1^\lambda) = 1] = 1$$
- **Perfect Soundness:** for all $\lambda$, unsatisfiable $\varphi$ with $|\varphi| \leq \lambda$, and $\pi$,
$$V(\varphi, \pi, 1^\lambda) = 0.$$
- **Subexponential Witness Indistinguishability/Security:** For some $\varepsilon(\lambda) = 2^{-\lambda^{\Omega(1)}}$, for all satisfiable formulas $\varphi$ and $w, w'$ such that $\varphi(w) = \varphi(w') = 1$, and for all $\lambda$,
$$P(\varphi, w, 1^\lambda) \approx_{\varepsilon(\lambda)} P(\varphi, w', 1^\lambda).$$

Though the following was originally only stated to yield standard (i.e., polynomially-secure) trapdoor one-way permutations, the proof actually immediately yields subexponentially-secure trapdoor one-way permutations. See [SZ25a, Theorem 53] in particular.

**Lemma 16** ([SZ25b])**.** *If subexponentially-secure one-way functions exist and subexponentially-secure $i\mathcal{O}$ exists, then subexponentially-secure full-domain trapdoor one-way permutations exist.*

The following was originally stated for the "polynomial regime" and assuming standard (non-trapdoor) one-way permutations, but the argument is easily modified to yield the following. In particular, see the remark in [BP15] on keyed families of permutations.

**Lemma 17** ([BP15])**.** *If subexponentially-secure $i\mathcal{O}$ exists and subexponentially-secure full-domain trapdoor one-way permutations exist, then there exist subexponentially-secure* NIWI*s with randomized verifiers and negligible soundness error, i.e., there exists a pair of randomized polynomial-time algorithms $(P, V)$ such that the following hold.*

- ***Completeness:*** *for all satisfiable formulas $\varphi$ and $w$ such that $\varphi(w) = 1$, and all security parameters $\lambda$,*
$$\Pr_{P,V}[V(\varphi, P(\varphi, w, 1^\lambda), 1^\lambda) = 1] = 1.$$
- ***Negligible Soundness Error:*** *there exists a negligible function $\mu$ such that for all $\lambda$, unsatisfiable $\varphi$ with $|\varphi| \leq \lambda$, and $\pi$,*
$$\Pr_V[V(\varphi, \pi, 1^\lambda) = 1] \leq \mu(\lambda).$$

- ***Subexponential Witness Indistinguishability/Security:*** *For some* $\varepsilon(\lambda) = 2^{-\lambda^{\Omega(1)}}$*, for all satisfiable formulas* $\varphi$ *and* $w, w'$ *such that* $\varphi(w) = \varphi(w') = 1$*, and for all* $\lambda$*,*

$$P(\varphi, w, 1^\lambda) \approx_{\varepsilon(\lambda)} P(\varphi, w', 1^\lambda).$$

**Definition 18** (Simulator)**.** Consider any randomized prover $P(\varphi, w, 1^\lambda)$, where $\varphi(w) = 1$ and $\lambda$ is the security parameter. For given $\lambda, s, \varepsilon^{-1} \in \mathbb{N}$, we say that the prover $P$ has an *$s$-size $\varepsilon$-indistinguishable simulator on $\lambda$* if there exists a size-$s$ randomized circuit $S_\lambda$ such that, for all satisfiable formulas $\varphi$ of size at most $\lambda$ and any satisfying assignment $w$ for $\varphi$, we have

$$S_\lambda(\varphi) \approx_\varepsilon P(\varphi, w, 1^\lambda).$$

**Definition 19** (Proof System for $\mathsf{SAT}$ based on Hard Tautologies [Ila25])**.** Assume that there exists a $\mathsf{NIWI}$ proof system $(P_{\mathsf{NIWI}}, V_{\mathsf{NIWI}})$ satisfying the definition above. Let $\Psi = \{\psi_\lambda\}$ be any sequence of formulas of size at most $\lambda$. Consider the following proof system $(P[\Psi], V[\Psi])$:

- **Prover:** $P[\Psi](\varphi, w, 1^\lambda)$ rejects if $|\varphi| > \lambda$ or $\varphi(w) = 0$, and otherwise outputs

$$P_{\mathsf{NIWI}}(\varphi \vee \psi_\lambda, w, 1^\lambda).$$

- **Verifier:** $V[\Psi](\varphi, \pi, 1^\lambda) := V_{\mathsf{NIWI}}(\varphi \vee \psi_\lambda, \pi, 1^\lambda)$.

**Lemma 20** (Properties of $(P[\Psi], V[\Psi])$ [Ila25])**.** *Assume that there exists a* $\mathsf{NIWI}$ *proof system* $(P_{\mathsf{NIWI}}, V_{\mathsf{NIWI}})$*. The proof system* $(P[\Psi], V[\Psi])$ *defined above has the following properties:*

1. *for any* $\Psi = \{\psi_\lambda\}$*,* $(P[\Psi], V[\Psi])$ *has perfect completeness;*
2. *for all* $\lambda$*, if* $\psi_\lambda$ *is unsatisfiable, then* $(P[\Psi], V[\Psi])$ *has perfect soundness on security parameter* $\lambda$*;*
3. *there exists* $\varepsilon^*(\lambda) = 2^{-\lambda^{\Omega(1)}}$ *and a polynomial* $p^*$ *such that for all* $\lambda$*, if* $\psi_\lambda$ *is satisfiable, then* $P[\Psi]$ *has* $p^*(\lambda)$*-size* $\varepsilon^*(\lambda)$*-indistinguishable simulator* $S_\lambda$ *on* $\lambda$*.*[8]

*Proof.* For completeness, if $\varphi(w) = 1$, then $(\varphi \vee \psi_\lambda)(w) = 1$, and we conclude by the completeness of the $\mathsf{NIWI}$ proof system.

For perfect soundness on security parameter $\lambda$ when $\psi_\lambda \notin \mathsf{SAT}$, note that if $\psi_\lambda \notin \mathsf{SAT}$ then for all unsatisfiable $\varphi$, $(\varphi \vee \psi_\lambda) \notin \mathsf{SAT}$. Perfect soundness on $\lambda$ follows from perfect soundness of the $\mathsf{NIWI}$.

Finally, suppose $\psi_\lambda$ is satisfiable with a satisfying assignment $w'$. For any satisfiable $\varphi$ with $\varphi(w) = 1$, and any $\lambda$, we have by the security of $\mathsf{NIWI}$ that

$$P[\Psi](\varphi, w, 1^\lambda) = P_{\mathsf{NIWI}}(\varphi \vee \psi_\lambda, w, 1^\lambda) \approx_{\varepsilon^*(\lambda)} P_{\mathsf{NIWI}}(\varphi \vee \psi_\lambda, w', 1^\lambda) =: S_\lambda(\varphi),$$

where $\varepsilon^*$ is the indistinguishability of the NIWI and $S_\lambda(\varphi)$ is a randomized circuit, with $w'$ and $\psi_\lambda$ hardwired, which simulates the $\mathsf{NIWI}$ prover $P_{\mathsf{NIWI}}(\varphi \vee \psi_\lambda, w', 1^\lambda)$. The size of $S_\lambda(\varphi)$ is $\mathsf{poly}(\lambda) =: p^*(\lambda)$, since the $\mathsf{NIWI}$ prover runs in time $\mathsf{poly}(\lambda)$ and the conversion from a uniform Turing machine to a circuit is efficient. □

[8] Note that in this case, we can't assume that the proof system $(P[\Psi], V[\Psi])$ has *perfect soundness.* This is in line with the impossibility results for variant zero-knowledge proof systems of [GO94] who show, in particular, that a proof system for a language $L$ with perfect soundness cannot have a simulator, unless $L \in \mathsf{BPP}$.

**Lemma 21.** *Let $p^*$ and $\varepsilon^*$ be as in Lemma 20. Assume that*

- *there exists a* NIWI *proof system $(P_{\mathsf{NIWI}}, V_{\mathsf{NIWI}})$, and*
- *there is no infinitely-often subexponentially-optimal PPS.*

*For any $\Psi$, let $(P[\Psi], V[\Psi])$ be a proof system as defined above. Let $\mathcal{P}[\psi_\lambda]$ be any $\Pi_1^{\mathsf{subexp}}$ property such that one can prove*

$$\text{“}P[\Psi] \text{ has a } p^*(\lambda)\text{-size } \varepsilon^*(\lambda)\text{-indistinguishable simulator } S_\lambda \text{ on } \lambda\text{”} \implies \mathcal{P}[\psi_\lambda].$$[9]

*Then there is a* P*-uniform sequence of unsatisfiable formulas $\Psi^* = \{\psi_\lambda^*\}$ such that*

- *for every $\lambda$, $\mathcal{P}[\psi_\lambda^*]$ is true, and*
- *the proof system $(P[\Psi^*], V[\Psi^*])$ has completeness and perfect soundness.*

*Proof.* By item (3) of Lemma 20, we know that a proof that $\mathcal{P}[\psi_\lambda]$ is false implies that $\psi_\lambda$ is unsatisfiable. This yields a PPS $\mathcal{L}_{\mathcal{P}}$ where the unsatisfiability of $\psi_\lambda$ has proof size $\mathsf{subexp}(\lambda)$, whose soundness is implied by the security of the NIWI proof system.

By Theorem 11, there is a P-uniform sequence of unsatisfiable formulas $\Psi^* = \{\psi_\lambda^*\}$ such that, for every $\lambda$, the PPS $\mathcal{L}_{\mathcal{P}}$ cannot prove the unsatisfiability of $\psi_\lambda^*$ with a subexp-size proof. It follows that $\mathcal{P}[\psi_\lambda^*]$ must be true for all $\lambda$. The last bullet point follows by Lemma 20. □

## 2.4 Learning

We recall the definition of the following decision-problem formulation of learning due to [ABX08].

**Definition 22** (CGL)**.** For $s, g, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$, an instance of the promise problem $\mathsf{CGL}_{g,\varepsilon}[s]$ at input length $n \in \mathbb{N}$ consists of a $\mathsf{poly}(n)$-size circuit $\mathcal{E}$ sampling a joint distribution supported over $\{0,1\}^n \times \{0,1\}$.

- $\mathcal{E}$ is a yes-instance of $\mathsf{CGL}_{g,\varepsilon}[s]$ if there exists a circuit $C$ of size at most $s(n)$ such that, for all $(x, b) \in \mathsf{supp}(\mathcal{E})$, $C(x) = b$.
- $\mathcal{E}$ is a no-instance of $\mathsf{CGL}_{g,\varepsilon}[s]$ if, for every circuit $C$ of size at most $g(s(n))$,

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < 1/2 + \varepsilon(n).$$

We will need the following result from [GK25].

**Lemma 23** (NP-hardness of CGL from WE [GK25])**.** *Assume that subexponentially-secure* WE *exists for* SAT*. Then there exist a polynomial $s$ and functions $g, \varepsilon^{-1} = 2^{n^{\Omega(1)}}$ such that $\mathsf{CGL}_{g,\varepsilon}[s]$ is* NP*-hard under a deterministic polynomial-time half-Levin reduction.*

We define the following variant of CGL that restricts itself to distributions that have full support over examples.

[9] When proving such an implication, one may assume the completeness property of the proof system $(P[\Psi], V[\Psi])$, but not its soundness; cf. the previous footnote.

**Definition 24** (Total-Learn)**.** For functions $s, g, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$, an instance of the promise problem $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ at input length $n \in \mathbb{N}$ consists of a $\mathsf{poly}(n)$-size circuit $\mathcal{E}$ sampling a joint distribution $(X, B)$, where $X$ has full support over $\{0,1\}^n$, and $B$ is supported over $\{0,1\}$.

- $\mathcal{E}$ is a yes-instance of $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ if there exists a circuit $C$ of size at most $s(n)$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1.$$

- $\mathcal{E}$ is a no-instance of $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ if, for every circuit $C$ of size at most $g(s(n))$,

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < 1/2 + \varepsilon(n).$$

Note that in the definition of learning problems CGL and Total-Learn, there is no requirement that a no-instance $\mathcal{E}$ of CGL (or Total-Learn) define a boolean function. That is, it is allowed that for some $x \in \{0,1\}^*$, both $(x, 0)$ and $(x, 1)$ are in the support of the distribution $\mathcal{E}$ of labeled pairs. (For a yes-instance $\mathcal{E}$ of Total-Learn (or CGL), the existence of a boolean circuit $C$ such that $C(x) = b$ for every pair $(x, b) \in \mathsf{supp}(\mathcal{E})$ ensures that this $\mathcal{E}$ defines a (possibly partial) boolean function.)

Adding a requirement that a distribution $\mathcal{E}$ defines a boolean function also in the no-case transforms the learning problem CGL into what we term *implicit partial* MCSP, and the learning problem Total-Learn into *implicit* MCSP, which we define next.

## 2.5 Implicit MCSP

**Definition 25** (Distribution Defining a Boolean Function)**.** We say that a distribution $\mathcal{E}$ over $\{0,1\}^n \times \{0,1\}$ *defines a (partial) boolean function* if, for every $x \in \{0,1\}^n$, there is a most one $b \in \{0,1\}$ such that $(x, b) \in \mathsf{supp}(\mathcal{E})$. We denote by $f_{\mathcal{E}}$ the corresponding (partial) boolean function defined by $\mathcal{E}$. If such a distribution $\mathcal{E}$ has full support over $\{0,1\}^n$, then it defines a total boolean function $f_{\mathcal{E}}$.

We define *implicit* MCSP, where an instance is given by a circuit sampling $(x, f(x))$ for a total boolean function $f$. This is a generalization of "succinct MCSP" (see, e.g., [AHK17]) where a given input circuit describes the truth table of a boolean function in the lexicographic order. In contrast, in an instance of ImpMCSP, the truth table is described by a sampling circuit in some random order: by enumerating all random inputs to a sampling circuit, one gets the graph $(x_1, f(x_1)), \ldots, (x_{2^n}, f(x_{2^n}))$ of a boolean function $f$ in some order over strings $x_i \in \{0,1\}^n$.

We define the following gap version of implicit MCSP, denoted Gap-ImpMCSP.

