NP-hardness of the Minimum Circuit Size Problem from Well-Studied Assumptions

Shuichi Hirahara
National Institue of Informatics
Tokyo, Japan
s_hirahara@nii.ac.jp

Rahul Ilango
Massachusetts Institute of Technology
Cambridge, MA, USA
rilango@mit.edu

***Abstract*—Whether the Minimum Circuit Size Problem (MCSP) is NP-hard or not is a long-standing open question. Indeed, Levin delayed the publication of his fundamental work on the theory of NP-completeness because he hoped to prove NP-completeness of MCSP.**

In this paper, we present the first plausible assumptions under which MCSP is NP-hard. Specifically, we prove that MCSP is NP-hard under deterministic quasi-polynomial-time nonadaptive reductions, assuming:

- **subexponentially-secure non-interactive witness indistinguishable proof systems for SAT exist,**
- **coNP requires subexponential-size non-deterministic circuits, and**
- **$\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ requires circuits of size $\Omega(2^n/n)$.**

This is arguably the first evidence that MCSP is not in coNP, which indicates that there is no short proof that witnesses the hardness of a function.

***Index Terms*—Minimum Circuit Size Problem, NP-hardness, Cryptography, Meta-complexity**

I. Introduction

The Cook–Levin theorem is a cornerstone of computational complexity theory, introducing the notion of NP-completeness and establishing that the Satisfiability problem (SAT) is NP-complete. This landmark result was independently proved by Cook [1] and Levin [2] during the Cold War. The paper of Levin was published in 1973, two years after the publication of Cook. An intriguing anecdote[1] is that Levin intentionally delayed his publication because he could not prove NP-completeness of "popular" problems, such as graph isomorphism, integer factorization, and "circuit minimization" — nowadays called the *Minimum Circuit Size Problem* (MCSP) [3]. The graph isomorphism problem is known to be in $\mathsf{NP} \cap \mathsf{coAM}$ [4], and, similarly, (the decision version of) the integer factorization problem is in $\mathsf{NP} \cap \mathsf{coNP}$ [5]. This implies that these problems cannot be NP-complete (even under randomized polynomial-time adaptive reductions) unless $\mathsf{NP} \subseteq \mathsf{coAM}$, providing strong complexity-theoretic evidence against their NP-completeness. However, no evidence *for* or *against* NP-completeness of MCSP is known. One of the oldest open questions, dating back to the beginning of the theory of NP-completeness, is the following:

Question. *Is* MCSP NP*-complete? Are there plausible assumptions under which this is true or false?*

MCSP is formally defined as follows. For a function $f\colon \{0,1\}^n \to \{0,1\}$, the *truth table of* f is defined as the string of length $N = 2^n$ obtained by concatenating $f(x)$ for all $x \in \{0,1\}^n$ in the lexicographical order. The *Minimum Circuit Size Problem* (MCSP) is the following decision problem.

Input: the truth table of a function $f\colon \{0,1\}^n \to \{0,1\}$ and a size parameter $s \in \mathbb{N}$.
Task: decide whether there is a Boolean circuit of size s that computes f.

It is easy to see that MCSP is in $\mathsf{E} = \mathsf{DTIME}\big(2^{O(N)}\big)$: Since any n-bit function can be computed by a circuit of size $(1+o(1)) \cdot 2^n/n$ and such circuits can be described by $(1+o(1)) \cdot 2^n$-bit strings [6], the minimum circuit size can be calculated by exhaustively searching all the $2^{(1+o(1))\cdot 2^n}$ circuits in time $2^{O(N)} = 2^{O(2^n)}$.[2] Similarly, it is easy to see that MCSP is in NP. One of the best hardness results was given by Allender and Das [9], who proved that MCSP is hard for SZK (Statistical Zero Knowledge). Since SZK contains cryptographic problems believed to be intractable, such as quadratic residuosity [10] and integer factorization [11], the SZK-hardness of MCSP provides evidence for the computational intractability of MCSP.

Meta-Complexity: MCSP is one of the central problems in the paradigm of *meta-complexity*, i.e., "complexity of complexity." The complexity of MCSP is referred to as meta-complexity because the problem itself asks for computing the *circuit complexity* of a given function. Another important example of meta-complexity problem is the problem of computing the *t-time-bounded Kolmogorov complexity* $\mathsf{K}^t(x)$ of a given string x, i.e., the length of a shortest program that prints x in time t. Meta-complexity has deep connections to many areas of theoretical computer science, such as computational learning theory [12], cryptography [13, 14], circuit lower bounds [15, 3, 16], derandomization [17, 18], and average-case complexity [19]. For example, Hirahara [19] proved that NP-hardness of an approximation version of MCSP implies the

Shuichi Hirahara was supported by JST FOREST Program, Grant Number JPMJFR226Y. Rahul Ilango was supported by NSF CCF-2420092 and an NSF graduate research fellowship.

[1] See Levin's webpage (https://www.cs.bu.edu/fac/lnd/research/hard.htm) for details.

[2] In fact, MCSP can be solved by a circuit of size $2^{(0.8+o(1))\cdot N}$ [7, 8].

equivalence between the worst- and average-case complexities of NP — one of the central problems in the theory of average-case complexity, known as the exclusion of Heuristica from Impagliazzo's five possible worlds [20]. Meta-complexity has been actively investigated, with the hope of ruling out Heuristica [19, 21, 17, 22, 23, 24, 25, 26, 27, 28, 29, 30], Pessiland [14, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 13, 41, 42], both of Heuristica and Pessiland [43, 40, 44, 45], and Minicrypt [46, 47].

Circumstantial Evidence for NP-*hardness of* MCSP*:* Many variants of MCSP have been shown to be NP-hard, providing circumstantial evidence for NP-hardness of MCSP. The original paper of Levin [2] presented six NP-complete problems, one of which is DNF-MCSP* — the problem of deciding whether a given *partial* function $f\colon \{0,1\}^n \to \{0,1,\star\}$ (represented as its truth table) has a DNF formula of size s for a given parameter s, where $f(x) = \star$ indicates that f is undefined on input x. Masek [48] proved NP-completeness of DNF-MCSP, i.e., MCSP for DNF formulas (see also [49]). Hirahara et al. [50] extended this to DNF $\circ$ XOR formulas. Ilango [51] proved NP-hardness of an oracle circuit version of MCSP. Ilango [52] proved NP-hardness of MCSP for constant depth formulas (under randomized quasi-polynomial-time reductions). Ilango et al. [53] proved NP-hardness of MCSP for multi-output functions. Ilango [54] proved ETH-hardness of the formula version of MCSP. Hirahara [55] proved NP-hardness of MCSP*, i.e., the partial function variant of MCSP under randomized polynomial-time reductions. Huang et al. [56] gave improved inapproximability of the oracle circuit version of MCSP and first suggested the idea of using cryptographic assumptions for proving the hardness of MCSP. Ilango [57] proved that SAT reduces to the $\mathcal{O}$-oracle circuit variant of MCSP, denoted MCSP$^{\mathcal{O}}$, with probability 1, where $\mathcal{O}$ is a random oracle. This result provides a concrete candidate reduction for showing NP-hardness of MCSP, as the random oracle could be replaced with practical hash functions. Still, it is an outstanding question whether MCSP is NP-hard under plausible assumptions.

Circumstantial Evidence against NP-*hardness of* MCSP*:* One difficulty is that restricted classes of reductions are not capable of establishing NP-hardness of MCSP. One of the strongest results in this direction was recently presented by Mazor and Pass [58]. They showed that a 2.01-factor approximation of MCSP is not NP-hard under randomized Levin reductions, assuming plausible cryptographic assumptions, such as the existence of an indistinguishable obfuscation. Informally, a *Levin reduction* from $L \in \mathsf{NP}$ to $L' \in \mathsf{NP}$ is a reduction that efficiently maps a witness of L to L' and vice versa. The reduction of Ilango [57] is a Levin reduction under a random oracle, which poses a fundamental barrier to extend it to NP-hardness of MCSP.