**Definition 26** (Gap-ImpMCSP)**.** For $s, g, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$, an instance of the promise problem $\mathsf{Gap}_{g,\varepsilon}\text{-}\mathsf{ImpMCSP}[s]$ at input length $n \in \mathbb{N}$ consists of a $\mathsf{poly}(n)$-size circuit $\mathcal{E}$ sampling a distribution over $\{0,1\}^n \times \{0,1\}$ defining a total boolean function $f_{\mathcal{E}}$.

- $\mathcal{E}$ is a yes-instance of $\mathsf{Gap}_{g,\varepsilon}\text{-}\mathsf{ImpMCSP}[s]$ if there exists a circuit $C$ of size at most $s(n)$ such that, for all $x \in \{0,1\}^n$, $C(x) = f_{\mathcal{E}}(x)$.

- $\mathcal{E}$ is a no-instance of $\mathsf{Gap}_{g,\varepsilon}\text{-}\mathsf{ImpMCSP}[s]$ if, for every circuit $C$ of size at most $g(s(n))$,

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < 1/2 + \varepsilon(n).$$

Note that in the yes-case of ImpMCSP, $\mathcal{E}$ defines an "easy" total boolean function $f_{\mathcal{E}}$; whereas, in the no-case, $\mathcal{E}$ defines a very hard total boolean function $f_{\mathcal{E}}$, and moreover, the restriction of $\mathcal{E}$ to its first coordinate is a *hard-core* distribution for $f_{\mathcal{E}}$.

**Remark 27.** The natural *non-gap* version of implicit MCSP would have as no-instances circuits sampling a distribution $\mathcal{E}$ such that no circuit of size at most $s(n)$ can compute the function defined by $\mathcal{E}$, correctly on *all* inputs. This version of implicit MCSP can be defined as $\mathsf{Gap}_{id,1/2}$-ImpMCSP, where $id$ is the identity function; it is easy to see that $\mathsf{Gap}_{id,1/2}$-ImpMCSP is in the class $\Sigma_2^p$. Strengthening the no-case condition to circuit size at most $g(s(n))$ is the first "gap" in our definition of Gap-ImpMCSP, yielding $\mathsf{Gap}_{g,1/2}$-ImpMCSP; requiring that no large circuit can agree with $\mathcal{E}$ with probability more than $1/2 + \varepsilon$ is another "gap", yielding $\mathsf{Gap}_{g,\varepsilon}$-ImpMCSP in promise-MA.

We also define implicit MCSP$^*$ where the function defined by a sampling circuit $\mathcal{E}$ need not be total.

**Definition 28** (Gap-ImpMCSP$^*$)**.** For $s, g, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$, an instance of the promise problem $\mathsf{Gap}_{g,\varepsilon}$-ImpMCSP$^*[s]$ at input length $n \in \mathbb{N}$ consists of a poly$(n)$-size circuit $\mathcal{E}$ sampling a distribution over $\{0,1\}^n \times \{0,1\}$ defining a (possibly partial) boolean function $f_{\mathcal{E}}$.

- $\mathcal{E}$ is a yes-instance of $\mathsf{Gap}_{g,\varepsilon}$-ImpMCSP$^*[s]$ if there exists a circuit $C$ of size at most $s(n)$ such that,
$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1.$$
- $\mathcal{E}$ is a no-instance of $\mathsf{Gap}_{g,\varepsilon}$-ImpMCSP$^*[s]$ if, for every circuit $C$ of size at most $g(s(n))$,
$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < 1/2 + \varepsilon(n).$$

# 3 NP-Hardness and Circuit Lower Bounds

## 3.1 Circuit Lower Bounds for NP

No-instances of CGL are distributions over labeled examples that have high circuit complexity. We can use Lemma 23 to generate no-instances of CGL, by simply feeding the NP-hardness reduction with unsatisfiable formulas. It is tempting to argue that these no-instances of CGL induce hard *functions* with high circuit complexity. However, this approach runs into a problem: while running the reduction on an unsatisfiable formulas does generate a no-instance $\mathcal{E}$ of CGL, that instance might not be consistent with any boolean function! The reason is that we may have, for some $x \in \{0,1\}^n$, that both $(x,0)$ and $(x,1)$ are in the support of $\mathcal{E}$. (In fact, it is trivial to create a hard instance of CGL by taking $\mathcal{E}$ to be the uniform distribution over $\{0,1\}^n \times \{0,1\}$; by definition, any circuit $C$ (of any size) will be correct on $\mathcal{E}$ with probability $1/2$.)

In order to get a hard boolean function, we need to ensure that the hard instance $\mathcal{E}$ of CGL defines a (partial) *boolean* function, i.e., that for every $x \in \{0,1\}^n$, there is at most one value of $b \in \{0,1\}$ such that $(x,b) \in \mathsf{supp}(\mathcal{E})$. We will achieve this by additionally assuming that there is no optimal PPS. More precisely, we have the following.

**Lemma 29** (Circuit lower bounds from WE and non-existence of optimal PPS)**.** *Assume that*

1. (CRYPTO) *subexponentially secure* WE *exists for* SAT*, and*

2. (PROOF COMPLEXITY) *there is no infinitely-often polynomially-optimal PPS.*

*Then* $\mathsf{NP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$*, and hence,* $\mathsf{promise\text{-}BPP} \subseteq \mathsf{promise\text{-}QP}$.

*Proof.* By Lemma 23, we have a deterministic polynomial-time many-one reduction from $\mathsf{SAT}$ to $\mathsf{CGL}_{g,\varepsilon}[s]$, for some polynomial $s$ and any subexponential $g, \varepsilon^{-1}$. The reduction is: given $\varphi$, output the circuit

$$D_\varphi(b, r) := (Enc(\varphi, b, r), b),$$

where $b \in \{0,1\}$ is a secret message to encode and $r \in \{0,1\}^{\mathsf{poly}(n)}$ is the internal randomness of $\mathsf{WE}$ Encoder.

The reduction has the following property: For $\varphi \in \mathsf{SAT}$, there are no distinct $r, r'$ such that $Enc(\varphi, 0, r) = Enc(\varphi, 1, r')$. Indeed, otherwise, a deterministic decoder for $\mathsf{WE}$ would not be able to recover a message $b \in \{0,1\}$ unambiguously from its $\mathsf{WE}$ encoding.

Hence, if there exist $r \neq r'$ such that $Enc(\varphi, 0, r) = Enc(\varphi, 1, r')$, then we know that $\varphi$ must be unsatisfiable. Moreover, the witness size is $|r| + |r'| \leq \mathsf{poly}(n)$, and it can be verified in some fixed deterministic time $\mathsf{poly}(n)$. This can be viewed as the following PPS $\mathcal{L}$:

- an input $\pi$ is interpreted as $(\varphi, r, r')$, with $|r| = |r'| \leq \mathsf{poly}(|\varphi|)$,
- if $Enc(\varphi, 0, r) = Enc(\varphi, 1, r')$, then the output is $\varphi$; otherwise, the output is some trivial unsatisfiable formula.

The soundness of this proof system $\mathcal{L}$ follows from the assumed existence of $\mathsf{WE}$ for $\mathsf{SAT}$.

By our proof complexity assumption, we get from Theorem 11 a deterministic polynomial-time algorithm for generating a sequence of unsatisfiable formulas $\psi_n$ (of length $n$ each) such that $\mathcal{L}$ cannot prove the unsatisfiability of $\psi_n$, with polynomial-size proofs, for all but finitely many $n \geq 1$.

It follows that for almost all such $\psi_n$, the distributions $Enc(\psi_n, 0, r)$ and $Enc(\psi_n, 1, r')$ (over uniformly random $r, r'$) have disjoint supports. Hence, the sampler $D_{\psi_n}(b, r)$ defines a partial boolean function $h\colon \{0,1\}^{\ell(n)} \to \{0,1\}$, for some $\ell(n) \leq \mathsf{poly}(n)$, that cannot be computed by any boolean circuit of size $2^{n^\nu}$, for some $\nu > 0$, with probability 1. That is, the worst-case circuit complexity of $h$ is at least $2^{n^\nu}$. Since $\ell(n) \leq \mathsf{poly}(n)$, the worst-case circuit complexity of $h$ (relative to its input length $\ell(n)$) is at least subexponential.

Since every boolean function on inputs of length $\ell$ can be computed by a circuit of size at most $2^\ell$, we get that the input length $\ell(n)$ of $h$ is such that $\ell(n) \geq n^\nu$. By evaluating $D_{\psi_n}(b, r)$ over all $b$ and $r$, we can compute the truth table of this partial boolean function $h$ in time exponential in $n$, and hence exponential in its input length $\ell(n)$. We can then make $h$ into a total boolean function by assigning the value 0 to all other inputs.

Moreover, this total boolean function $f$ is computable in $\mathsf{NP}$: Once we have a circuit $D_{\psi_n}$ (which can be efficiently and uniformly computed since $\psi_n$ is a $\mathsf{P}$-uniform family), to check if $f(x) = 1$ on a given $x$, we try to nondeterministically guess $r$ and check if $D_{\psi_n}(1, r) = (x, 1)$.

So we get that $\mathsf{NP} \subseteq \mathsf{EXP}$ requires subexponential circuit size almost everywhere. The derandomization consequence is by applying standard hardness-randomness tradeoffs [NW94; Bab+93; IW97; Uma03]. □

**Remark 30.** Since $\mathsf{WE}$ follows from $i\mathcal{O}$ [Gar+16] (with essentially the same security), Item 1 in Lemma 29 can be replaced by the assumption that subexponentially secure $i\mathcal{O}$ exists.

Lemma 29 should be contrasted with a result of [IKV23] who show that the assumed existence of $i\mathcal{O}$ (secure against all polynomial-size circuits) would imply that $\mathsf{NEXP} \not\subset \mathsf{P/poly}$. Note that the concluded circuit lower bound is for a large complexity class $\mathsf{NEXP}$, and the circuit complexity is superpolynomial only infinitely often. On the other hand, thanks to an extra proof-complexity assumption, our Lemma 29 achieves *almost-everywhere* (subexponential) circuit lower bounds for a smaller class $\mathsf{NP} \subseteq \mathsf{EXP}$ which, in particular, implies strong derandomization of $\mathsf{BPP}$.

Komargodski et al. [Kom+22] showed that if there exists $i\mathcal{O}$ and $\mathsf{NP} \not\subseteq \mathsf{io\text{-}P/poly}$, then there exist one-way functions. At least in the case of perfect $i\mathcal{O}$, their argument (outlined in Appendix A of their paper) extends to the subexponential regime, i.e., if there exists subexponentially-secure $i\mathcal{O}$ and $\mathsf{NP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$, then there exist subexponentially-secure one-way functions. We thus have the following lemma.

**Lemma 31.** *Assume that*

1. (CRYPTO) *subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often polynomially-optimal PPS.*

*Then subexponentially-secure one-way functions exist.*

### 3.2 NP-Hardness of Implicit MCSP*

**Theorem 32** ($\mathsf{NP}$-hardness of $\mathsf{ImpMCSP}^*$)**.** *Assume the following:*

1. (CRYPTO) *there exist a subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then there exists a polynomial $s'$ such that for all subexponential functions $g'$ and $(\varepsilon')^{-1}$, we have that $\mathsf{Gap}_{g',\varepsilon'}\text{-}\mathsf{ImpMCSP}^*[s']$ is $\mathsf{NP}$-hard under a deterministic polynomial-time half-Levin reduction.*

*Proof.* We follow the proof of Lemma 29. Let $R$ be a deterministic $\mathsf{NP}$-hardness reduction from $\mathsf{SAT}$ to $\mathsf{CGL}_{g,\varepsilon}[s]$, as in Lemma 23. For a $\mathsf{P}$-uniform sequence of unsatisfiable formulas $\psi_n$ (of size $n$ each) to be defined, consider the following reduction $R'$ from $\mathsf{SAT}$ to $\mathsf{CGL}$:

> On input formula $\varphi$ of size $n$, output $R(\varphi \vee \psi_n)$.

By the properties of $R$ (as discussed in the proof of Lemma 29), if $\varphi \in \mathsf{SAT}$, we get that $\varphi \vee \psi_n \in \mathsf{SAT}$, and hence the sampling circuit $\mathcal{E}$ produced by $R(\varphi \vee \psi_n)$ defines a (partial) boolean function of circuit complexity at most $s(\mathsf{poly}(n))$.

Next, suppose that for some $\varphi \notin \mathsf{SAT}$, we get that the sampling circuit $\mathcal{E}$ produced by $R(\varphi \vee \psi_n)$ does *not* define a function, i.e., there are inconsistent-label pairs $(x, 0)$ and $(x, 1)$ in the support of $\mathcal{E}$. This implies that $\varphi \vee \psi_n \notin \mathsf{SAT}$ and hence that $\psi_n \notin \mathsf{SAT}$.

We get a PPS for proving $\psi_n$ is unsatisfiable, where a proof is $(r, r', \varphi)$ such that $r \neq r'$, $R(\varphi \vee \psi_n) = \mathcal{E}$, and $\mathcal{E}(r) = (x, 0)$ and $\mathcal{E}(r') = (x, 1)$ for some $x$.

As in the proof of Lemma 29, we get a $\mathsf{P}$-uniform family of unsatisfiable formulas $\psi_n$ that cannot be efficiently proved by our PPS defined above. The resulting reduction $R'$ is then a deterministic polynomial-time half-Levin reduction from $\mathsf{SAT}$ to $\mathsf{Gap}_{g',\varepsilon'}\text{-}\mathsf{ImpMCSP}^*[s']$. □

### 3.3 NP-Hardness of Total-Learn

We prove our conditional NP-hardness result for Total-Learn assuming the existence of NIWIs. The NP-hardness item of Theorem 2 from the introduction is proved as a corollary (Corollary 38).