In addition to the barrier of [58], any NP-hardness proof of MCSP must avoid all the other barriers presented in the literature. Murray and Williams [59] showed that MCSP is not NP-hard under $n^{.49}$-time reductions. An analogous result for time-bounded Kolmogorov complexity was proved by Saks and Santhanam [60]. In fact, almost all NP-complete problems, such as SAT, Vertex Cover, and Hamiltonian Path, are also complete under poly$(\log n)$-time reductions (see [59] and references therein). Thus, these barriers suggest that any NP-hardness proof of MCSP must look quite different from standard NP-hardness proofs. Hirahara and Watanabe [61] proved that there is no NP-hardness reduction to the A-oracle version MCSP^A for some oracle A unless NP $\not\subseteq$ coAM. Ko [62] (resp. Ren and Santhanam [63]) presented oracles under which approximating t-time-bounded Kolmogorov complexity is not NP-hard (resp. MCSP does not admit a search-to-decision reduction, and, in particular, is not NP-hard). A related line of research [3, 59, 64, 61, 65] presented the difficulty of establishing NP-hardness of MCSP unconditionally under *deterministic* reductions, by deriving complexity class separations, such as EXP $\neq$ ZPP, from deterministic NP-hardness reductions. We also mention that an approximation version of MCSP is provably NP-intermediate if the approximation error is sufficiently large and a one-way function exists [66].

In summary, no conclusive results for NP-hardness of MCSP have been obtained so far. The result of Ilango [57] comes close to NP-hardness of MCSP, but it is subject to the barrier of Mazor and Pass [58], suggesting that a new idea would be necessary.

A. Our Results

In this paper, we present the first plausible assumptions under which MCSP is NP-hard, circumventing all the barriers in the literature [58, 60, 59, 61, 62, 63] and answering the oldest open question from the beginning of the theory of NP-completeness.

Theorem I.1. *(A constant factor approximation of) the Minimum Circuit Size Problem is NP-hard under deterministic quasipolynomial-time non-adaptive reductions assuming all of the following:*

1) *subexponentially-secure non-interactive witness indistinguishable proof systems for* SAT *exist,*
2) coNP *requires subexponential-size non-deterministic circuits almost everywhere, and*
3) $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ *requires circuits of size* $\delta 2^n/n$ *almost everywhere for some constant* $\delta > 0$*.*

All the assumptions are plausible and involve only well-studied concepts. We describe the details of each assumption below.

1) The first assumption states that a *non-interactive witness indistinguishable proof systems* (NIWI) [67] for SAT exist. This is a proof system in which a prover can convince a verifier that a given formula is satisfiable without revealing which satisfying assignment the prover used to generate the proof. There are several different constructions of NIWIs from various widely believed assumptions, such as the existence of a trapdoor one-way permutation and the non-deterministic exponential circuit lower bound for E [67]. See Section III-A for the

precise definition of NIWIs and the list of constructions of NIWIs.

2) The second assumption states that $\mathsf{coNP} \not\subseteq \text{i.o.}\mathsf{NSIZE}\left(2^{n^\epsilon}\right)$ for some constant $\epsilon > 0$. Here, i.o. stands for infinitely often, and $\text{i.o.}\mathsf{NSIZE}\left(2^{n^\epsilon}\right)$ is the class of problems that can be solved by a non-deterministic circuit of size 2^{n^ϵ} on inputs of length n for infinitely many $n \in \mathbb{N}$. The assumption is a sub-exponential and almost-everywhere-hardness version of the widely believed assumption that $\mathsf{coNP} \not\subseteq \mathsf{NP/poly}$, which holds unless the Polynomial Hierarchy collapses [68].
3) The third assumption states that $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly} \not\subseteq \text{i.o.}\mathsf{SIZE}(\delta 2^n/n)$ for some constant $\delta > 0$, where $\mathsf{SIZE}(\delta 2^n/n)$ is the class of problems that can be solved by circuits of size $\delta 2^n/n$ on inputs of length n. Chen et al. [69], Li [70] proved that an exponential-time version $\mathsf{S}_2^{\mathsf{E}}$ of $\mathsf{S}_2^{\mathsf{P}} \subseteq \mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ [71] requires circuits of nearly maximum size, i.e., $\mathsf{S}_2^{\mathsf{E}} \not\subseteq \text{i.o.}\mathsf{SIZE}(2^n/n)$. Our assumption is that a similar circuit lower bound holds for $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$. It can be shown that this holds with probability 1 under a random oracle; see Appendix A.

How our proof circumvents the barriers of [58, 60, 59, 61, 62, 63] will be discussed in Section II-C.

As an immediate corollary, we obtain the first plausible and well-studied assumptions under which MCSP is not in coNP.

Corollary I.2. $\mathsf{MCSP} \notin \mathsf{coNP}$ *under the three assumptions of Theorem I.1.*

This indicates the "impossibility of witnessing hardness" — that there is no short proof for the statements for circuit lower bounds, i.e., the statements that a certain function f cannot be computed by circuits of size s. The non-existence of an efficient proof system for this coNP statement (which corresponds to YES instances of the complement of MCSP) is equivalent to $\mathsf{MCSP} \notin \mathsf{coNP}$. It is widely recognized that proving circuit lower bounds for explicit functions is extremely difficult, as the current best circuit lower bound for any problem in NP is only $(3.1-o(1))\cdot n$ [72]. One explanation for this difficulty of proving circuit lower bounds is the lack of short proofs for the circuit lower bound statements. Our results indicate that every efficient proof system fails to prove that a function f cannot be computed by small circuits for some f; in other words, no worst-case variant of NP-constructive natural proofs exists in the sense of Razborov and Rudich [15].

Previously, Rudich [73] conjectured some average-case variant of $\mathsf{MCSP} \notin \mathsf{coNP}$, based on the non-deterministic hardness of a specific pseudorandom generator [74]. Our result comes somewhat close to basing his conjecture on the well-studied assumptions because of the equivalence between the worst-case complexity of the approximation version of MCSP and the average-case complexity of MCSP [19].

It is natural to ask whether our assumptions are necessary. Because our reduction is deterministic, the results in the line of research [3, 59, 64, 61, 65] indicate that some assumptions would be necessary. (For example, Murray and Williams [59] showed that NP-hardness of MCSP under deterministic reductions implies $\mathsf{EXP} \neq \mathsf{ZPP}$.) In fact, using non-uniform reductions, we can remove the third assumption of Theorem I.1, at the cost of worst-case hardness. Below, being NP-*hard on average* roughly means that solving any NP problem on average with respect to the uniform distribution reduces to MCSP.[3]

Corollary I.3 (informal; see Corollary V.9)**.** *Assume that*

1) *subexponentially-secure NIWIs for* SAT *exist, and*
2) coNP *requires subexponential-size non-deterministic circuits almost everywhere.*

Then, MCSP *is* NP*-hard on average under infinitely-often non-uniform subexponential-time non-adaptive reductions.*

II. Technical Overview

Our starting point is (a simplified version of) Hirahara's proof that it is NP-hard to compute the circuit complexity of a partial function $f\colon \{0,1\}^n \to \{0,1,\star\}$ [55]. We begin by reviewing this in Section II-A. We present the ideas of our reduction in Section II-B. Lastly, we explain how our proof circumvents all the barriers in the literature in Section II-C.

A. An Exposition of Hirahara's NP-hardness of MCSP*

a) Starting Problem: Vertex Cover: The reduction is from the NP-hard problem Vertex Cover.[4] Recall that a vertex cover is a subset of vertices that intersect every edge. It is NP-hard to calculate the minimum size OPT of a vertex cover in a given graph.

b) Tool: Extractable Information Theoretic Encryption: It will be useful to consider the following information-theoretic symmetric encryption scheme, closely related to Shannon's one-time pad and the standard construction of an encryption scheme from a pseudorandom function.