**Theorem 33** (NP-hardness of Total-Learn)**.** *Assume the following:*

1. (CRYPTO) *there exist subexponentially secure $i\mathcal{O}$ and a subexponentially secure* NIWI*, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then there exists a polynomial $s'$ such that for all subexponential functions $g'$ and $(\varepsilon')^{-1}$, we have that* $\mathsf{Total\text{-}Learn}_{g',\varepsilon'}[s']$ *is* NP*-hard under a deterministic polynomial-time half-Levin reduction.*

*Proof.* Let $s'$ be a polynomial to be set later, and fix $g'$ and $(\varepsilon')^{-1}$ to be arbitrary subexponential functions. Let $s$, $g$, an $\varepsilon$ be the parameters given by Lemma 23. Note that the existence of subexponentially secure $i\mathcal{O}$ implies the existence of subexponentially secure WE. Thus, by Lemma 23 there is a polynomial-time many-one reduction that, given a SAT instance $\varphi$, outputs a $\mathsf{CGL}_{g,\varepsilon}[s]$ instance $\mathcal{E}_\varphi$ samplable by a polynomial-size circuit $C_\varphi$ on $\rho(n) = \mathsf{poly}(n)$ inputs.

By definition of CGL, samples from $\mathcal{E}_\varphi$ are tuples $(x, b)$ with $|x| =: a(n) = \mathsf{poly}(n)$ and $b \in \{0,1\}$. Consider the language

$$L = \{(\varphi, x) \mid |x| = a(|\varphi|) \wedge \exists\, b \in \{0,1\}\,, (x,b) \in \mathsf{supp}(\mathcal{E}_\varphi)\}.$$

We have $(\varphi, x) \in L$ if and only if there exist $b \in \{0,1\}$ and $r \in \{0,1\}^{\rho(|\varphi|)}$ such that $C_\varphi(r) = (x, b)$.

Observe that $L \in \mathsf{NP}$. Let $\Pi = \Pi[\Psi] = (P[\Psi], V[\Psi])$ be a proof system for membership in $L$ given by Definition 19, for some family $\Psi$ of unsatisfiable formulas to be specified later. Let $\lambda$ be a polynomial in $n$ to be set later, and let $\ell(\lambda)$ be the length of proofs produced by $\Pi$ with security parameter $\lambda$. For a suitably small parameter $\alpha(n) = 2^{-n^{\Omega(1)}}$, define the following distribution $\mathcal{E}_{\varphi,\Pi}$ on $\{0,1\}^{a(n)+\ell(\lambda)} \times \{0,1\}$:

1. sample a uniformly random $r \in \{0,1\}^{\rho(n)}$ and let $(x, b) = C_\varphi(r)$;
2. with probability $1 - \alpha(n)$, sample a proof $\pi \leftarrow P[\Psi]((\varphi, x), r, 1^{\lambda(n)})$, and output $((x, \pi), b)$;
3. with probability $\alpha(n)$, sample $\pi$ uniformly at random from $\{0,1\}^{\ell(\lambda)}$, and
   - if $V[\Psi]((\varphi, x), \pi, 1^\lambda) = 1$, then output $((x, \pi), b)$;
   - otherwise, sample a proof $\pi' \leftarrow P[\Psi]((\varphi, x), r, 1^{\lambda(n)})$, and output $((x, \pi'), b)$.

Observe that $\mathcal{E}_{\varphi,\Pi}$ is supported (in its first component) over all pairs $(x, \pi)$ such that $(\varphi, x) \in L$ and $V[\Psi]((\varphi, x), \pi, 1^\lambda) = 1$.

For the remainder of the proof, let $p^*$ and $\varepsilon^*$ be as in Lemma 20.

**Claim 34.** *If there is a circuit $D$ of size $t$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}_\varphi}[D(x) = b] = 1,$$

*then there is a circuit $D'$ of size at most $t$ such that*

$$\Pr_{((x,\pi),b)\sim\mathcal{E}_{\varphi,\Pi}}[D'(x, \pi) = b] = 1.$$

*Proof of Claim 34.* Define $D'(x,\pi) = D(x)$. □

Note that, as a consequence of Claim 34, if $\mathcal{E}_\varphi$ defines a (possibly partial) boolean function, then so does $\mathcal{E}_{\varphi,\Pi}$.

**Claim 35.** *Suppose there is a circuit $D'$ of size $t$ such that*

$$\Pr_{((x,\pi),b)\sim\mathcal{E}_{\varphi,\Pi}}[D'(x,\pi)=b] \geq \frac{1}{2}+\nu.$$

*If the proof system $\Pi[\Psi]$ admits a $p^*(\lambda)$-size, $\varepsilon^*(\lambda)$-indistinguishable simulator $S$ on $\lambda$ and $t < 1/\varepsilon^*(\lambda)$, then there is a circuit $D$ of size at most $t+p^*(\lambda)$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}_\varphi}[D(x)=b] \geq \frac{1}{2}+(\nu-\varepsilon^*(\lambda)-\alpha(n)).$$

*Proof of Claim 35.* Define a randomized circuit $\tilde{D}(x) = D'(x, S(\varphi,x))$, where $S$ is a $p^*(\lambda)$-size, $\varepsilon^*(\lambda)$-indistinguishable randomized simulator on $\lambda$. We have

$$\begin{aligned}
\frac{1}{2}+\nu &\leq \Pr_{((x,\pi),b)\sim\mathcal{E}_{\varphi,\Pi}}[D'(x,\pi)=b] \\
&\leq \Pr_{r,P}[D'(C_\varphi(r)_1, P[\Psi]((\varphi,C_\varphi(r)_1),r,1^\lambda)) = C_\varphi(r)_2] + \alpha(n) \\
&\leq \Pr_{r,S}[D'(C_\varphi(r)_1, S(\varphi,C_\varphi(r)_1)) = C_\varphi(r)_2] + \varepsilon^*(\lambda) + \alpha(n) \\
&= \Pr_{(x,b)\sim\mathcal{E}_\varphi;\tilde{D}}[\tilde{D}(x)=b] + \varepsilon^*(\lambda) + \alpha(n),
\end{aligned}$$

where the third inequality uses the fact that $S$ is $\varepsilon^*(\lambda)$-indistinguishable for circuits of size $1/\varepsilon^*(\lambda) \geq t = |D'|$.

By averaging, there is a choice of randomness $\widetilde{r}$ used by $\tilde{D}$ such that the deterministic circuit $D$ obtained by fixing $\tilde{D}$'s randomness to $\widetilde{r}$ has the required success probability over $\mathcal{E}_\varphi$. Note that size of this circuit $D$ is at most $|D'|+|S| = t+p^*(\lambda)$. □

Next, we define an extension $\widehat{\mathcal{E}}_{\varphi,\Pi}$ of the distribution $\mathcal{E}_{\varphi,\Pi}$ of labeled pairs $((x,\pi),b)$ so that, for every $(x,\pi)$, there is some $b\in\{0,1\}$, such that $((x,\pi),b)$ has nonzero probability. For a suitably small parameter $\alpha(n) = 2^{-n^{\Omega(1)}}$, define $\widehat{\mathcal{E}}_{\varphi,\Pi}$ on $\{0,1\}^{a(n)+\ell(\lambda)}\times\{0,1\}$ as follows:

1. with probability $1-\alpha(n)$, sample and output $((x,\pi),b)\sim\mathcal{E}_{\varphi,\Pi}$;
2. with probability $\alpha(n)$, sample $(x,\pi)$ uniformly at random from $\{0,1\}^{a(n)+\ell(\lambda)}$. Then,
   - if $V[\Psi]((\varphi,x),\pi,1^\lambda)=0$ then output $((x,\pi),0)$;
   - otherwise, output a fresh sample $((x',\pi'),b')\sim\mathcal{E}_{\varphi,\Pi}$.

Observe that $\widehat{\mathcal{E}}_{\varphi,\Pi}$ has full support over its first component because, for every $(x,\pi)$, we have

- if $V[\Psi]((\varphi,x),\pi,1^\lambda)=0$, then $((x,\pi),0)$ will be output (with some nonzero probability) in the first bullet point of step 2;

- if $V[\Psi]((\varphi, x), \pi, 1^\lambda) = 1$, then $(\varphi, x) \in L$; hence, $(x, \pi)$ is in the support of $\mathcal{E}_{\varphi,\Pi}$, and will be output in step 1.

**Claim 36.** *If there is a circuit $D$ of size $t$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}_\varphi}[D(x) = b] = 1,$$

*then there is a circuit $D'$ of size at most $t + \mathsf{poly}(\lambda)$ such that*

$$\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}_{\varphi,\Pi}}[D'(x, \pi) = b] = 1.$$

*Proof of Claim 36.* Define

$$D'(x, \pi) = \begin{cases} D(x) & \text{if } V[\Psi]((\varphi, x), \pi, 1^\lambda) = 1 \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that $D'$ is correct with probability 1 over $\widehat{\mathcal{E}}_{\varphi,\Pi}$. The size of $D'$ is $|D| + |V| = t + \mathsf{poly}(\lambda)$. □

Note that, as a consequence of Claim 36 and Claim 34, if $\mathcal{E}_\varphi$ defines a (possibly partial) boolean function, then so does $\widehat{\mathcal{E}}_{\varphi,\Pi}$.

**Claim 37.** *Suppose there is a circuit $D'$ of size $t$ such that*

$$\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}_{\varphi,\Pi}}[D'(x, \pi) = b] \geq \frac{1}{2} + \nu.$$

*If the proof system $\Pi[\Psi]$ has a $p^*(\lambda)$-size, $\varepsilon^*(\lambda)$-indistinguishable simulator $S$ on $\lambda$ and $t < 1/\varepsilon^*(\lambda)$, then there is a circuit $D$ of size at most $t + p^*(\lambda)$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}_\varphi}[D(x) = b] \geq \frac{1}{2} + (\nu - \varepsilon^*(\lambda) - 2\alpha(n)).$$

*Proof of Claim 37.* We have

$$\begin{aligned} \frac{1}{2} + \nu &\leq \Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}_{\varphi,\Pi}}[D'(x, \pi) = b] \\ &\leq \Pr_{((x,\pi),b)\sim\mathcal{E}_{\varphi,\Pi}}[D'(x, \pi) = b] + \alpha(n). \end{aligned}$$

Hence,

$$\Pr_{((x,\pi),b)\sim\mathcal{E}_{\varphi,\Pi}}[D'(x, \pi) = b] \geq \frac{1}{2} + \nu - \alpha(n).$$

We conclude the proof by appealing to Claim 35. □

Our reduction, given as input a size-$n$ boolean formula $\varphi$, outputs the description of a $\mathsf{poly}(n)$-size circuit that samples $\widehat{\mathcal{E}}_{\varphi,\Pi}$. It is easy to see that the reduction runs in uniform polynomial time as long as $\Psi$ is a $\mathsf{P}$-uniform family. We now prove completeness and soundness of our reduction.

**Completeness.** Suppose $\varphi \in \mathsf{SAT}$, $|\varphi| = n$. Then, by completeness of the reduction in Lemma 23, there is a circuit $D$ of size $s(a(n))$ such that $\Pr_{(x,b)\sim\mathcal{E}_\varphi}[D(x) = b] = 1$. By Claim 36, there is a circuit $D'$ of size $s(a(n)) + \mathsf{poly}(\lambda(n))$ such that $\Pr_{((x,\pi),b)\sim\mathcal{E}_\varphi}[D'(x,\pi) = b] = 1$. Setting $s'$ so that $s(a(n)) + \mathsf{poly}(\lambda(n)) \le s'(a(n) + \ell(\lambda))$ ensures completeness of our reduction to $\mathsf{Total\text{-}Learn}_{g',\varepsilon'}[s']$. Note that $s'$ can be set independently of our choice of $\lambda$ because $\ell$ and the $\mathsf{poly}$ involved are independent of $\lambda$.

**Soundness.** Define $M$ to be the following Turing machine.

On input $w$:

1. Interpret $w$ as a tuple $(n, \varphi, D')$ with $|\varphi| = n$.
2. Accept if $\varphi \in \mathsf{SAT}$ or $|D'| \ge g'(s'(a(n) + \ell(\lambda)))$.
3. Estimate, using the derandomization given by Lemma 29,
$$\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}_{\varphi,\Pi}}[D'(x,\pi) = b]$$
to a suitably small additive error $\delta = 1/\mathsf{subexp}(\lambda) \ll \varepsilon'(a(n) + \ell(\lambda))$.
4. Reject if the estimate is at least $1/2 + \varepsilon'(a(n) + \ell(\lambda))$. Accept otherwise.

Note that for every $w = (n, \varphi, D')$, $M(w)$ runs in time $t_M \le 2^{\mathsf{poly}(n)} \cdot \mathsf{subexp}(\lambda)$ where the $\mathsf{poly}$ factor does not depend on $\lambda$. This is because one can check whether $\varphi \in \mathsf{SAT}$ in $2^{\mathsf{poly}(n)}$ time (independent of $\lambda$) and whether $|D'| \ge g'(s'(a(n) + \ell(n))) = \mathsf{subexp}(\lambda)$ in $\mathsf{subexp}(\lambda)$ time, and one can estimate the success probability of $D'$ (when $D'$ has size subexponential in $\lambda$) in time subexponential in $\lambda$.

Suppose $\Pi$ admits a $p^*(\lambda(n))$-size, $\varepsilon^*(\lambda(n))$-indistinguishable simulator on $\lambda(n)$. Then we claim that for every $w = (n, \varphi, D')$, $M^t(w) = 1$. Indeed, suppose there is some $w = (n, \varphi, D')$ where $M^t(w) = 0$. Then $D'$ is a small circuit (with size $\mathsf{subexp}(\lambda)$) such that
$$\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}_{\varphi,\Pi}}[D'(x,\pi) = b] \ge \frac{1}{2} + \varepsilon'(a(n) + \ell(\lambda)) - \delta.$$
Since $1/\varepsilon^*(\lambda) = 2^{\lambda^{\Omega(1)}} > |D'|$, Claim 37 gives us $D'$, a circuit with size $\mathsf{subexp}(\lambda) + p^*(\lambda) \ll g(s(a(n))) = 2^{n^{\Omega(1)}}$ (since $\lambda$ is polynomial in $n$) that has success probability at least $1/2 + (\varepsilon'(a(n) + \ell(\lambda)) - \delta - \varepsilon^*(\lambda) - 2\alpha(n)) \ge 1/2 + \varepsilon(a(n))$ over $\mathcal{E}_\varphi$ (the inequality holds since $\varepsilon(z) = 2^{-z^{\Omega(1)}}$ whereas the advantage term is inverse subexponential). This contradicts the correctness of the reduction in Lemma 23.

For all $n$, if $\psi_{\lambda(n)} \in \mathsf{SAT}$, then we get by Lemma 20 that $\Pi$ admits a $p^*(\lambda(n))$-size, $\varepsilon^*(\lambda(n))$-indistinguishable simulator on $\lambda(n)$, and hence $M^t(w) = 1$ for all $w = (n, \varphi, D')$. Thus, any $w$ such that $M^t(w) = 0$ yields a proof (checkable in time $t_M$) that $\psi_\lambda$ is unsatisfiable. For any given $0 < \nu < 1$, we can set $\lambda = n^c$ for a large constant $c > 0$ so that the runtime $t_M \le 2^{\lambda^\nu}$.

By Lemma 21, we conclude that one can set $\Psi$ to a $\mathsf{P}$-uniform sequence of unsatisfiable formulas such that the proof system $\Pi^* = \Pi[\Psi^*] = (P^*, V^*)$ for $L$ has perfect completeness and soundness, and for all $w = (n, \varphi, D')$ with sufficiently large $n$, $M^t(w) = 1$ for all $w$. Hence, when instantiated with this $\Pi$, for sufficiently large $n$ and every $\varphi \notin \mathsf{SAT}$, $|\varphi| = n$, no circuit with size at most $g'(s'(a(n) + \ell(\lambda)))$ succeeds with probability greater than $1/2 + \varepsilon'(a(n) + \ell(\lambda))$ on the distribution $\widehat{\mathcal{E}}_{\varphi,\Pi}$. Soundness of our reduction follows. □

**Corollary 38.** *Assume the following:*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then there exists a polynomial $s'$ such that*

1. *for all subexponential functions $g', (\varepsilon')^{-1}$, $\mathsf{Total\text{-}Learn}_{g',\varepsilon'}[s']$ is $\mathsf{NP}$-hard under a deterministic quasipolynomial-time half-Levin reduction; and*
2. *for all subexponential functions $g'$ and for every sufficiently large subexponential function $(\varepsilon')^{-1}$, $\mathsf{Total\text{-}Learn}_{g',\varepsilon'}[s']$ is $\mathsf{NP}$-hard under a randomized polynomial-time half-Levin reduction.*[10]

*Proof.* By Lemma 31, our two assumptions imply that there exist subexponentially-secure one-way functions. Lemmas 16 and 17 then imply that there exists a $\mathsf{NIWI}$ $(P, V)$ with a randomized verifier and negligible soundness error $\mu$. By Lemma 29, the two assumptions of this lemma imply $\mathsf{promise\text{-}BPP} \subseteq \mathsf{promise\text{-}QP}$, which in turn implies that the randomized verifier $V$ can be transformed into a *deterministic* quasipolynomial-time verifier $V'$. Using $\Pi' = (P, V')$ as the $\mathsf{NIWI}$ in the reduction from Theorem 33, we get that there is some polynomial $s'$ such that for all subexponential $g', (\varepsilon')^{-1}$ there is a deterministic quasipolynomial-time half-Levin reduction $R_1^{g',\varepsilon'}$ from $\mathsf{SAT}$ to $\mathsf{Total\text{-}Learn}_{g',\varepsilon'}[s']$, proving item (1). The argument in Theorem 33 works even with a quasipolynomial-time verifier because the only places where the running time of the verifier matter are the following.

1. The circuit the reduction produces embeds the verifier as a circuit. This circuit is now quasipolynomial-size, and is the reason the deterministic reduction in this lemma becomes quasipolynomial-time.
2. The running time of the verifier adds to the running time of the machine $M$ defined in the soundness argument. Recall that this running time (as well as the exact description of the verifier) is used to determine which $\mathsf{P}$-uniform sequence $\Psi$ is used in the reduction. Even with a quasipolynomial-time verifier, the machine $M$ runs in time $2^{\mathsf{poly}(n)} \cdot \mathsf{subexp}(\lambda)$, and so this part of the argument still holds.

We now prove item (2). We begin by showing that a randomized verifier can be turned into a deterministic one that has perfect soundness for some fixed setting of the security parameter by repetition. Let $\ell(\lambda)$ be the length of proofs produced by $P$ and let $\rho(\lambda)$ be the number of random bits used by $V$. Let $k(\lambda) = \lambda + 1 + \ell(\lambda)$. For $r \in (\{0,1\}^{\rho(\lambda)})^{k(\lambda)}$, define $V_r$ to be the following deterministic polynomial-size circuit.

Given $(\varphi, \pi, 1^\lambda)$:

1. Run $V(\varphi, \pi, 1^\lambda; r_1), V(\varphi, \pi, 1^\lambda; r_2), \ldots, V(\varphi, \pi, 1^\lambda; r_{k(\lambda)})$.
2. Accept if all executions accept. Reject otherwise.

[10] "Sufficiently large subexponential" here means larger than some fixed subexponential function.

We have

$$\begin{aligned}
&\Pr_{r\sim(\{0,1\}^{\rho(\lambda)})^{k(\lambda)}}[\exists \varphi \notin \mathsf{SAT}, |\varphi| \le \lambda, \pi \in \{0,1\}^{\ell(\lambda)} : V_r(\varphi,\pi,1^\lambda) = 1] \\
&\qquad \le \sum_{\substack{\varphi\notin\mathsf{SAT}:|\varphi|\le\lambda \\ \pi\in\{0,1\}^{\ell(\lambda)}}} \Pr_{r\sim(\{0,1\}^{\rho(\lambda)})^{k(\lambda)}}[V_r(\varphi,\pi,1^\lambda)=1] \\
&\qquad \le 2^{\lambda+1+\ell(\lambda)}\mu(\lambda)^{k(\lambda)} \\
&\qquad < 2^{\lambda+1+\ell(\lambda)} \cdot \frac{1}{\lambda^{k(\lambda)}} \\
&\qquad = 2^{-k(\lambda)\cdot(\log\lambda - 1)}.
\end{aligned}$$

where the last inequality follows from the fact that $\mu$ is negligible. Thus, on security parameter $\lambda$, sampling $r$ uniformly at random yields a verifier circuit $V_r$ that has perfect soundness with overwhelming probability.

Fix arbitrary subexponential $g'$ and $(\varepsilon')^{-1}$. Let $\alpha$ be the "$\alpha$" parameter from the proof of Theorem 33 as a function of $n$, and let $\nu(n) = \varepsilon' - 3\alpha(n) = 1/\mathsf{subexp}(n)$ (as long as $\varepsilon' > 3\alpha(n)$). Inspecting the construction in that theorem, we note that the reduction $R_1^{g',\nu}(\varphi)$ uses the verifier $V'$ on only one security parameter $\lambda$ that can be efficiently computed given $|\varphi|$. Furthermore, the reduction itself does not simulate $V'$; it merely embeds its description into that of the circuit sampling $\widehat{\mathcal{E}}_{\varphi,\Pi'}$. Consider the following *randomized* reduction $R_2$, which we claim reduces $\mathsf{SAT}$ to $\mathsf{Total\text{-}Learn}_{g',\varepsilon'}[s']$.

On input $\varphi$:

1. Compute $\lambda(|\varphi|)$ where $\lambda$ is the security parameter used in $R_1^{g',\nu}$.
2. Sample $r \in (\{0,1\}^{\rho(\lambda(|\varphi|))})^{k(\lambda(|\varphi|))}$ uniformly at random.
3. Compute the deterministic circuit $V_r$.
4. Compute and output the description of the circuit output by $R_1^{g',\nu}(\varphi)$, but replace all instances of $V'$ with $V_r$.

It is easy to see that $R_2$ runs in polynomial time, as the only super-polynomial-time operation in computing $R_1$ is embedding the description of $V'$ into that of the sampler produced.

Fix some $\varphi$ and let $\widehat{\mathcal{E}} \leftarrow R_2(\varphi)$ with randomness $r$. Suppose $\varphi \in \mathsf{SAT}$. As mentioned above, with overwhelming probability over $r$, $V_r$ has perfect soundness. Since Claim 36 only uses the structure of the sampler $\widehat{\mathcal{E}}$ and perfect soundness of the verifier on security parameter $\lambda$, there is a circuit $D'$ of size at most $s'$ such that $\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}}[D'(x,\pi) = b] = 1$. Hence our reduction is complete with high probability.

Now suppose $\varphi \notin \mathsf{SAT}$. Let $\widehat{\mathcal{E}}' = R_1^{g',\nu}(\varphi)$. Suppose $D'$ is a circuit such that

$$\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}}[D'(x,\pi) = b] \ge 1/2 + \varepsilon'.$$

From the definition of $\widehat{\mathcal{E}}$ in the proof of Theorem 33,

$$\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}}[D'(x,\pi) = b] \le (1-\alpha(n)) \Pr_{\substack{w\sim\mathcal{U};(x,b)\leftarrow C_\varphi(w) \\ \pi\leftarrow P[\Psi]((\varphi,x),w,1^\lambda)}}[D'(x,\pi)=b] + 2\alpha(n).$$

On the other hand,

$$\begin{aligned}\Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}'}[D'(x,\pi)=b] &\geq (1-\alpha(n)) \Pr_{\substack{w\sim\mathcal{U};(x,b)\leftarrow C_\varphi(w)\\ \pi\leftarrow P[\Psi]((\varphi,x),w,1^\lambda)}}[D'(x,\pi)=b]-\alpha(n)\\ &\geq \Pr_{((x,\pi),b)\sim\widehat{\mathcal{E}}}[D'(x,\pi)=b]-3\alpha(n)\\ &\geq \frac{1}{2}+\varepsilon'-3\alpha(n)=\frac{1}{2}+\nu'.\end{aligned}$$

Soundness follows by soundness of $R_1$. □

**Corollary 39.** *Assume the following:*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$,*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system, and*
3. (DERANDOMIZATION) promise-BPP = promise-P.

*Then there exists a polynomial $s'$ such that for all subexponential functions $g'$ and $(\varepsilon')^{-1}$, we have that* Total-Learn$_{g',\varepsilon'}[s']$ *is* NP*-hard under a deterministic polynomial-time half-Levin reduction.*

*Proof.* As outlined in the proof of Corollary 38, the first two assumptions imply that there exist NIWIs with randomized verifiers and negligible soundness error. The third assumption implies that one can derandomize these NIWIs into NIWIs with deterministic verifiers and perfect soundness. We conclude by appealing to Theorem 33. □

### 3.4 NP-Hardness of Implicit MCSP

We get the following strengthening of Theorem 33 from Total-Learn to Gap-ImpMCSP.

**Theorem 40** (NP-hardness of Gap-ImpMCSP)**.** *Assume the following:*

1. (CRYPTO) *there exist subexponentially secure $i\mathcal{O}$ and a subexponentially secure* NIWI*, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then there exists a polynomial $s'$ such that for all subexponential $g', (\varepsilon')^{-1}$,* Gap$_{g',\varepsilon'}$-ImpMCSP$[s']$ *is* NP*-hard under deterministic polynomial-time half-Levin reductions.*

*Proof.* Run the proof of Theorem 33, but instead of starting with the deterministic reduction from SAT to CGL of Lemma 23, start with the deterministic reduction from SAT to ImpMCSP$^*$ of Theorem 32, getting a distribution $\mathcal{E}_\varphi$ for a SAT instance $\varphi$. Note that, by the correctness of the reduction, we have that $\mathcal{E}_\varphi$ defines a (partial) boolean function.

The rest of the argument is the same as in the proof of Theorem 33. By the properties of the distributions $\mathcal{E}_{\varphi,\Pi}$ and $\widehat{\mathcal{E}}_{\varphi,\Pi}$ constructed in that proof, we get that since $\mathcal{E}_\varphi$ defines a boolean function, so does the final distribution $\widehat{\mathcal{E}}_{\varphi,\Pi}$. Since $\widehat{\mathcal{E}}_{\varphi,\Pi}$ has full support over its first component, we get that it defines a total boolean function, as required. □

The following corollary settles the conditional NP-hardness item of Theorem 1 from the introduction.

**Corollary 41** (NP-hardness of Gap-ImpMCSP under randomized reductions)**.** *Assume the following:*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then there exists a polynomial $s'$ such that*

1. *for all subexponential functions $g', (\varepsilon')^{-1}$, $\mathsf{Gap}_{g',\varepsilon'}\text{-}\mathsf{ImpMCSP}[s']$ is NP-hard under a deterministic quasipolynomial-time half-Levin reduction; and*
2. *for all subexponential functions $g'$ and for every sufficiently large subexponential function $(\varepsilon')^{-1}$, $\mathsf{Gap}_{g',\varepsilon'}\text{-}\mathsf{ImpMCSP}[s']$ is NP-hard under a randomized polynomial-time half-Levin reduction.*

*Proof.* The corollary holds by the same proof as that of Corollary 38, with the reduction from SAT to CGL of Lemma 23 replaced by the reduction from SAT to $\mathsf{ImpMCSP}^*$ of Theorem 32. Completeness and soundness of the reduction hold by the same argument as in that corollary. As in Theorem 40, the fact that the reduction produces distributions that define total boolean functions follows from (1) the correctness of the reduction in Theorem 32, (2) the properties of the distributions $\mathcal{E}_{\varphi,\Pi}$ and $\widehat{\mathcal{E}}_{\varphi,\Pi}$, and (3) the fact that Total-Learn instances have full support over their first component. □

### 3.5 Circuit Lower Bounds for NP ∩ coNP

The following is a strengthening of conditional circuit lower bounds of Lemma 29 (under additional assumptions).

**Corollary 42** (Improved circuit lower bounds from crypto and proof complexity assumptions)**.** *Assume the following:*

1. (CRYPTO) *there exist subexponentially secure* WE *and a subexponentially secure* NIWI*, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then* $\mathsf{NP} \cap \mathsf{coNP} \not\subseteq \mathsf{io}\text{-}\mathsf{SIZE}[2^{n^{o(1)}}]$.