- *Key Generation:* sample a uniformly random function $H\colon [\lambda] \to \{0,1\}$ where λ is the security parameter.
- *Encryption:* To encrypt a bit b, one samples $r \leftarrow [\lambda]$ uniformly at random and outputs $\mathrm{ct} = (r, b \oplus H(r))$. We denote this algorithm by $\mathrm{Enc}(H, b)$.
- *Decryption:* To decrypt $\mathrm{ct} = (r, c)$, one outputs $c \oplus H(r)$. We denote this algorithm by $\mathrm{Dec}(H, \mathrm{ct})$.

An useful property of this scheme is *extractability*. If an adversary A can distinguish an encryption of 1 from an encryption of 0, then one can extract out the secret key (in a certain quantitative sense). We delay describing this until we have to use it, but we mention that it will follow immediately from the distinguisher-to-predictor lemma [75].

[3]The uniform distribution can be replaced with any polynomial-time samplable distribution because Impagliazzo and Levin [13] reduced solving NP on any polynomial-time samplable distribution to solving NP on the uniform distribution.

[4]In our actual proof, we will need to switch to hypergraph vertex cover to get better hardness of approximation. We ignore this in this high-level overview.

Given an n-vertex graph with edge set E:

1) Sample secret keys $H_v \colon [\lambda] \to \{0,1\}$ for all $v \in [n]$
2) Define the partial function $f \colon E \times ([\lambda] \times \{0,1\}) \times ([\lambda] \times \{0,1\}) \to \{0,1,\star\}$ as follows:

$$f(e=(v,w),\mathrm{ct}_v,\mathrm{ct}_w) = \begin{cases} \mathrm{Dec}(H_v,\mathrm{ct}_v) & \text{if } \mathrm{Dec}(H_v,\mathrm{ct}_v) = \mathrm{Dec}(H_w,\mathrm{ct}_w), \\ \star & \text{otherwise.} \end{cases}$$

3) Output that OPT is "roughly" the circuit complexity of f divided by λ.

Fig. 1. The reduction from Vertex Cover to MCSP^*.

c) Reduction: We can now describe the reduction. Given a graph with edge set E, one first samples independent secret keys for each vertex. Then one considers the function f that takes as input an edge $e = (v,w) \in E$ and two "purported" encryptions ct_v and ct_w. The function is undefined in the case where ct_v and ct_w encrypt different bits (according to the secret keys of the respective vertices). But if they encrypt the same bit b, the function outputs b. We have written the reduction formally in Figure 1 (parameterized by a choice of security parameter λ).

The intuition is that in order to compute f one ought to memorize the secret key of at least one vertex in each edge (so that you can decrypt at least one of the ciphertexts). Hence, one should need to memorize at least OPT (the optimal vertex cover size) many λ-length secret keys. So the complexity of f should be roughly $\mathrm{OPT} \cdot \lambda$.

For the upper bound, one can show that indeed f can be computed by a circuit of size roughly $\mathrm{OPT} \cdot \lambda$ (as long as we set λ to be a sufficiently large polynomial in n).

d) Lower Bound Proof: The proof of the lower bound works as follows. Fix a circuit C computing f. Fix an arbitrary e in E. Now we will use the encryption scheme's extractability property. By the definition of f, we have that

$$\Pr_{\substack{b \leftarrow \{0,1\}, \\ \mathrm{ct}_v \leftarrow \mathrm{Enc}(H_v,b) \\ \mathrm{ct}_w \leftarrow \mathrm{Enc}(H_w,b)}} [C(e,\mathrm{ct}_v,\mathrm{ct}_w) = b] = 1.$$

On the other hand, if we instead sampled ct_v and ct_w as truly random strings, the chance $C(e,\mathrm{ct}_v,\mathrm{ct}_w) = b$ is exactly one half (because the inputs to C are independent of b). Hence, by a hybrid argument and the distinguisher-to-predictor lemma (Lemma III.3), one gets that for at least one element of e (for simplicity, assume v) we have

$$\mathsf{K}^C_{1/2+\Omega(1)}(H_v) \leq O(n + \log\lambda),$$

where we use the notation $\mathsf{K}^C_p(z)$ to denote the length of the shortest program that, given oracle access to C, outputs a string that agrees with z on a p-fraction of coordinates.[5] By incorporating error-correction into the encryption scheme, one can improve this to instead be

$$\mathsf{K}^C(H_v) \leq O(n + \log\lambda).$$

[5]See Definition III.2 for a formal definition.

Now since the argument above held for an arbitrary edge, we in fact get that in every edge there is a vertex v with

$$\mathsf{K}^C(H_v) \leq O(n + \log\lambda).$$

Thus, the above bound holds for OPT many vertices. Label these vertices $v_1, \ldots, v_{\mathrm{OPT}}$. One can then show that

$$\mathsf{K}^C(H_{v_1} \ldots H_{v_{\mathrm{OPT}}}) \leq O((n^2 + \log\lambda) \cdot \mathrm{OPT})$$

so

$$\mathsf{K}(H_{v_1} \ldots H_{v_{\mathrm{OPT}}}) \leq O((n^2 + \log\lambda) \cdot \mathrm{OPT}) + |C|.$$

Because we choose the λ-length secret keys uniformly at random, we expect $\mathsf{K}(H_{v_1} \ldots H_{v_{\mathrm{OPT}}}) \approx \mathrm{OPT} \cdot \lambda$. Hence, setting $\lambda = \mathsf{poly}(n)$ sufficient large we get that

$$|C| \approx \mathrm{OPT} \cdot \lambda,$$

as desired.

B. Our Reduction

The reason why Hirahara's reduction requires partial functions is to handle the case of two ciphertexts that encrypt different bits. We will fix this by modifying f to also include a proof π that the two ciphertexts are equivalent. However, this approach faces two immediate problems.

a) Less Serious Problem: The Statement Lacks Short Proofs: Recall, that we have chosen the secret keys H_v and H_w uniformly at random. It does not seem possible in this case to give a proof that ciphertexts ct_v and ct_w decrypt to the same bit (under their corresponding secret keys), unless we assume the verifier reading the proof knows a significant amount of information about our random choice of H_v and H_w. This seems impossible in our setting because we actually want the verifier to *not* know one of these secret keys.

To fix this, we do not choose our secret keys at random. Imagine instead that we choose the λ-length secret keys H_v and H_w such that any bit in the secret key corresponds to a $O(\log\lambda)$-length instance of an $\mathsf{NP} \cap \mathsf{coNP}$ language (in fact, by being more clever one can even consider a P^{NP} language, but for now consider $\mathsf{NP} \cap \mathsf{coNP}$).

Then to prove that two ciphertexts decrypt to the same value we can do the following. Recall that a ciphertext is of the form (r_v, c_v) where $c_v = b \oplus H_v(r_v)$ and b is the bit being encrypted. Also, recall $H_v(r_v)$ is a bit that corresponds to an instance of a $\mathsf{NP} \cap \mathsf{coNP}$ problem. Hence, to prove that

Given an n-vertex graph with edge set E:

1) Let $\{H_v : [\lambda] \to \{0,1\}\}_{v \in [n]}$ be "very hard" P^{NP} functions.
2) For a formula ψ, define the function f_ψ as follows:

$$f_\psi(e = (v,w) \in E, \mathrm{ct}_v, \mathrm{ct}_w, \pi) = \begin{cases} \mathrm{Dec}(H_v, \mathrm{ct}_v) & \text{if } \pi \text{ is a NIWI-proof that} \\ & \text{"either } \mathrm{Dec}(H_v, \mathrm{ct}_v) = \mathrm{Dec}(H_w, \mathrm{ct}_w) \text{ or } \psi \text{ is satisfiable"}, \\ 0 & \text{otherwise.} \end{cases}$$

3) Output that OPT is "roughly" the maximum (over all "small"[a] unsatisfiable ψ) of the circuit complexity of f_ψ divided by λ.

[a] For example, one setting of "small" we use is $\mathsf{poly}\log(n)$.

Fig. 2. The informal description of our reduction from Vertex Cover to MCSP.

two ciphertexts (r_v, c_v) and (r_w, c_w) encrypt the same bit, we can provide the $\mathsf{NP} \cap \mathsf{coNP}$ witnesses that reveal the value of $H_v(r_v)$ and $H_w(r_w)$ respectively. Given this, we can easily check if the two ciphertexts encrypt the same bit.

Now at least a short proof of equivalence exists, but the proof we described above is actually not good for us. In the process of proving that the ciphertexts are equivalent it also reveals the bit they encrypt. As a result, the function f actually becomes easy to compute, regardless of the minimum vertex cover size.

b) More Serious Problem: How Do You Keep f Hard: Indeed, this brings us to our main technical challenge. How can we prove that the ciphertexts are equivalent while still maintaining the hardness of f? A natural idea is to use a zero knowledge proof [10], which guarantees the verifier "learns nothing except the validity of the statement."

Unfortunately, this idea runs into two significant issues. First, in our setting the proof is *completely non-interactive*: the circuit is just given a proof π and needs to decide whether it is valid or not. Second, we need the proof to have *perfect soundness*. The verifier can never be convinced that two ciphertexts are equivalent when they are not. No known zero knowledge proofs have either of these two properties, and indeed, Goldreich and Oren [76] show that achieving either of them is impossible (under a standard complexity assumption).

Luckily, we do not actually need the full strength of zero knowledge. We only need a very specific security property to hold: that computing f is still hard. A proof system with such a property was constructed by Kuykendall and Zhandry [77]. For every NP-search problem R that is hard on average, they constructed a non-uniform *witness-hiding* proof system, in which a prover can convince a verifier that an input x has some witness in R without revealing the witness: no efficient adversary can solve R on average even given a proof from the prover.

c) Using Non-interactive Witness Indistinguishable Proofs: The proof system of Kuykendall and Zhandry [77] is based on non-interactive witness indistinguishable proofs (NIWIs). This is one relaxation of zero knowledge that is achievable with no interaction and perfect soundness. A NIWI proof that a circuit C is satisfiable has the following "witness indistinguishability" security property: one cannot tell which satisfying assignment to C was used to generate the proof (except if C is uniquely satisfiable).

Building on Kuykendall and Zhandry [77], we will do the following. Let ψ be a formula that we choose later. We modify f to include a NIWI proof π that "either these two ciphertexts encrypt the same bit or ψ is satisfiable."

Now if ψ is satisfiable, then this proof says nothing about whether the two ciphertexts encrypt the same bit. However, precisely because of this, one can show (via the witness indistinguishability property) that providing such a proof is useless and that f is still hard. In fact, f is even hard "on average," which will be useful for us later.

On the other hand if ψ is unsatisfiable, then a proof that "either these two ciphertexts encrypt the same bit or ψ is satisfiable" does indicate that the ciphertexts indeed encrypt the same bit. However, it is not clear whether f is still hard.

Luckily, using the ideas of Kuykendall and Zhandry, we get that there should be *some* unsatisfiable ψ for which f is still hard. The reason is that f being easy is a short certificate that ψ is unsatisfiable. This is because (as mentioned earlier) f is indeed hard, even on average, when ψ is satisfiable. So if you are given a circuit that computes f on random inputs, it must be that ψ is unsatisfiable. (We note that this argument is subtle and breaks down for super-linear-time-bounded Kolmogorov complexity.)

But if one believes $\mathsf{NP} \neq \mathsf{coNP}$ (and its extensions), then it is impossible to certify every unsatisfiable formula is indeed unsatisfiable. Hence, f should be hard for some unsatisfiable ψ. In our reduction, we will brute force over all small unsatisfiable ψ in order to find one with this property.

d) The Reduction: We give a very informal description our reduction in Figure 2.

C. How the Barriers Are Circumvented

Here, we explain how our proof circumvents all the barriers in the literature.

a) The Barrier of Mazor and Pass: Mazor and Pass [58] show that, under cryptographic assumptions, approximating MCSP to a factor of 2.01 is not NP-hard under Levin reductions. Because our results give hardness of approximation to any constant factor, our reduction must overcome this barrier. Indeed, we overcome it in a rather remarkable way: our reduction is not Levin, but it is in some sense *indistinguishable* from a Levin reduction.

In more detail, let R be a reduction that maps instances x of a language $L \in \mathsf{NP}$ to an instance $R(x)$ of another language $L' \in \mathsf{NP}$. For this to be a valid reduction, it must be sound, meaning that if $R(x) \in L'$, then $x \in L$. A *Levin* reduction has the property that its soundness proof is algorithmic: there is a polynomial-time algorithm that maps a witness that $R(x) \in L$ to a witness that $x \in L$.[6] For the vast majority of NP-hardness reductions, such an algorithm usually follows immediately by a reduction's proof of soundness, which are almost always constructive. Consequently, this barrier seems formidable: what non-constructive techniques could be useful in a soundness proof?

The key idea for us is as follows. In our reduction, we iterate through all unsatisfiable formulas ψ. But in our analysis, we say that — because of non-deterministic hardness of UNSAT — this should actually be indistinguishable from the case where we iterate through at least one satisfiable ψ. For a satisfiable ψ, however, we can actually prove soundness algorithmically. Specifically, we can use the encryption scheme's extractability property to extract from any small circuit a satisfying assignment.

In this sense, our reduction is not Levin (because we only iterate through unsatisfiable ψ), but almost Levin (because in our analysis we pretend ψ is satisfiable).

b) The Barrier of Saks and Santhanam, Murray and Williams: Saks and Santhanam [59] showed that any randomized nonadaptive reduction that shows NP-hardness of approximating t-time-bounded Kolmogorov complexity must spend at least $t^{\Omega(1)}$ time under plausible assumptions. Similarly,[7] Murray and Williams [59] showed that there is no deterministic $n^{.49}$-time reduction from NP to MCSP.

A high-level intuition behind these barriers is that any t-time (deterministic) reduction cannot create a string with high t-time-bounded Kolmogorov complexity. Our proof overcomes the barriers, by spending (at least) $\mathsf{poly}(t)$ time to construct some string with nearly maximum t-time-bounded Kolmogorov complexity, which can be obtained from the hard function in $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ that requires the nearly maximum circuit complexity (Lemma V.6).

c) The Barrier of Hirahara and Watanabe: Hirahara and Watanabe [61] showed that there is no randomized NP-hardness reduction to MCSP^A for some oracle A unless $\mathsf{NP} \not\subseteq \mathsf{coAM}$. We bypass this barrier because our result dose not generalize to MCSP^f, where f is the hard function in $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ with nearly maximum circuit complexity. This is in contrast to the NP-hardness of $\mathsf{MCSP}^\star$ [55], which can be generalized to $\mathsf{MCSP}^{\star,A}$ for every oracle A.

d) The Relativization Barriers of Ko [62], Ren and Santhanam [63]: Our proof does not relativize, just as in the case of NP-hardness of $\mathsf{MCSP}^\star$ [55]. Hirahara also proved NP-hardness of a problem called MINLT [62], which is provably non-relativizing. The primary non-relativizing component in our reduction (and [55]) is the use of the NP-hardness of approximating (hypergraph) vertex cover [79], which relies on the PCP theorem. The PCP theorem is non-relativizing [80].

Organization

The remainder of this paper is organized as follows. After presenting preliminaries in Section III, we introduce the notion of somewhat $\mathsf{NP} \cap \mathsf{coNP}$ computable functions, which enables us to reduce the computation of P^{NP} to some form of non-uniform $\mathsf{NP} \cap \mathsf{coNP}$ computations in Section IV. The proof of Theorem I.1 is given in Section V. In Appendix A, a strong lower bound for NP is proved under a random oracle.