*Proof.* Let $R$ be a deterministic polynomial-time half-Levin NP-hardness reduction from Theorem 40. For each $n \in \mathbb{N}$, let $\perp_n$ be some canonical unsatisfiable formula of size $n$. Imagine running $R(\perp_n)$, getting a no-instance $\mathcal{E}$ of ImpMCSP. By the properties of $R$, $\mathcal{E}$ defines a total boolean function $f_{\mathcal{E}} \colon \{0,1\}^m \to \{0,1\}$, for some $m = \mathsf{poly}(n)$, that requires circuit size $\mathsf{subexp}(n)$; in fact, $\mathcal{E}$ is a "hard-core" distribution for $f_{\mathcal{E}}$ with respect to all boolean circuits of size $\mathsf{subexp}(n)$.

We observe that $f_{\mathcal{E}}$ is computable in $\mathsf{NP} \cap \mathsf{coNP}$:

> On input $x \in \{0,1\}^m$, run $R(\perp_n)$ to get a no-instance $\mathcal{E}$ of ImpMCSP, defining the boolean function $f_{\mathcal{E}}$. Note that $f_{\mathcal{E}}(x) = 1$ iff there exists a string $r \in \{0,1\}^{\mathsf{poly}(m)}$ such that $\mathcal{E}(r) = (x, 1)$. Similarly, $f_{\mathcal{E}}(x) = 0$ iff there exists a string $r \in \{0,1\}^{\mathsf{poly}(m)}$ such that $\mathcal{E}(r) = (x, 0)$.

The required conclusion follows. □

# 4 Conditional Impossibility of Levin Reductions

**Theorem 43** (Non-NP-hardness of Total-Learn under Levin reductions)**.** *Assume that subexponentially-secure $i\mathcal{O}$ and subexponentially-secure one-way functions exist. Then there exists a polynomial $g$ such that for every polynomial $s$ and $\varepsilon \leq 1/2$, $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ is not $\mathsf{NP}$-hard under randomized subexponential-time Levin reductions.*

*Proof Sketch.* Mazor and Pass [MP24] proved an analogous result for GapMCSP. We sketch their argument here, modified to work for Total-Learn. We refer the reader to [MP24] for a more detailed proof; the purpose of this sketch is to show that the non-hardness result holds even if we consider samplers as in Total-Learn instead of truth tables as in GapMCSP. Furthermore, we restrict our sketch to honest deterministic reductions, and mention how the result can be extended to potentially non-honest deterministic reductions. By "honest" we mean that the reduction does not output Total-Learn instances of sub-polynomial size. The extension to randomized Levin reductions holds by the argument outlined in [MP24, Section 3.2].

By a result of Rompel [Rom90], the existence of subexponentially-secure one-way functions implies that there is a subexponentially secure "target collision resistant hash" $h : \{0,1\}^* \to \{0,1\}^*$ mapping $n$-bit inputs to $(n - \omega(\log n))$-bit outputs. That is, there is an efficiently computable function $h$ and a constant $\delta > 0$ such that for every sufficiently large $n$ and every $2^{n^\delta}$-time randomized algorithm $A$,

$$\Pr_{w \sim \{0,1\}^n, A}[w' \leftarrow A(w) : h(w) = h(w') \wedge w \neq w'] \leq \mathsf{negl}(n).$$

Consider the language $L = \{x \mid \exists w \ h(w) = x\} \in \mathsf{NP}$. Let $g$ be a polynomial such that for all circuits $C$, $i\mathcal{O}(C, 1^{|C|})$ outputs a circuit of size at most $g(|C|)$, let $s$ be any polynomial, and let $\varepsilon \leq 1/2$ be arbitrary. Assume that there is an honest, deterministic, subexponential-time Levin reduction $(R_0, R_1, R_2)$ from $L$ to $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$. Define the following randomized algorithm $A$:

On input $w$:

1. Compute $x = h(w)$.
2. Compute $\mathcal{E} = R_0(x)$ and $C = R_1(x, w)$.
3. Compute $\widetilde{C} = i\mathcal{O}(C, 1^{|C|})$.
4. Output $\widetilde{w} = R_2(x, \widetilde{C})$.

By definition of $L$, $w$ is a witness for $x \in L$. Thus by correctness of the Levin reduction, $\mathcal{E}$ is a yes-instance of $\mathsf{Total\text{-}Learn}_{g,1/2}[s]$ and $C$ is a circuit of size at most $s$ such that $\Pr_{(y,b)\sim\mathcal{E}}[C(y) = b] = 1$. By correctness of $i\mathcal{O}$, $\Pr_{(y,b)\sim\mathcal{E}}[\widetilde{C}(y) = b] = 1$. Furthermore, $|\widetilde{C}| \leq g(|C|) \leq g(s)$. It follows that $\widetilde{C}$ is a witness for $\mathcal{E}$ not being a no-instance, which means (by correctness of $R_2$) that $\widetilde{w}$ must be a witness for $x \in L$. That is, $h(\widetilde{w}) = x$. Notice that $A$ runs in subexponential time (and hence time less than $2^{n^\delta}$), so if one can argue that $w \neq \widetilde{w}$ with non-negligible probability over uniformly random $w$ and $A$'s internal randomness, then $A$ contradicts security of $h$.

We argue that $w \neq \widetilde{w}$ with non-negligible probability by appealing to the security of $i\mathcal{O}$. First, we note that with overwhelming probability over $w$, there indeed exists a $w' \neq w$ such that $h(w) = h(w')$. Consider any such pair $(w, w')$, and consider running $A(w)$ and $A(w')$. In both instances, the string $x = h(w) = h(w')$ computed is the same, meaning $\mathcal{E} = R_0(x)$ is the same. Let $C$ and $C'$ be the circuit "$C$" computed in step 2 of $A$. Since they are both witnesses for the same

Total-Learn instance $\mathcal{E}$, they are functionally equivalent. Thus the "$\widetilde{C}$" circuits computed (call them $\widetilde{C}$ and $\widetilde{C}'$) are $2^{-n^\delta}$-indistinguishable to $2^{n^\delta}$-time algorithms. Since $R_2$ is a subexponential-time algorithm, it follows that $p := \Pr[R_2(x, \tilde{C}) = w] \approx \Pr[R_2(x, \tilde{C}') = w]$. If $p \geq 1/2$, then we conclude that given $w'$, $A$ recovers $w \neq w'$ with probability at least $1/2$. If $p < 1/2$, then, given $w$, $A$ recovers some $w'' \neq w$, with probability at least $1/2$. In either case, we conclude that $A$ "breaks" $h$.

The assumption that the reduction is honest is used implicitly to ensure that the security parameter fed to the $i\mathcal{O}$ is sufficiently large. If the reduction is non-honest, the argument can be modified with a case analysis: in step 2, if $|C| < n^\delta$, we can brute force equivalent circuits in time $O(2^{n^{2\delta}})$ to find a canonical (the lexicographically first equivalent) circuit $\widetilde{C}$; in that case $R_2(x, \widetilde{C})$ is (statistically) identically distributed regardless of which witness $w$ we started with. The rest of the argument is the same as above. □

The ensuing corollaries settle the non-NP-hardness items of Theorem 2 and Theorem 1 respectively.

**Corollary 44.** *Assume that*

1. (CRYPTO) *there exists subexponentially-secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal PPS.*

*Then there exists a polynomial $g$ such that for every polynomial $s$ and $\varepsilon \leq 1/2$, $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ is not NP-hard under randomized subexponential-time Levin reductions.*

*Proof.* By Lemma 31, the two assumptions imply that subexponentially-secure one-way functions exist. The conclusion follows from Theorem 43. □

**Corollary 45.** *Assume that*

1. (CRYPTO) *there exists subexponentially-secure $i\mathcal{O}$, and*
2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal PPS.*

*Then there exists a polynomial $g$ such that for every polynomial $s$ and $\varepsilon \leq 1/2$, $\mathsf{Gap}_{g,\varepsilon}\mathsf{ImpMCSP}[s]$ is not NP-hard under randomized subexponential-time Levin reductions.*

*Proof.* The proof of Theorem 43 only uses properties of yes-instances of $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ and the fact that taking a witness to a yes-instance and applying $i\mathcal{O}$ to it yields a witness that the instance is not a no-instance. The yes-instances of $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$ and $\mathsf{Gap}_{g,\varepsilon}\mathsf{ImpMCSP}[s]$ are identical, and the fact also holds for $\mathsf{Gap}_{g,\varepsilon}\mathsf{ImpMCSP}[s]$. The corollary thus holds by the same proof as Theorem 43. □

# 5 Proof of Theorem 3

We re-state Theorem 3 below.

**Theorem 46** (Synergy between $i\mathcal{O}$ and "no optimal PPS")**.** *Assume*

1. (CRYPTO) *there exists subexponentially secure $i\mathcal{O}$, and*

2. (PROOF COMPLEXITY) *there is no infinitely-often subexponentially-optimal proof system.*

*Then all of the following hold:*

1. $\mathsf{NP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$,
2. $\mathsf{promise\text{-}BPP} \subseteq \mathsf{promise\text{-}TIME}[n^{\mathsf{poly}(\log n)}]$,
3. *subexponentially-secure one-way functions exist,*
4. *public-key encryption exists, and*
5. *if* $\mathsf{promise\text{-}BPP} = \mathsf{promise\text{-}P}$ *then* $\mathsf{NP} \cap \mathsf{coNP} \not\subseteq \mathsf{io\text{-}SIZE}[2^{n^{o(1)}}]$.

*Proof.* Items (1) and (2) follow from Lemma 29 and the fact that $i\mathcal{O}$ implies WE. Item (3) was proved in Lemma 31. Item (4) follows from the $i\mathcal{O}$ assumption and item (3) by [SW14]. Finally, observe that by Lemma 16 and Lemma 17, our assumptions imply that there exists a subexponentially-secure NIWI with a randomized polynomial-time verifier. If $\mathsf{promise\text{-}BPP} = \mathsf{promise\text{-}P}$ then we can derandomize it to a NIWI with a deterministic verifier. Item (5) thus follows from Corollary 42. □

# 6 Strong Cryptography and NP-hardness of Learning

## 6.1 Strengthenings of Witness Encryption

**Definition 47** (Extractable Witness Encryption [Gol+13])**.** A witness encryption scheme as in Definition 14 is *extractable* if there exists a PPT extractor algorithm $E$ such that for any polynomial $q$, $q(n)$-size adversary circuit $A$, and $x \in \{0,1\}^n$, if

$$\Pr_{b \sim \mathcal{U},\, \mathsf{Enc}}[A(x, \mathsf{Enc}(x,b)) = b] \geq \frac{1}{2} + \frac{1}{q(n)},$$

then

$$\Pr_{E}[\, E(x,A) = w \;\text{ s.t. }\; (x,w) \in R_L \,] \geq \frac{2}{3}.$$

The notion of *public verifiability*, defined below, means that whether a given string is a "valid" ciphertext or not can be checked efficiently, without any knowledge of secret information (in this case, a witness). On the other hand, *private verifiability* only requires that validity can be checked by a party in possession of a witness. We are not aware of any work studying these strengthenings of witness encryption previously. However, a number of papers have studied similar notions in the context of public-key encryption and generalizations thereof (see, for instance, [LCL10; Nie+13; BBH06; BMW05]). We mention that a number of well-known candidate public-key encryption schemes have the analogous properties.

**Definition 48** (Publicly Verifiable Witness Encryption)**.** A witness encryption scheme as in Definition 14 is *publicly verifiable* if $\{(x, ct) \mid ct \in \mathsf{range}(\mathsf{Enc}(x, -; -))\} \in \mathsf{P}$.

**Definition 49** (Privately Verifiable Witness Encryption)**.** A witness encryption scheme as in Definition 14 is *privately verifiable* if, for all $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $w \in \{0,1\}^{\mathsf{poly}(n)}$ with $(x,w) \in R_L$, and $z \notin \mathsf{range}(\mathsf{Enc}(x, -; -))$,

$$\mathsf{Dec}(x, w, z) = \bot.$$

**Definition 50** (Full-support Witness Encryption)**.** An $\ell$-length witness encryption scheme as in Definition 14 is *full-support* if for all $n \in \mathbb{N}$ and $x \in \{0,1\}^n$, $\mathsf{range}(\mathsf{Enc}(x, -; -)) = \{0,1\}^{\ell(n)}$.

## 6.2 Characterizing Strong WE by NP-hardness of Learning

In this subsection, we prove the following characterizations of strong forms of witness encryption by NP-hardness of learning problems.

**Theorem 51.** *The following are equivalent.*

- (HALF-LEVIN NP-HARDNESS OF Total-Learn) *There exist a polynomial $s$ and a pair of machines $(f, g)$ such that, for all polynomial $G, \varepsilon^{-1}$, $(f, g)$ is an honest deterministic half-Levin reduction from* SAT *to* $\mathsf{Total\text{-}Learn}_{G,\varepsilon}[s]$*;*
- *full-support witness encryption exists;*
- *publicly verifiable witness encryption exists.*

*Proof.* The theorem follows from Lemma 55 (Item 2), Lemma 56 (Item 2), and Lemma 57. □

**Theorem 52.** *The following are equivalent.*

- (LEVIN NP-HARDNESS OF CGL) *There exist a polynomial $s$ and a tuple of machines $(f, g, h)$ such that, for all polynomial $G, \varepsilon^{-1}$, $(f, g, h)$ is an honest Levin reduction from* SAT *to* $\mathsf{CGL}_{G,\varepsilon}[s]$ *in which $f$ and $g$ are deterministic and $h$ may be randomized.*
- *Extractable witness encryption exists.*

*Proof.* The theorem follows from Lemma 55 (Item 1) and Lemma 56 (Item 1). □

We also consider a variant of Total-Learn, which we denote Extension-Learn. We will show below that *privately* verifiable witness encryption implies the NP-hardness of this problem.

**Definition 53** (Extension-Learn)**.** For $s, g, \varepsilon^{-1} : \mathbb{N} \to \mathbb{N}$, an instance of the promise problem $\mathsf{Extension\text{-}Learn}_{g,\varepsilon}[s]$ at input length $n \in \mathbb{N}$ consists of a $\mathsf{poly}(n)$-size circuit $\mathcal{E}$ sampling a joint distribution $(X, B)$ supported over $\{0,1\}^n \times \{0,1\}$.