III. Preliminaries

a) Notation: $[n]$ denotes $\{1, \ldots, n\}$. The *circuit complexity* of a function $f\colon \{0,1\}^n \to \{0,1\}$, denoted by $\mathsf{CC}(f)$, is the minimum size of any circuit that computes f. We use the same notation $\mathsf{CC}(t)$ for the truth table $t \in \{0,1\}^{2^n}$ of f. For a string $x \in \{0,1\}^*$, let $|x|$ denote the length of x.

A. Non-Interactive Witness Indistinguishable Proofs

We recall the definition of a non-interactive witness indistinguishable proof system (NIWI) [81, 82, 67] for SAT. (In this paper, whenever we refer to NIWIs, we refer to NIWIs for SAT.)

Definition III.1 (Non-Interactive Witness Indistinguishable Proof (NIWI))**.** A *non-interactive witness indistinguishable proof system* is a tuple of algorithms $(\mathsf{NIWI.Prove}, \mathsf{NIWI.Verify})$ satisfying all of the following properties:

- $\mathsf{NIWI.Prove}(\text{formula } \varphi, \text{satisfying assignment } w, \text{security parameter } 1^\lambda)$ is a uniform randomized polynomial time algorithm that outputs a binary string (usually denoted π).
- $\mathsf{NIWI.Verify}(\text{formula } \varphi, \text{purported proof } \pi)$ is a uniform polynomial-time algorithm that outputs either 0 or 1.
- **Completeness:** For all formulas φ with $\varphi(w) = 1$ and all λ,

$$\Pr[\mathsf{NIWI.Verify}(\varphi, \mathsf{NIWI.Prove}(\varphi, w, 1^\lambda), 1^\lambda) = 1] = 1,$$

  where the probability is taken over the internal randomness of $\mathsf{NIWI.Prove}$.
- **Soundness:** For all unsatisfiable formulas φ, all strings π, and all λ,

$$\mathsf{NIWI.Verify}(\varphi, \pi, 1^\lambda) = 0.$$

- **Security (Witness Indistinguishability):** There is a negligible function ϵ such that for all formulas φ and all

[6] It also has the vice-versa property, but that will be less important to us.

[7] MCSP can be regarded as the problem of computing sublinear-time-bounded Kolmogorov complexity [78].

assignments w and w' with $\varphi(w) = \varphi(w') = 1$, for all adversary circuits A of size at most $1/\epsilon(\lambda)$, we have that

$$\big|\Pr[A(\mathsf{NIWI.Prove}(\varphi, w, 1^\lambda)) = 1] - \Pr[A(\mathsf{NIWI.Prove}(\varphi, w', 1^\lambda)) = 1]\big| < \epsilon(\lambda), \quad (1)$$

where the probability is taken over the internal randomness of NIWI.Prove.

There are several different constructions of NIWIs from various widely-believed cryptographic assumptions. For example, NIWIs exist either

- if indistinguishability obfuscation and one-way permutations exist [83],
- if a trapdoor one-way permutation exists and E requires $2^{\Omega(n)}$-size non-deterministic circuits [67],
- if indistinguishability obfuscation and one-way functions exist and E requires $2^{\Omega(n)}$-size non-deterministic circuits [67, 83], or
- assuming certain hardness [84] for bilinear groups [85].

B. Time-Bounded Kolmogorov Complexity

Fix an efficient universal Turing machine U. We write $U^t(d)$ to denote the output of running U on d for t steps.

Definition III.2. The p-average t-time $\mathcal{O}$-oracle Kolmogorov complexity of a string x is $\mathsf{K}^{t,\mathcal{O}}_p(x) = \min\{|d| : U^t(d,i) = x_i \text{ for a } p\text{-fraction of } i \in [|x|]\}$. We omit writing p when $p = 1$. We omit writing $\mathcal{O}$ when $\mathcal{O} = \emptyset$.

The following lemma will be useful to us.

Lemma III.3 (Distinguisher to Predictor [75])**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$. Let $\mathcal{D}$ be a distribution on $\{0,1\}^q$. Let D be a circuit that outputs a bit and satisfies*

$$\left| \Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D} \\ b=f(r)}}[D(z,r,b)=1] - \Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D} \\ b\leftarrow\{0,1\}}}[D(z,r,b)=1] \right| > \epsilon \quad (2)$$

Then $\mathsf{K}^{\mathsf{poly}(n,|z|),D}_{1/2+\epsilon}(f) \le |z| + O(1)$.

Proof. Without loss of generality, assume (2) holds without the absolute value signs (if not, then the argument is similar, except one negates the output of D). Consider the randomized algorithm $P(r)$ that samples $b \leftarrow \{0,1\}$, $z \leftarrow \mathcal{D}$ and outputs $b \oplus 1 \oplus D(z,r,b)$. Then we have that

$$\begin{aligned}
&\Pr_{r\leftarrow\{0,1\}^n}[P(r) = f(r)] \\
=&\frac{1}{2}\Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D}}}[D(z,r,f(r))=1] + \frac{1}{2}\Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D}}}[D(z,r,1\oplus f(r))=0] \\
=&\frac{1}{2}\left(\mathbb{E}_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D}}}[D(z,r,f(r)) + 1 - D(z,r,1\oplus f(r))]\right) \\
=&\frac{1}{2} + \frac{1}{2}\Bigg(\Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D}}}[D(z,r,f(r))=1] - \\
&\qquad \Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D}}}[D(z,r,1\oplus f(r))=1]\Bigg) \\
=&\frac{1}{2} + \frac{1}{2}\cdot 2\Bigg(\Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D} \\ b=f(r)}}[D(z,r,f(r))=1] - \\
&\qquad \Pr_{\substack{r\leftarrow\{0,1\}^n \\ z\leftarrow\mathcal{D} \\ b\leftarrow\{0,1\}}}[D(z,r,1\oplus f(r))=1]\Bigg) \\
\ge&\frac{1}{2} + \epsilon.
\end{aligned}$$

Then by an averaging argument we can fix $b \in \{0,1\}$ and $z \in \{0,1\}^q$, so that $P_{b,z}$ is a deterministic algorithm satisfying $\Pr_r[P(r) = f(r)] \ge 1/2 + \epsilon$. Thus, we get that $\mathsf{K}^{\mathsf{poly}(n,q),D}(f) \le q + O(1)$. □

We will also use the following bound on the complexity of biased strings.

Theorem III.4 (Time-Bounded Kolmogorov Complexity of Biased Strings [86])**.** *Let $x \in \{0,1\}^n$ with $k \le n/2$ ones. Then for all t,*

$$\mathsf{K}^{O(t^2)}(x) \le \log\binom{n}{k} + O\Big(\frac{n}{t}\Big) \le nH\Big(\frac{k}{n}\Big) + O\Big(\frac{n}{t}\Big)$$

where H is the binary entropy function which satisfies $H(x) \le 3\sqrt{x}$.

C. Hardness Amplification

Yao's XOR lemma implies that one can amplify average-case hardness in the following way.