- $\mathcal{E}$ is a yes-instance of $\mathsf{Extension\text{-}Learn}_{g,\varepsilon}[s]$ if there exists a circuit $C$ of size at most $s(n)$ such that
$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1$$
and for all $x \notin \mathsf{supp}(X)$, $C(x) = 0$.
- $\mathcal{E}$ is a no-instance of $\mathsf{Extension\text{-}Learn}_{g,\varepsilon}[s]$ if, for every circuit $C$ of size at most $g(n)$,
$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] < 1/2 + \varepsilon(n).$$

**Theorem 54.** *The first item implies the second in the following.*

- *Privately verifiable witness encryption exists;*
- (HALF-LEVIN NP-HARDNESS OF Extension-Learn) *there exist a polynomial $s$ and a pair of machines $(f, g)$ such that, for all polynomial $G, \varepsilon^{-1}$, $(f, g)$ is an honest deterministic half-Levin reduction from* SAT *to* $\mathsf{Extension\text{-}Learn}_{G,\varepsilon}[s]$*.*

The theorems are proved in the lemmas below.

**Lemma 55.** *Assume there exist a polynomial $s$ and a tuple of machines $(f, g)$ such that, for all polynomial $G, \varepsilon^{-1}$, $(f, g)$ is an honest deterministic half-Levin reduction from* SAT *to* $\mathsf{CGL}_{G,\varepsilon}[s]$*. Then, witness encryption exists. Morevover,*

1. *if there is a third (possibly randomized) machine $h$ making $(f, g, h)$ Levin, then the witness encryption scheme is extractable;*
2. *if the reduction is to* $\mathsf{Total\text{-}Learn}_{G,\varepsilon}[s]$*, then the witness encryption scheme is full-support.*

*Proof.* Let $(f, g)$ be a half-Levin reduction from SAT to $\mathsf{CGL}[s]$ as described in the statement of the lemma. We define a witness encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ as in [GK25], as follows.

- $\mathsf{Enc}(\varphi, a; r)$ simulates $f(\varphi) = \mathcal{E}$, samples $(x, b) = \mathcal{E}(r)$, and then outputs $(x, a \oplus b)$.
- $\mathsf{Dec}(\varphi, w, ct = (x, c))$ simulates $g(\varphi, w) = C$, and then outputs $C(x) \oplus c$.

Correctness and standard (non-extractable) security of $(\mathsf{Enc}, \mathsf{Dec})$ follow from [GK25, Lemma 54]. We now demonstrate the additional properties stated.

**Item (1).** Suppose the reduction is full-Levin. That is, it includes a third machine $h$ as in Definition 5. We define an extractor algorithm $E$ as follows.

> On input $(\varphi, A)$ where $A$ is a $q(n)$-size circuit and $\varphi \in \{0,1\}^n$, for $a \in \{0,1\}$, consider the circuits $C_a$ defined as $C_a(x) = A(\varphi, x, a) \oplus a$. Simulate $f(\varphi) = \mathcal{E}$. By empirical estimation, taking $(q(n) \cdot n)^2$ random samples from $\mathcal{E}$, determine the bit $a' \in \{0,1\}$ maximizing $\Pr_{(x,b)\sim\mathcal{E}}[C_{a'}(x) = b]$. Simulate $w \leftarrow h(\varphi, C_{a'})$, and output $w$.

We now argue for the correctness of the extractor $E$. Let $A$ be a $q(n)$-size circuit such that for all $\varphi \in \{0,1\}^n$,

$$\Pr_{a\sim\mathcal{U},\ \mathsf{Enc}}[A(\varphi, \mathsf{Enc}(\varphi, a)) = a] \geq \frac{1}{2} + q^{-1}(n).$$

Re-writing according to the definition of $\mathsf{Enc}$, letting $\mathcal{E} = f(\varphi)$,

$$\Pr_{a\sim\mathcal{U},\ (x,b)\sim\mathcal{E}}[A(\varphi, x, a \oplus b) = a] \geq \frac{1}{2} + q^{-1}(n).$$

Since $a \oplus b$ is a uniformly random bit when $a \sim \mathcal{U}$,

$$\Pr_{a\sim\mathcal{U},\ (x,b)\sim\mathcal{E}}[A(\varphi, x, a) = a \oplus b] \geq \frac{1}{2} + q^{-1}(n).$$

Then, for some fixed choice of $a$,

$$\begin{aligned}\Pr_{(x,b)\sim\mathcal{E}}[C_a(x) = b] &= \Pr_{(x,b)\sim\mathcal{E}}[A(\varphi, x, a) \oplus a = b] \\ &= \Pr_{(x,b)\sim\mathcal{E}}[A(\varphi, x, a) = a \oplus b] \\ &\geq \frac{1}{2} + q^{-1}(n).\end{aligned}$$

Observe that $E$ sets $a'$ with $\Pr_{(x,b)\sim\mathcal{E}}[C_{a'}(x) = b] \geq (1 + q^{-1}(n))/2$ unless the empirical estimation step is far from correct, which occurs with probability at most $2^{-\Omega(n)}$, by a standard concentration bound. When this happens, $C_{a'}$ is a witness for $\mathcal{E} = f(\varphi)$ not being a no-instance of $\mathsf{CGL}_{p,p^{-1}}[s]$ for some $p = \mathsf{poly}(q)$, in which case $h(\varphi, C_{a'})$ outputs a witness $w$ for $\varphi \in \mathsf{SAT}$ with probability greater than $3/4$. Therefore, $E$ has the extractor property of Definition 47.

**Item (2).** Suppose the reduction $(f, g)$ maps instances of $\mathsf{SAT}$ to instances of $\mathsf{Total\text{-}Learn}$. By definition of $\mathsf{Total\text{-}Learn}$, for all $n \in \mathbb{N}$ and $\varphi \in \{0,1\}^n$, $f(\varphi) = \mathcal{E}$ samples a joint distribution $(X, B)$, where $X$ has full support over $\{0,1\}^{m(n)}$, for some polynomial $m$. Then, for any string $(x, b) \in \{0,1\}^{m(n)+1}$, there exists a pair $(a, r)$ such that $\mathsf{Enc}(\varphi, a; r) = (x, b)$: namely, let $r$ be such that $\mathcal{E}(r) = (x, b')$ for some $b' \in \{0,1\}$, and let $a = b' \oplus b$. It follows that $\mathsf{range}(\mathsf{Enc}(\varphi, -; -)) = \{0,1\}^{m(n)+1}$, as desired. □

**Lemma 56.** *Assume witness encryption exists. Then, there exist a polynomial $s$ and a tuple of machines $(f, g)$ such that, for all polynomial $G, \varepsilon^{-1}$, $(f, g)$ is an honest half-Levin reduction from $\mathsf{SAT}$ to $\mathsf{CGL}_{G,\varepsilon}[s]$. Morevover,*

1. *if the witness encryption scheme is extractable, then there is a randomized third machine $h$ making the reduction full-Levin;*
2. *if the witness encryption scheme is full-support, then the reduction is to $\mathsf{Total\text{-}Learn}_{g,\varepsilon}[s]$;*
3. *if the witness encryption scheme is privately verifiable, then the reduction is to $\mathsf{Extension\text{-}Learn}_{g,\varepsilon}[s]$.*

*Proof.* Let $(\mathsf{Enc}, \mathsf{Dec})$ be a witness encryption scheme for $\mathsf{SAT}$. We define a half-Levin reduction $(f, g)$ as in [GK25], as follows.

- $f(\varphi)$ outputs $\mathcal{E}$, which samples $b \sim \mathcal{U}$, $r \sim \mathcal{U}_{\mathsf{poly}(n)}$, lets $x = \mathsf{Enc}(\varphi, b; r)$, and outputs $(x, b)$.
- $g(\varphi, w)$ outputs the circuit $C$ such that $C(x) = \mathsf{Dec}(\varphi, w, x) = b$ if $b \neq \perp$ and $C(x) = 0$ if $b = \perp$.

The correctness and soundness of $(\mathsf{Enc}, \mathsf{Dec})$ follow from [GK25, Lemma 52]. Also note that the reduction is honest, since $\mathcal{E}$ has a description of $\varphi$ hard-wired. We now argue for the additional properties.

**Item (1).** Suppose that $(\mathsf{Enc}, \mathsf{Dec})$ is extractable. That is, there exists an extractor machine $E$ as in Definition 47. We simply define $h$ as $E$.

To see that $h$ makes $(f, g, h)$ into a Levin reduction, consider any $\mathsf{SAT}$-instance $\varphi \in \{0,1\}^n$ and circuit $C$ of size at most $q(n)$ such that

$$\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] \geq \frac{1}{2} + \frac{1}{q(n)},$$

where $\mathcal{E} = f(\varphi)$, and $q$ is some polynomial. By definition of $f$,

$$\Pr_{b\sim\mathcal{U},\mathsf{Enc}}[C(\mathsf{Enc}(\varphi, b)) = b] \geq \frac{1}{2} + \frac{1}{q(n)}.$$

Then, by the guarantee of $E = h$,

$$\Pr_h[\ h(\varphi, C) = w \text{ s.t. } \varphi(w) = 1\ ] \geq \frac{2}{3},$$

as desired.

**Item (2).** Suppose $(\mathsf{Enc}, \mathsf{Dec})$ is full-support and $\ell$-length, for some polynomial $\ell$. That is, for any $\varphi \in \{0,1\}^n$, $\mathsf{range}(\mathsf{Enc}(\varphi, -; -)) = \{0,1\}^{\ell(n)}$. It is easy to see that $\mathcal{E} = f(\varphi)$ is supported over a distribution $(X, B)$ with $\mathsf{supp}(X) = \mathsf{range}(\mathsf{Enc}(\varphi, -; -)) = \{0,1\}^{\ell(n)}$. This completes the proof.

**Item (3).** Suppose $(\mathsf{Enc}, \mathsf{Dec})$ is privately verifiable. Consider $\varphi \in \mathsf{SAT} \cap \{0,1\}^n$ with witness $w$. For any $x \notin \mathsf{supp}(X)$, where $\mathcal{E} = f(\varphi)$ samples $(X, B)$, we have $x \notin \mathsf{range}(\mathsf{Enc}(\varphi, -; -))$. By private verifiability, $\mathsf{Dec}(\varphi, w, x) = \bot$, and hence $C(x) = 0$, where $C = g(\varphi, w)$. Thus, the reduction produces witnesses for $\mathsf{Extension\text{-}Learn}_{G,\varepsilon}[s]$, as desired. □

**Lemma 57.** *Publicly verifiable witness encryption exists if and only if full-support witness encryption exists.*

*Proof.* It is easy to see that a full-support witness encryption scheme is trivially already publicly verifiable.

To see the other direction, let $(\mathsf{Enc}, \mathsf{Dec})$ be a publicly verifiable $\ell$-length witness encryption scheme. We define a modified scheme as follows. For $\varphi \in \{0,1\}^n$ with witness $w \in \{0,1\}^{\mathsf{poly}(n)}$ (if any), for some negligible function $\alpha : \mathbb{N} \to [0,1]$,

> $\mathsf{Enc}'(\varphi, 1)$ is the same as $\mathsf{Enc}(\varphi, 1)$.
>
> $\mathsf{Enc}'(\varphi, 0)$ outputs $\mathsf{Enc}(\varphi, 0)$ with probability $1 - \alpha(n)$. With the remaining probability $\alpha(n)$, it samples $z \sim \mathcal{U}_{\ell(n)}$ and checks if $z \in \mathsf{range}(\mathsf{Enc}(\varphi, -; -))$. If not, it outputs $z$; if so, it outputs $\mathsf{Enc}(\varphi, 0)$.
>
> $\mathsf{Dec}'(\varphi, w, z)$ outputs $\mathsf{Dec}(\varphi, w, z)$ if $z \in \mathsf{range}(\mathsf{Enc}(\varphi, -; -))$, and outputs 0 otherwise.

It is easy to see that $(\mathsf{Enc}', \mathsf{Dec}')$ is full-support. Also note that

$$\Pr[\mathsf{Dec}'(\varphi, w, \mathsf{Enc}'(\varphi, 1)) = 1] = \Pr[\mathsf{Dec}(\varphi, w, \mathsf{Enc}(\varphi, 1)) = 1] = 1$$

and

$$\begin{aligned}&\Pr[\mathsf{Dec}'(\varphi, w, \mathsf{Enc}'(\varphi, 0)) = 0] \\ &\quad \geq 1 - \Pr[\mathsf{Dec}(\varphi, w, \mathsf{Enc}(\varphi, 0)) \neq 0] - \Pr[\mathsf{Dec}'(\varphi, w, z) \neq 0 \mid z \notin \mathsf{range}(\mathsf{Enc}(\varphi, -; -))] \\ &\quad = 1,\end{aligned}$$

so $(\mathsf{Enc}', \mathsf{Dec}')$ satisfies correctness.

For any adversary $A$ and polynomial $q$, if

$$\Pr_{b \sim \mathcal{U},\ \mathsf{Enc}',\ A}[A(\mathsf{Enc}'(\varphi, b)) = b] \geq \frac{1}{2} + q^{-1}(n),$$

then

$$\Pr_{b\sim\mathcal{U},\,\mathsf{Enc},\,A}[A(\mathsf{Enc}(\varphi,b))=b] \geq \frac{1}{2}+q^{-1}(n)-\alpha(n)$$
$$\geq \frac{1}{2}+\frac{q^{-1}(n)}{2}.$$

The security of $(\mathsf{Enc}', \mathsf{Dec}')$ follows. This completes the proof. □

### 6.3 Impossibility of Indistinguishability Obfuscation and Strong Witness Encryption

In this section, we show a dichotomy in cryptography: assuming only worst-case hardness of NP, either indistinguishability obfuscation does not exist, or strong (i.e., privately verifiable and extractable) witness encryption does not exist. If one believes in the hardness of NP and the existence of $i\mathcal{O}$, then one should believe that witness encryption cannot simultaneously have the features of extractability and private (or public) verifiability.

**Theorem 58.** *Assume that* $\mathsf{NP} \not\subseteq \mathsf{BPP}$. *At most one of the following holds.*

- *$i\mathcal{O}$ exists;*
- *privately verifiable extractable witness encryption exists.*

We will prove the theorem via the following lemma.