Lemma III.5. *There is a deterministic algorithm that takes as input $\epsilon > 0$ such that $\epsilon^{-1} \in \mathbb{N}$ and the truth table of a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ and outputs in time $\mathsf{poly}(2^{n'})$ the truth table of a function $\mathrm{Amp}(f,\epsilon)\colon \{0,1\}^{n'} \to \{0,1\}$ with the following properties:*

- *$n' \le O(n/\epsilon^2)$.*
- *$\mathrm{Amp}(f,\epsilon)$ is computable by an oracle circuit of size $\mathsf{poly}\big(\frac{n}{\epsilon^2}\big)$ that makes $O(1/\epsilon^2)$ non-adaptive oracle queries to f.*
- *The following holds for all t:*

$$\mathsf{K}^{\mathsf{poly}(tn/\epsilon)}_{1-\epsilon}(f) \le \mathsf{K}^t_{1/2+\epsilon}(\mathrm{Amp}(f,\epsilon)) + \mathsf{poly}\Big(\frac{n}{\epsilon}\Big).$$

Proof Sketch. We define $\mathrm{Amp}(f,\epsilon)$ as the k-wise XOR of f such that

$$\mathrm{Amp}(f,\epsilon)(x_1,\dots,x_k) \coloneqq f(x_1)\oplus\cdots\oplus f(x_k)$$

for some parameter k chosen later. Impagliazzo et al. [87] proved a uniform version of Yao's XOR lemma: There exists a randomized algorithm R that, given oracle access to a function g that agrees with $\mathrm{Amp}(f,\epsilon)$ on a $(1/2+\epsilon)$ fraction of inputs, outputs a list of $O(1/\epsilon^2)$ many oralce circuits C^g, one of which computes f on a $(1-\delta)$ fraction of inputs with probability $\Omega(\epsilon)$ in time $\mathsf{poly}(n,k,1/\epsilon)$, where $\delta = O((\log(1/\epsilon))/k)$. We choose $k \coloneqq O\big(1/\epsilon^2\big)$ so that $\delta \le \epsilon$. Then we have $n' = nk = O(n/\epsilon^2)$.

To see the last property, let A be a t-time machine that computes $\mathrm{Amp}(f,\epsilon)$ on a $(1/2+\epsilon)$ fraction of inputs. By running R with oracle access to A, with a positive probability, R^A outputs a list of machines, one of which computes f on a $(1-\delta)$ fraction of inputs. Such a machine can be specified by the internal randomness of R^A and the index of the machine in the list, which cost $\mathsf{poly}(n/\epsilon)$ bits and $O(\log 1/\epsilon)$ bits, respectively. □

D. Hypergraph Vertex Cover

A τ-uniform hypergraph is specified by the vertex set $[n]$ and an edge set $E \subseteq [n]^\tau$. The minimum vertex cover size for such a hypergraph is defined as

$$\min\{|V| : |V\cap e| \ge 1 \text{ for all } e\in E\}.$$

The following NP-hardness is known for this problem.

Theorem III.6 ([79])**.** *There is a universal constant $\gamma \ge 1$ such that the following promise problem is* NP*-hard for all $\gamma < \tau \le \log^{1/\gamma} n$ under $n^{O(\log\tau)}$-time reductions*

- ***Given:*** *A τ-uniform hypergraph on vertex set $[n]$ and edge set $E \subseteq [n]^\tau$*
- ***Output YES:*** *if the minimum vertex cover size is at most $\gamma\frac{n}{\tau}$*
- ***Output NO****: if the minimum vertex cover size is at least $n(1-\gamma/\tau)$*

We refer to this problem as Gap τ-Uniform Vertex Cover.

Throughout this paper, γ refers to the universal constant in Theorem III.6.

E. Uhlig's Theorem

Uhlig [88, 89] shows an efficient way of computing a function on multiple inputs, at least when the function is very hard.

Theorem III.7 (Uhlig's theorem [88, 89]; see also [90])**.** *Let $f : \{0,1\}^n \to \{0,1\}$. Let $f^k : (\{0,1\}^n)^k \to \{0,1\}^k$ be the k-wise direct sum of f. If $k \le 2^{n/\log^2 n}$ and n is sufficiently large, then there is a circuit for f^k of size at most $2\cdot\frac{2^n}{n}$*

IV. Somewhat NP ∩ coNP Computable Functions

We define a certain class of functions that we call *somewhat* NP ∩ coNP *computable*.

Definition IV.1. We say a function $f\colon \{0,1\}^n \to \{0,1\}$ is (S,s)-*somewhat* NP ∩ coNP-*computable* if there exists a S-size circuit Gen that on every input $x \in \{0,1\}^n$ outputs a circuit D_x of size at most s such that all the following hold:

- The range of D_x is $\{0,1,\perp\}$,
- if $D_x(w) \ne \perp$, then $D_x(w) = f(x)$, and
- $D_x(w) = f(x)$ for some w.

Any $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly}$ computation can be simulated by somewhat NP ∩ coNP computation.

Proposition IV.2. *Let $m \in [n]$. Let $f\colon \{0,1\}^n \to \{0,1\}$ be computed by a* SAT*-oracle circuit C^{SAT}. Then f is (S,s)-somewhat* NP ∩ coNP*-computable, where*

$$S \le n^2 2^{n-m}|C| + \mathsf{poly}(|C|, 2^m)$$

and

$$s \le \mathsf{poly}(|C|, 2^m).$$

Moreover, the S-size generator for f can be computed in exponential time given as input the description of the oracle circuit C.

Proof. The idea is that we can divide the set of all inputs into buckets. In each bucket B, we reveal the value $v_{i,B}$ of how many inputs in that bucket have their i'th oracle query equal to YES. Then to certify the value of $f(x)$ you reveal witnesses to every YES oracle query for every input in the same bucket as x.

In more detail, let $\mathcal{B}$ be a partition of $\{0,1\}^n$ that we choose later. For a bucket $B \in \mathcal{B}$, let $v_{i,B}$ be given by $v_{i,B} = \big|\{x \in B : \text{the answer to the } i\text{'th oracle query in } C^{\mathsf{SAT}}(x) \text{ is YES}\}\big|$.

For an input x in the bucket B, we construct the verifier circuit D_x as follows.

Circuit D_x

Given a collection of witnesses $\mathcal{W}_i$:

1) Simulate running $C^{\mathsf{SAT}}(y)$ on every $y \in B$. Answer an oracle query φ as follows: evaluate φ on every element of $\mathcal{W}$. Reply YES if it ever outputs one and reply NO otherwise.
2) Let $\tilde{v}_i$ be the number of times we reply YES on the i'th query in the above simulation (i.e., summed over all $y \in B$).
3) If $\tilde{v}_i \ne v_{i,B}$ for some i, then output $\perp$. (To do this, we hardcode in the values of $v_{i,B}$ for all i.)
4) Otherwise, output the simulated output of $C^{\mathsf{SAT}}(x)$

It is easy to see that we can construct a D_x of size at most $\mathsf{poly}(|C|,|B|)$. Furthermore, it is easy to see that we have that both

$$D_x(\mathcal{W}) \ne \perp \implies f(x) = D_x(\mathcal{W})$$

and

$$D_x(\mathcal{W}) \ne \perp \text{ for some } \mathcal{W}.$$

It remains to construct Gen. Let $m \in [n]$ be a parameter we choose later. We choose buckets to be sets of the form

$$B[z] = \{x \in \{0,1\}^n : \text{the first } (n-m) \text{ bits of } x \text{ are equal to } z\},$$

where $z \in \{0,1\}^{n-m}$.

Then given z, $x \in B[z]$ and the values of $v_{i,B[z]}$ for all i, one can construct D_x in time $\mathsf{poly}(|C|, |B[z]|) = \mathsf{poly}(|C|, 2^m)$. Furthemore, each $v_{i,B[z]}$ is an integer between 0 and 2^m, so it requires m bits to specify. Hence, using the trivial DNF construction, there is a circuit of size $mn2^{n-m}|C| \leq n^2 2^{n-m}|C|$ that takes as input z and outputs $v_{i,B[z]}$ for all i. Thus, Gen has size at most

$$n^2 2^{n-m}|C| + \mathsf{poly}(|C|, 2^m).$$

□

We will also use the fact that being somewhat $\mathsf{NP} \cap \mathsf{coNP}$ computable is closed under non-adaptive oracle calls.

Proposition IV.3. *Let f be (S,s)-somewhat $\mathsf{NP} \cap \mathsf{coNP}$-computable. If f' is computable by a t-size f-oracle circuit C making q non-adaptive queries, then f' is (S',s')-somewhat $\mathsf{NP} \cap \mathsf{coNP}$-computable where*

$$S' = t + q \cdot S + \mathsf{poly}(q)$$

and

$$s' = t + q \cdot s + \mathsf{poly}(q).$$

Moreover, given the corresponding S-size Gen for f and the circuit C, one can generate the corresponding S'-size Gen' for f' in uniform polynomial-time.

Proof. Let Gen be the circuit witnessing that f is (S,s)-somewhat $\mathsf{NP} \cap \mathsf{coNP}$-computable. The proposition follows from the following construction of Gen' for f'.

On input x:

1) Run $C(x)$ to generate the q oracle queries $y_1, \ldots, y_q$.
2) Let $D_i = \mathrm{Gen}(y_i)$ for all $i \in [q]$.
3) Output the circuit D'_x that takes inputs to each D_i and does the following. Output $\perp$ if D_i outputs $\perp$ for some $i \in [q]$. Otherwise simulate running $C(x)$ using the output of D_i to answer the i'th query and output the simulated answer.

□

V. Proof of NP-hardness of MCSP

A. Reduction

We present the formal description of our reduction from Gap τ-Uniform Vertex Cover to MCSP below:

- **Parameterized by:**
  - a formula size value $|\psi| \leq n$,
  - an integer "security parameter" $1 \leq \lambda \leq 2^n$,
  - the hypergraph uniformity $\tau \leq \log n$,
  - a function $f\colon [n] \times [\lambda] \to \{0,1\}$ that is (Q,q)-somewhat $\mathsf{NP} \cap \mathsf{coNP}$-computable function by a Q-sized circuit Gen_f and $q \leq n$,
  - an ϵ-secure non-interactive witness indistinguishable proof system NIWI with its security parameter set to $|\psi|$, and
  - a size threshold S.
- **Notation:**
  - Let $H\colon [n] \times [\lambda'] \to \{0,1\}$ be the function given by

$$H(v,i) = \mathrm{Amp}\left(f(v,\cdot), \frac{1}{5\tau}\right)(i),$$

    where $\mathrm{Amp}(\cdot)$ is the hardness amplification procedure of Lemma III.5. Note that $\lambda' \leq \lambda^{O(\tau^2)}$. By Proposition IV.3, note that H is (Q',q')-somewhat $\mathsf{NP} \cap \mathsf{coNP}$-computable by some Q'-sized circuit Gen where $Q' = O(Q\tau^2)$ and $q' = O(q\tau^2)$. Moreover, Gen can be produced in uniform $\mathsf{poly}(Q,\tau)$-time given Gen_f and τ.
  - For a τ-uniform hypergraph on n vertices with edge set E and a formula ψ, define the function

$$f_{\psi,E}\Big(x = (e \in [n]^\tau, \{r_v \in [\lambda']\}_{v \in e}, \{c_v \in \{0,1\}\}_{v \in e}),$$
$$\pi \in \{0,1\}^{\mathsf{poly}(|\psi|, q\tau)}\Big) =$$
$$\begin{cases} c_w \oplus H(w, r_w) & \text{if } \mathsf{NIWI.Verify}(\psi \vee \varphi_x, \pi) = 1 \\ & \text{and } e \in E \text{ where } \varphi_x \text{ is} \\ & \text{defined below and } w \text{ is the} \\ & \text{first entry of } e \\ 0 & \text{otherwise.} \end{cases}$$

    To define φ_x, let $D_v = \mathrm{Gen}(v, r_v)$ for all $v \in e$. We define φ_x as the $\mathsf{poly}(q,\tau)$-sized formula that is satisfiable if and only if there exist $b \in \{0,1\}$ and $\{w_v\}_{v \in e}$ such that for all $v \in e$ we have $D_v(w_v) = b \oplus c_v$.
- **Reduction:**
  Given a τ-uniform hypergraph on n vertices with edge set E:
  1) Accept if and only if for every unsatisfiable formula ψ of size at most $|\psi|$ (we iterate through such formulas by brute force) $f_{\psi,E}$ has complexity at most S.

B. Hardness

By construction, we have the following bound on the running time of the reduction (including the $n^{O(\log \tau)}$ time to run the reduction from SAT to Gap τ-Uniform Vertex Cover).

Proposition V.1. *The reduction can be implemented in uniform time $(n\lambda)^{O(\tau^2)} 2^{\mathsf{poly}(|\psi| q\tau)}$, given Gen_f.*

In Section V-C, we prove the following upper bound of the complexity of $f_{\psi,E}$ when E is a YES instance.

Lemma V.2 (Upper Bound on YES Instances)**.** *If ψ is unsatisfiable and E is the edge set of a YES instance of Gap τ-Uniform Vertex Cover, then*

$$\mathsf{CC}(f_{\psi,E}) \le O\left(\frac{\gamma \cdot n \cdot \lambda}{\tau \log \lambda} + \tau^3 Q\right) + \mathsf{poly}(n^\tau).$$

In Section V-D, we prove the following lower bound of the complexity of $f_{\psi,E}$ when E is a NO instance.

Lemma V.3 (Lower Bound on NO Instances)**.** *There is a universal constant c such that the following holds. Assume that*

- UNSAT *on formulas of size $|\psi|$ requires non-deterministic circuits of size greater than $(n\lambda' t)^c$,*
- $\epsilon(|\psi|) < \frac{1}{(n\lambda t)^2}$*, and*
- n *is sufficiently large.*

Then, for every edge set E of a NO instance of Gap τ-Uniform Vertex Cover, there exists an unsatisfiable $|\psi|$-size formula ψ such that

$$\mathsf{K}^t(f_{\psi,E}) \ge \mathsf{K}^{\mathsf{poly}(t\log(n\lambda'))}(f) - 4\sqrt{\frac{2\gamma}{\tau}}n\lambda - \mathsf{poly}(n\tau).$$

The proof of the main results will be presented in Section V-E.

C. YES instance analysis

We now prove Lemma V.2.

Lemma V.2 (Upper Bound on YES Instances)**.** *If ψ is unsatisfiable and E is the edge set of a YES instance of Gap τ-Uniform Vertex Cover, then*

$$\mathsf{CC}(f_{\psi,E}) \le O\left(\frac{\gamma \cdot n \cdot \lambda}{\tau \log \lambda} + \tau^3 Q\right) + \mathsf{poly}(n^\tau).$$

Proof. Fix an optimal vertex cover $V \subseteq [n]$ of size $\mathrm{OPT} \le \gamma\frac{n}{\tau}$. We can compute $f_{\psi,E}$ as follows.

On input $x = (e, \{r_v\}_{v\in e}, \{c_v\}_{v\in e})$ and π:

1) Check if $e \in E$.
2) Compute $D_v = \mathrm{Gen}(v, r_v)$ for all $v \in e$.
3) Construct φ_x.
4) Reject if $\mathsf{NIWI.Verify}(\psi \vee \varphi_x, \pi) \neq 1$.
5) Fix the lexicographically first vertex $v' \in e$ in the vertex cover V.
6) Compute $H(v', r_{v'})$. (We will do this step carefully to make sure it is efficient.)
7) Output $c_{v'} \oplus H(v', r_{v'})$.

It is easy to see that this procedure does indeed compute $f_{\psi,E}$ as long as we have that

$$c_{v'} \oplus H(v', r_{v'}) = c_w \oplus H(w, r_w)$$

where w (as in the reduction) is the first entry of e. This is true because of the following argument. Since $\mathsf{NIWI.Verify}(\psi \vee \varphi_x, \pi) = 1$, we get (by the perfect soundness of the NIWI) that "either ψ or φ_x is satisifable." Then because ψ is unsatisifable (by the assumption in this lemma), we get that φ_x is satisfiable. Then by construction of φ_x, we get that

$$c_{v'} \oplus H(v', r_{v'}) = c_v \oplus H(v, r_v)$$

for all $v \in e$, in particular including w. This completes our argument that this procedure computes $f_{\psi,E}$.

Thus, it just remains to bound the complexity of each step in the procedure.