**Lemma 59.** *Assume that $i\mathcal{O}$ exists and that* $\mathsf{NP} \not\subseteq \mathsf{BPP}$. *Then for any polynomial $s$, there exist polynomial $G, \varepsilon^{-1}$ such that* $\mathsf{Extension\text{-}Learn}_{G,\varepsilon}[s]$ *is not* NP*-hard under randomized polynomial-time honest Levin reductions.*

*Proof Sketch.* We apply the proof idea of [MP24]. We refer the reader to Theorem 43 for an overview. The only differences in this case are that we would like to apply the proof for the "polynomial regime" (ie. assuming only polynomially-secure $i\mathcal{O}$ and worst-case hardness of NP against BPP, as in [MP24, Theorem 1.2])[11] and that the reduction is to Extension-Learn rather than Total-Learn. For the latter modification, we observe that, for any yes-instance $x$ of $L$ (as in the proof of Theorem 43), for any witnesses $w$, $w'$ for $x \in L$, the corresponding witnesses $C = R_1(x, w)$ and $C' = R_1(x, w')$ are still *functionally equivalent circuits*, by definition of Extension-Learn. Thus, the obfuscations of these circuits are computationally indistinguishable, as desired. □

*Proof of Theorem 58.* Assume $\mathsf{NP} \not\subseteq \mathsf{BPP}$ and $i\mathcal{O}$ exists. Suppose privately verifiable extractable witness encryption also exists. By Lemma 56, Items (1) and (3), there exist a polynomial $s$ and a tuple of machines $(f, g, h)$ such that, for all polynomial $G, \varepsilon^{-1}$, $(f, g, h)$ is an honest Levin reduction from SAT to $\mathsf{Extension\text{-}Learn}_{G,\varepsilon}[s]$. Applying Lemma 59 yields a contradiction. □

[11] We remark that although [MP24, Theorem 1.2]) states the assumption as $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$, $\mathsf{NP} \not\subseteq \mathsf{BPP}$ is actually sufficient: by [Kom+22], one obtains *infinitely often* one-way functions, which the assumed NP-hardness reduction together with existence of $i\mathcal{O}$ would rule out.

## 7 Open Questions

Are the two assumptions used in our main theorems necessary? Can one conditionally (for instance, under cryptographic assumptions) rule out NP-hardness Levin reductions for a polynomial-gap version of CGL? Can one *unconditionally* show that (the non-gap version of) ImpMCSP, or Total-Learn, is NP-hard?

**Acknowledgements.** We thank Shuichi Hirahara for sharing an idea about using zero-knowledge proofs to certify domain membership in partial functions. We also thank Ran Canetti and Yiding Zhang for helpful discussions about cryptography.

## References

[ABX08] Benny Applebaum, Boaz Barak, and David Xiao. "On Basing Lower-Bounds for Learning on Worst-Case Assumptions". In: *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*. IEEE Computer Society, 2008, pp. 211–220. DOI: 10.1109/FOCS.2008.35.

[AH19] Eric Allender and Shuichi Hirahara. "New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems". In: *ACM Trans. Comput. Theory* 11.4 (2019), 27:1–27:27. DOI: 10.1145/3349616.

[AHK17] Eric Allender, Dhiraj Holden, and Valentine Kabanets. "The Minimum Oracle Circuit Size Problem". In: *Comput. Complex.* 26.2 (2017), pp. 469–496. DOI: 10.1007/S00037-016-0124-0.

[Ale+01] Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. "Minimum Propositional Proof Length Is NP-Hard to Linearly Approximate". In: *J. Symb. Log.* 66.1 (2001), pp. 171–191. DOI: 10.2307/2694916.

[Bab+93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. "BPP Has Subexponential Time Simulations Unless EXPTIME has Publishable Proofs". In: *Comput. Complex.* 3 (1993), pp. 307–318. DOI: 10.1007/BF01275486.

[Bar+01] Boaz Barak et al. "On the (Im)possibility of Obfuscating Programs". In: *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*. Ed. by Joe Kilian. Lecture Notes in Computer Science. Springer, 2001, pp. 1–18. DOI: 10.1007/3-540-44647-8_1.

[Bar+12] Boaz Barak et al. "On the (Im)possibility of Obfuscating Programs". In: *J. ACM* 59.2 (2012), 6:1–6:48. DOI: 10.1145/2160158.2160159.

[BBH06] Dan Boneh, Xavier Boyen, and Shai Halevi. "Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles". In: *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*. Ed. by David Pointcheval. Lecture Notes in Computer Science. Springer, 2006, pp. 226–243. DOI: 10.1007/11605805_15.

[Bit+14] Nir Bitansky et al. “The Impossibility of Obfuscation with Auxiliary Input or a Universal Simulator”. In: *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*. Ed. by Juan A. Garay and Rosario Gennaro. Lecture Notes in Computer Science. Springer, 2014, pp. 71–89. DOI: 10.1007/978-3-662-44381-1_5.

[BKM09] Olaf Beyersdorff, Johannes Köbler, and Jochen Messner. “Nondeterministic Functions and the Existence of Optimal Proof Systems”. In: *Theor. Comput. Sci.* 410.38–40 (Sept. 2009), pp. 3839–3855. DOI: 10.1016/j.tcs.2009.05.021.

[BKM11] Olaf Beyersdorff, Johannes Köbler, and Sebastian Müller. “Proof Systems that Take Advice”. In: *Inf. Comput.* 209.3 (2011), pp. 320–332. DOI: 10.1016/J.IC.2010.11.006.

[BMW05] Xavier Boyen, Qixiang Mei, and Brent Waters. “Direct Chosen Ciphertext Security from Identity-Based Techniques”. In: *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005*. Ed. by Vijay Atluri, Catherine Meadows, and Ari Juels. ACM, 2005, pp. 320–329. DOI: 10.1145/1102120.1102162.

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. “Derandomization in Cryptography”. In: *SIAM J. Comput.* 37.2 (2007), pp. 380–400. DOI: 10.1137/050641958.

[BP15] Nir Bitansky and Omer Paneth. “ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation”. In: *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. Lecture Notes in Computer Science. Springer, 2015, pp. 401–427. DOI: 10.1007/978-3-662-46497-7_16.

[Buh+00] Harry Buhrman, Stephen A. Fenner, Lance Fortnow, and Dieter van Melkebeek. “Optimal Proof Systems and Sparse Sets”. In: *STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science, Lille, France, February 2000, Proceedings*. Ed. by Horst Reichel and Sophie Tison. Lecture Notes in Computer Science. Springer, 2000, pp. 407–418. DOI: 10.1007/3-540-46541-3_34.

[Can+24] Ran Canetti, Claudio Chamon, Eduardo R. Mucciolo, and Andrei E. Ruckenstein. “Towards General-Purpose Program Obfuscation via Local Mixing”. In: *Theory of Cryptography - 22nd International Conference, TCC 2024, Milan, Italy, December 2-6, 2024, Proceedings, Part IV*. Ed. by Elette Boyle and Mohammad Mahmoody. Lecture Notes in Computer Science. Springer, 2024, pp. 37–70. DOI: 10.1007/978-3-031-78023-3_2.

[CFM14] Yijia Chen, Jörg Flum, and Moritz Müller. “Hard Instances of Algorithms and Proof Systems”. In: *ACM Trans. Comput. Theory* 6.2 (2014), 7:1–7:25. DOI: 10.1145/2601336.

[Che+24] Lijie Chen, Ce Jin, Rahul Santhanam, and Ryan Williams. “Constructive Separations and Their Consequences”. In: *TheoretiCS* 3 (2024). DOI: 10.46298/THEORETICS.24.3.

[CK07] Stephen A. Cook and Jan Krajíček. “Consequences of the Provability of NP $\subseteq$ P/poly”. In: *J. Symb. Log.* 72.4 (2007), pp. 1353–1371. DOI: 10.2178/JSL/1203350791.

[Coo71] Stephen A. Cook. "The Complexity of Theorem-Proving Procedures". In: *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*. Ed. by Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman. ACM, 1971, pp. 151–158. DOI: 10.1145/800157.805047.

[CR79] Stephen A. Cook and Robert A. Reckhow. "The Relative Efficiency of Propositional Proof Systems". In: *J. Symb. Log.* 44.1 (1979), pp. 36–50. DOI: 10.2307/2273702.

[DN07] Cynthia Dwork and Moni Naor. "Zaps and Their Applications". In: *SIAM J. Comput.* 36.6 (2007), pp. 1513–1543. DOI: 10.1137/S0097539703426817.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. "Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract)". In: *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*. IEEE Computer Society, 1990, pp. 308–317. DOI: 10.1109/FSCS.1990.89549.

[Gar+13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. "Witness Encryption and its Applications". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM, 2013, pp. 467–476. DOI: 10.1145/2488608.2488667.

[Gar+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. "Candidate Indistinguishability Obfuscation and Functional Encryption for All Circuits". In: *SIAM J. Comput.* 45.3 (2016), pp. 882–929. DOI: 10.1137/14095772X.

[Gar+17] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. "On the Implausibility of Differing-Inputs Obfuscation and Extractable Witness Encryption with Auxiliary Input". In: *Algorithmica* 79.4 (2017), pp. 1353–1373. DOI: 10.1007/S00453-017-0276-6.

[Gen+15] Craig Gentry, Allison Bishop Lewko, Amit Sahai, and Brent Waters. "Indistinguishability Obfuscation from the Multilinear Subgroup Elimination Assumption". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. Ed. by Venkatesan Guruswami. IEEE Computer Society, 2015, pp. 151–170. DOI: 10.1109/FOCS.2015.19.

[GK05] Shafi Goldwasser and Yael Tauman Kalai. "On the Impossibility of Obfuscation with Auxiliary Input". In: *46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005, Pittsburgh, PA, USA, October 23-25, 2005, Proceedings*. IEEE Computer Society, 2005, pp. 553–562. DOI: 10.1109/SFCS.2005.60.

[GK25] Halley Goldberg and Valentine Kabanets. "Witness Encryption and NP-Hardness of Learning". In: *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*. Ed. by Srikanth Srinivasan. Vol. 339. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 34:1–34:43. DOI: 10.4230/LIPICS.CCC.2025.34.

[Gla+14] Christian Glaßer, Andrew Hughes, Alan L. Selman, and Nils Wisiol. "Disjoint NP-Pairs and Propositional Proof Systems". In: *SIGACT News* 45.4 (2014), pp. 59–75. DOI: 10.1145/2696081.2696095.

[GO94] Oded Goldreich and Yair Oren. "Definitions and Properties of Zero-Knowledge Proof Systems". In: *J. Cryptol.* 7.1 (1994), pp. 1–32. DOI: 10.1007/BF00195207.

[Gol+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. "Overcoming the Worst-Case Curse for Cryptographic Constructions". In: *IACR Cryptol. ePrint Arch.* (2013), p. 229.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "Non-interactive Zaps and New Techniques for NIZK". In: *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*. Ed. by Cynthia Dwork. Lecture Notes in Computer Science. Springer, 2006, pp. 97–111. DOI: 10.1007/11818175_6.

[HI25] Shuichi Hirahara and Rahul Ilango. "NP-hardness of the Minimum Circuit Size Problem from Well-Studied Assumptions". In: *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*. IEEE, 2025, pp. 1648–1664. DOI: 10.1109/FOCS63196.2025.00087.

[Hir22] Shuichi Hirahara. "NP-Hardness of Learning Programs and Partial MCSP". In: *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 2022, pp. 968–979. DOI: 10.1109/FOCS54457.2022.00095.

[HIR25] Yizhi Huang, Rahul Ilango, and Hanlin Ren. "NP-Hardness of Approximating Meta-Complexity: A Cryptographic Approach". In: *SIAM J. Comput.* 54.4 (2025), pp. 819–886. DOI: 10.1137/23M1608483.

[HP15] John M. Hitchcock and Aduri Pavan. "On the NP-Completeness of the Minimum Circuit Size Problem". In: *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, Bangalore, India, December 16-18, 2015*. Ed. by Prahladh Harsha and G. Ramalingam. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 236–245. DOI: 10.4230/LIPICS.FSTTCS.2015.236.

[IKV23] Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. "Synergy Between Circuit Obfuscation and Circuit Minimization". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, Atlanta, Georgia, USA, September 11-13, 2023*. Ed. by Nicole Megow and Adam D. Smith. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 31:1–31:21. DOI: 10.4230/LIPICS.APPROX/RANDOM.2023.31.

[Ila20] Rahul Ilango. "Approaching MCSP from Above and Below: Hardness for a Conditional Variant and $AC^0[p]$". In: *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, Seattle, Washington, USA, January 12-14, 2020*. Ed. by Thomas Vidick. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 34:1–34:26. DOI: 10.4230/LIPICS.ITCS.2020.34.

[Ila23] Rahul Ilango. "SAT Reduces to the Minimum Circuit Size Problem with a Random Oracle". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 733–742. DOI: 10.1109/FOCS57990.2023.00048.

[Ila25] Rahul Ilango. "Gödel in Cryptography: Effectively Zero-Knowledge Proofs for NP with No Interaction, No Setup, and Perfect Soundness". In: *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*. IEEE, 2025, pp. 1100–1127. DOI: 10.1109/FOCS63196.2025.00058.

[ILO20] Rahul Ilango, Bruno Loff, and Igor C. Oliveira. "NP-Hardness of Circuit Minimization for Multi-Output Functions". In: *35th Computational Complexity Conference, CCC 2020, Saarbrücken, Germany (Virtual Conference), July 28-31, 2020*. Ed. by Shubhangi Saraf. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 22:1–22:36. DOI: 10.4230/LIPICS.CCC.2020.22.

[IW97] Russell Impagliazzo and Avi Wigderson. "P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma". In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*. Ed. by Frank Thomson Leighton and Peter W. Shor. ACM, 1997, pp. 220–229. DOI: 10.1145/258533.258590.

[Jin+24] Zhengzhong Jin, Yael Kalai, Alex Lombardi, and Vinod Vaikuntanathan. "SNARGs under LWE via Propositional Proofs". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM, 2024, pp. 1750–1757. DOI: 10.1145/3618260.3649770.

[Jin+25] Zhengzhong Jin, Yael Tauman Kalai, Alex Lombardi, and Surya Mathialagan. "Universal SNARGs for NP from Proofs of Correctness". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*. Ed. by Michal Koucký and Nikhil Bansal. ACM, 2025, pp. 933–943. DOI: 10.1145/3717823.3718104.

[JJ22] Abhishek Jain and Zhengzhong Jin. "Indistinguishability Obfuscation via Mathematical Proofs of Equivalence". In: *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 2022, pp. 1023–1034. DOI: 10.1109/FOCS54457.2022.00100.

[JLS26] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability Obfuscation from Well-Founded Assumptions". In: *J. ACM* 73.1 (2026), 1:1–1:30. DOI: 10.1145/3785007.

[Kab01] Valentine Kabanets. "Easiness Assumptions and Hardness Tests: Trading Time for Zero Error". In: *J. Comput. Syst. Sci.* 63.2 (2001), pp. 236–252. DOI: 10.1006/JCSS.2001.1763.

[Kar72] Richard M. Karp. "Reducibility Among Combinatorial Problems". In: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*. Ed. by Raymond E. Miller and James W. Thatcher. The IBM Research Symposia Series. Plenum Press, New York, 1972, pp. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.

[KC00] Valentine Kabanets and Jin-yi Cai. "Circuit Minimization Problem". In: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*. Ed. by F. Frances Yao and Eugene M. Luks. ACM, 2000, pp. 73–79. DOI: 10.1145/335305.335314.

[Kha22] Erfan Khaniki. "Nisan-Wigderson Generators in Proof Complexity: New Lower Bounds". In: *37th Computational Complexity Conference, CCC 2022, Philadelphia, PA, USA, July 20-23, 2022*. Ed. by Shachar Lovett. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 17:1–17:15. DOI: 10.4230/LIPICS.CCC.2022.17.

[Kha24] Erfan Khaniki. "Jump Operators, Interactive Proofs and Proof Complexity Generators". In: *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024, pp. 573–593. DOI: 10.1109/FOCS61266.2024.00044.

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. "Indistinguishability Obfuscation for Turing Machines with Unbounded Memory". In: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*. Ed. by Rocco A. Servedio and Ronitt Rubinfeld. ACM, 2015, pp. 419–428. DOI: 10.1145/2746539.2746614.

[KM98] Johannes Köbler and Jochen Messner. "Complete Problems for Promise Classes by Optimal Proof Systems for Test Sets". In: *Proceedings of the 13th Annual IEEE Conference on Computational Complexity, Buffalo, New York, USA, June 15-18, 1998*. IEEE Computer Society, 1998, pp. 132–140. DOI: 10.1109/CCC.1998.694599.

[KMT03] Johannes Köbler, Jochen Messner, and Jacobo Torán. "Optimal proof systems imply complete sets for promise classes". In: *Inf. Comput.* 184.1 (2003), pp. 71–92. DOI: 10.1016/S0890-5401(03)00058-0.

[Kom+22] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. "One-Way Functions and (Im)perfect Obfuscation". In: *SIAM J. Comput.* 51.6 (2022), pp. 1769–1795. DOI: 10.1137/15M1048549.

[KP89] Jan Krajíček and Pavel Pudlák. "Propositional Proof Systems, the Consistency of First Order Theories and the Complexity of Computations". In: *J. Symb. Log.* 54.3 (1989), pp. 1063–1079. DOI: 10.2307/2274765.

[Kra13] Jan Krajíček. "On the computational complexity of finding hard tautologies". In: *Bulletin of the London Mathematical Society* 46.1 (Oct. 2013), pp. 111–125. DOI: 10.1112/blms/bdt071. eprint: https://academic.oup.com/blms/article-pdf/46/1/111/6694021/bdt071.pdf.

[KST23] Caleb Koch, Carmen Strassle, and Li-Yang Tan. "Properly learning decision trees with queries is NP-hard". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 2383–2407. DOI: 10.1109/FOCS57990.2023.00146.

[KZ20] Benjamin Kuykendall and Mark Zhandry. "Towards Non-interactive Witness Hiding". In: *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part I*. Ed. by Rafael Pass and Krzysztof Pietrzak. Lecture Notes in Computer Science. Springer, 2020, pp. 627–656. DOI: 10.1007/978-3-030-64375-1_22.

[LCL10] Yujun Liu, Yonggang Cui, and Limin Liu. "A Publicly Verifiable Encryption Scheme with Short Public/Private Keys". In: *Wireless Algorithms, Systems, and Applications, 5th International Conference, WASA 2010, Beijing, China, August 15-17, 2010. Proceedings*. Ed. by Gopal Pandurangan, V. S. Anil Kumar, Gu Ming, Yunhao Liu, and Yingshu Li. Lecture Notes in Computer Science. Springer, 2010, pp. 261–265. DOI: 10.1007/978-3-642-14654-1_32.

[Lev73] Leonid A Levin. "Universal Sequential Search Problems". In: *Problemy Peredachi Informatsii* 9.3 (1973), pp. 115–116.

[Lin+16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. "Output-Compressing Randomized Encodings and Applications". In: *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*. Ed. by Eyal Kushilevitz and Tal Malkin. Lecture Notes in Computer Science. Springer, 2016, pp. 96–124. DOI: 10.1007/978-3-662-49096-9_5.

[Lin16] Huijia Lin. "Indistinguishability Obfuscation from Constant-Degree Graded Encoding Schemes". In: *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Lecture Notes in Computer Science. Springer, 2016, pp. 28–57. DOI: 10.1007/978-3-662-49890-3_2.

[Lu+24] Zhenjian Lu, Noam Mazor, Igor C. Oliveira, and Rafael Pass. "Lower Bounds on the Overhead of Indistinguishability Obfuscation". In: *Electron. Colloquium Comput. Complex.* TR24, TR24-146 (2024). ECCC: TR24-146.

[MDS25] Yaohua Ma, Chenxin Dai, and Elaine Shi. "Quasi-Linear Indistinguishability Obfuscation via Mathematical Proofs of Equivalence and Applications". In: *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part III*. Ed. by Serge Fehr and Pierre-Alain Fouque. Lecture Notes in Computer Science. Springer, 2025, pp. 157–186. DOI: 10.1007/978-3-031-91131-6_6.

[MP24] Noam Mazor and Rafael Pass. "Gap MCSP Is Not (Levin) NP-Complete in Obfustopia". In: *39th Computational Complexity Conference, CCC 2024, Ann Arbor, MI, USA, July 22-25, 2024*. Ed. by Rahul Santhanam. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, 36:1–36:21. DOI: 10.4230/LIPICS.CCC.2024.36.

[MW17] Cody D. Murray and R. Ryan Williams. "On the (Non) NP-Hardness of Computing Circuit Complexity". In: *Theory Comput.* 13.1 (2017), pp. 1–22. DOI: 10.4086/TOC.2017.V013A004.

[Nie+13] Juan Manuel González Nieto, Mark Manulis, Bertram Poettering, Jothi Rangasamy, and Douglas Stebila. "Publicly verifiable ciphertexts". In: *J. Comput. Secur.* 21.5 (2013), pp. 749–778. DOI: 10.3233/JCS-130473.

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs Randomness". In: *J. Comput. Syst. Sci.* 49.2 (1994), pp. 149–167. DOI: 10.1016/S0022-0000(05)80043-1.

[PS19] Ján Pich and Rahul Santhanam. "Why are Proof Complexity Lower Bounds Hard?" In: *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*. Ed. by David Zuckerman. IEEE Computer Society, 2019, pp. 1305–1324. DOI: 10.1109/FOCS.2019.00080.

[PV88] Leonard Pitt and Leslie G. Valiant. "Computational Limitations on Learning from Examples". In: *J. ACM* 35.4 (1988), pp. 965–984. DOI: 10.1145/48014.63140.

[Rom90] John Rompel. "One-Way Functions are Necessary and Sufficient for Secure Signatures". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. Ed. by Harriet Ortiz. ACM, 1990, pp. 387–394. DOI: 10.1145/100216.100269.

[RS22] Hanlin Ren and Rahul Santhanam. "A Relativization Perspective on Meta-Complexity". In: *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, Marseille, France (Virtual Conference), March 15-18, 2022*. Ed. by Petra Berenbrink and Benjamin Monmege. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 54:1–54:13. DOI: 10.4230/LIPICS.STACS.2022.54.

[Sad02] Zenon Sadowski. "On an Optimal Propositional Proof System and the Structure of Easy Subsets of TAUT". In: *Theor. Comput. Sci.* 288.1 (2002), pp. 181–193. DOI: 10.1016/S0304-3975(01)00155-4.

[SS20] Michael E. Saks and Rahul Santhanam. "Circuit Lower Bounds from NP-Hardness of MCSP Under Turing Reductions". In: *35th Computational Complexity Conference, CCC 2020, Saarbrücken, Germany (Virtual Conference), July 28-31, 2020*. Ed. by Shubhangi Saraf. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 26:1–26:13. DOI: 10.4230/LIPICS.CCC.2020.26.

[SW14] Amit Sahai and Brent Waters. "How to Use Indistinguishability Obfuscation: Deniable Encryption, and More". In: *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*. Ed. by David B. Shmoys. ACM, 2014, pp. 475–484. DOI: 10.1145/2591796.2591825.

[SZ25a] Omri Shmueli and Mark Zhandry. "On One-Shot Signatures, Quantum vs Classical Binding, and Obfuscating Permutations". In: *Cryptology ePrint Archive* Paper 2025/486 (2025). Revised 2026-01-25.

[SZ25b] Omri Shmueli and Mark Zhandry. "On One-Shot Signatures, Quantum vs. Classical Binding, and Obfuscating Permutations". In: *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part II*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Vol. 16001. Lecture Notes in Computer Science. Springer, 2025, pp. 350–383. DOI: 10.1007/978-3-032-01878-6_12.

[Uma03] Christopher Umans. "Pseudo-Random Generators for all Hardnesses". In: *J. Comput. Syst. Sci.* 67.2 (2003), pp. 419–440. DOI: 10.1016/S0022-0000(03)00046-1.

[Val84] Leslie G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (1984), pp. 1134–1142. DOI: 10.1145/1968.1972.

[WW21] Hoeteck Wee and Daniel Wichs. "Candidate Obfuscation via Oblivious LWE Sampling". In: *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*. Ed. by Anne Canteaut and François-Xavier Standaert. Lecture Notes in Computer Science. Springer, 2021, pp. 127–156. DOI: 10.1007/978-3-030-77883-5_5.

# A Unconditional coNP-Hardness of a Variant of Total-Learn

Here we consider the promise-problem $\mathsf{Total\text{-}Learn}_{g,1/2}[s]$. It is easily seen to be in $\mathsf{promise}\text{-}\Sigma_2^p$. It is natural to conjecture that this problem is also hard for $\Sigma_2^p$-hard. Below we show, unconditionally, a weaker coNP-hardness result.

**Lemma 60.** *For all sufficiently large $s$ and every polynomial $g$, $\mathsf{Total\text{-}Learn}_{g,1/2}[s]$ is coNP-hard under (coRP-type) randomized many-one reductions.*

*Proof.* By the Valiant–Vazirani theorem it suffices to give a randomized reduction from the complement of Unique-SAT, the promise problem where "yes" instances are unsatisfiable formulas and "no" instances are formulas with a unique satisfying assignment. Let $s$ be sufficiently large, and let $g$ by a polynomial. We claim that the following is a randomized reduction from the complement of Unique-SAT to $\mathsf{Total\text{-}Learn}_{g,1/2}[s]$.

On input $\varphi$:

1. Let $m$ be the number of variables in $\varphi$.
2. Pick $r_1, \ldots, r_{10g(s)} \in \{0,1\}$ independently and uniformly at random.
3. Let $a_1, \ldots, a_{10g(s)} \in \{0,1\}^m$ denote the $10g(s)$ lexicographically first strings.
4. Output a sampler $\mathcal{E}$ that works as follows.
   (a) Pick $x \in \{0,1\}^m$ uniformly at random.
   (b) Let
   $$y = \begin{cases} x_1 & \text{if } \forall i, \varphi(x \oplus a_i) = 0 \\ r_i & \text{if } i \text{ is the first number such that } \varphi(x \oplus a_i) = 1 \end{cases}$$
   (c) Output $(x, y)$.

**Completeness** Suppose $\varphi$ is a "no" instance of Unique-SAT. Then for all $x \in \{0,1\}^m$ and $i \in [10g(s)]$, $\varphi(x \oplus a_i) = 0$. Thus the sampler $\mathcal{E}$ produced by the reduction always samples $(x, x_1)$ for some $x \in \{0,1\}^m$. This means that the size-1 circuit $C$ that simply outputs the first bit of its input satisfies $\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1$. Hence $\mathcal{E}$ is a "yes" instance of $\mathsf{Total\text{-}Learn}_g[s]$.

**Soundness** Now suppose $\varphi$ is a "yes" instance of Unique-SAT. Then there is a unique $x_0 \in \{0,1\}^m$ such that $\varphi(x_0) = 1$. This means that for each $i \in [10g(s)]$, $\mathcal{E}$ samples $(x_0 \oplus a_i, r_i)$ with positive probability. Suppose there is a circuit $C$ of size $g(s)$ such that $\Pr_{(x,b)\sim\mathcal{E}}[C(x) = b] = 1$. Consider the following Turing machine $M$.

Memorize $C$ and $x_0$. On input $i \in [10g(s)]$, compute $a_i$, run $C(x_0 \oplus a_i)$, and output the resulting bit.

The description length of $M$ is roughly the description length of $C$ plus $m$, which is at most $g(s) + m$. It is easy to see that for all $i \in [10g(s)]$, $M(i)$ outputs $r_i$. Thus $\mathsf{K}(r_1, \ldots, r_{10g(s)}) \leq g(s) + m \leq 10g(s) - O(\log s)$. However, since the $r_i$ are uniformly random bits, we know that $\mathsf{K}(r_1, \ldots, r_{10g(s)}) > 10g(s) - O(\log s)$ with overwhelming probability. Thus with overwhelming probability such a circuit $C$ cannot exist. □

**Corollary 61.** *There is a constant $C$ such that for every $s > C$ and every polynomial $g$, the promise-problem* $\mathsf{Total\text{-}Learn}_{g,1/2}[s]$ *is* $\mathsf{NP}$*-hard under randomized one-query Turing reductions.*



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il