1) This step can be done by a DNF of size $\mathsf{poly}(n^\tau)$.
2) This step can be done by a circuit of size $O(\tau Q') = O(\tau^3 Q)$ essentially by construction.
3) By construction of φ_x, this can be done in size $\mathsf{poly}(q, \tau)$.
4) By the efficiency of the NIWI, this can be done in size $\mathsf{poly}(|\pi| + |\psi| + \tau q) = \mathsf{poly}(|\psi|, \tau q) = \mathsf{poly}(n)$.
5) This step can be done in size $\mathsf{poly}(n)$.
6) Let $H|_V \colon V \times [\lambda'] \to \{0,1\}$ be the restriction of H to its first input being in V. The circuit complexity of this step in the procedure is exactly the complexity of computing $H|_V$.
   To compute $H|_V$, it will be helpful to compute another function. Set $k = \lfloor \log^3 n \rfloor$. Let $f^k|_V \colon (V \times [\lambda])^k \to \{0,1\}^k$ be given by
   $$f^k|_V(v_1, i_1, \ldots, v_k, i_k) = f(v_1, i_1) \cdots f(v_k, i_k).$$
   Since $\lambda \ge n$, for sufficiently large n we have that
   $$\begin{aligned} k = \lfloor \log^3 n \rfloor &\le n^{1/\log^2 \log(2n)} \\ &\le (\mathrm{OPT} \cdot \lambda)^{1/\log^2 \log(2\mathrm{OPT}\cdot\lambda)}. \end{aligned}$$
   Hence, we can apply Uhlig's theorem (Theorem III.7) to say there is a circuit for $f^k|_V$ of size at most
   $$4\frac{\mathrm{OPT} \cdot \lambda}{\log(\mathrm{OPT} \cdot \lambda)}.$$
   Now recall $H(v, \cdot) = \mathrm{Amp}(f(v, \cdot), \frac{1}{5\tau})$. By the properties of Amp this means that there a circuit of size $\mathsf{poly}(n\tau)$ computes $H(v, \cdot)$ given $O(\tau^2) = O(\log^2 n)$ non-adaptive oracle queries to $f(v, \cdot)$. Thus, for sufficiently large n, we can answer the non-adaptive $f(v, \cdot)$ queries using our circuit for $f^k|_V$ whenever $v \in V$. Thus, there is a circuit H_V of size at most
   $$\mathsf{poly}(n\tau) + \frac{\mathrm{OPT} \cdot \lambda}{\log(\mathrm{OPT} \cdot \lambda)}.$$
7) We can do this with $O(1)$ gates.

Thus, in total we obtain the bound

$$\mathsf{CC}(f_{\psi,E}) \le \mathsf{poly}(n^\tau) + O\left(\tau^3 Q + \frac{\mathrm{OPT} \cdot \lambda}{\log(\mathrm{OPT} \cdot \lambda)}\right),$$

which proves the lemma plugging in $\mathrm{OPT} \le \gamma\frac{n}{\tau}$. □

D. NO instance analysis

We begin by introducing some notation.

- For φ_x as in the code of the reduction, let w_{φ_x} be the lexicographically first witness to φ_x (which will always exist in the cases we refer to it).
- For $0 \le p \le 1$, we say a Turing machine A is *p-successful* on $f_{\psi,E}$ if for all $e \in E$ we have that

$$\Pr_{\substack{b \leftarrow \{0,1\},\\ r_v \leftarrow [\lambda'] \text{ for all } v \in e\\ c_v = b \oplus H(v,r_v) \text{ for all } v \in e\\ \pi \leftarrow \mathsf{NIWI.Prove}(\varphi_x \vee \psi, w_{\varphi_x})}} [A\left(e, \{r_v\}_{v\in e}, \{c_v\}_{v\in e}\ , \pi\right) = b]$$

  is at least p where x is defined as $(e, \{r_v \in [\lambda']\}_{v\in e}, \{c_v \in \{0,1\}\}_{v\in e})$.
- We let $|A|$ be short hand for $|d|$ where our universal Turing machine $U(d,i)$ computes $A(i)$.

By definition of $f_{\psi,E}$, we get the following.

Proposition V.4. *For all $0 \le p \le 1$, every A that computes $f_{\psi,E}$ is also p-successful on $f_{\psi,E}$.*

Now we prove a lower bound on the size of any successful Turing machine if one chooses a satisfiable ψ.

Lemma V.5. *Let ψ be a satisfiable formula. Let E be the edge set of a NO instance of Gap τ-Uniform Vertex Cover. Assume $\epsilon(|\psi|) < \frac{1}{(n\lambda t)^2}$. If A is .9-successful on $f_{\psi,E}$ and runs in t-time, then*

$$|A| \ge \mathsf{K}^{\mathsf{poly}(t\log(n\lambda'))}(f) - 4\sqrt{\frac{2\gamma}{\tau}}n\lambda - \mathsf{poly}(n\tau).$$

Proof. For contradiction suppose the bound does not hold. Fix an arbitrary edge $e \in E$. We now use a hybrid argument. Let w_ψ be the lexicographically first witness to ψ. First, for sufficiently large n we have

$$\epsilon(|\psi|) \le \frac{1}{(n\lambda t)^2} \le \frac{1}{t^2 \cdot (\mathsf{K}^{t+\mathsf{poly}(n\lambda')}(f))^2} \le \frac{1}{t^2 \cdot |A|^2},$$

so the security of the NIWI says we can replace π to come from the witness for ψ and get

$$\Pr_{\substack{b \leftarrow \{0,1\},\\ r_v \leftarrow [\lambda'] \text{ for all } v \in e\\ c_v = b \oplus H(v,r_v) \text{ for all } v \in e\\ \pi \leftarrow \mathsf{NIWI.Prove}(\varphi_x \vee \psi, w_\psi)}} [A\left(e, \{r_v\}_{v\in e}, \{c_v\}_{v\in e}\ , \pi\right) = b] \ge .9 - \epsilon(|\psi|) \ge .89,$$

for sufficiently large n.

On the other hand if we replace all the c_v with uniformly random bits, then the chance of outputting b must be one half (because b is independent of all the inputs to A). In other words,

$$\Pr_{\substack{b \leftarrow \{0,1\},\\ r_v \leftarrow [\lambda'] \text{ for all } v \in e\\ c_v \leftarrow \{0,1\} \text{ for all } v \in e\\ \pi \leftarrow \mathsf{NIWI.Prove}(\varphi_x \vee \psi, w_\psi)}} [A\left(e, \{r_v\}_{v\in e}, \{c_v\}_{v\in e}\ , \pi\right) = b] = 1/2,$$

Thus, by a hybrid argument and the distinguisher-to-predictor lemma (Lemma III.3), for some $v \in e$ we must have that

$$\mathsf{K}^{\mathsf{poly}(\log(n\lambda')),A}_{1/2+1/(3\tau)}(H_v) \le O(\tau \log \lambda') + |\pi| \le \mathsf{poly}(n\tau),$$

where H_v denotes the truth table of $H(v,\cdot)$. Hence by the construction of H and the properties of Amp (Lemma III.5), we get that

$$\mathsf{K}^{\mathsf{poly}(\log(n\lambda')),A}_{1-1/\tau}(f_v) \le \mathsf{poly}(n\tau), \tag{3}$$

where f_v denote the truth table of $f(v,\cdot)$, and we used that $\tau \le \log n$ and $\lambda \le 2^n$

Now, recall we picked an arbitrary $e \in E$. Thus, for every $e \in E$, there exists a vertex $v \in e$ such that (3) holds. Since the smallest vertex cover is of size at least $n(1-\frac{\gamma}{\tau})$, we then get that (3) holds for all but a $\frac{\gamma}{\tau}$ fraction of vertices. This immediately implies that

$$\mathsf{K}^{\mathsf{poly}(\log(n\lambda')),A}_{1-2\gamma/\tau}(f) \le \mathsf{poly}(n\tau).$$

By hardcoding in corrections to the small number of errors (Theorem III.4) we get (noting $\tau \le \log n$) that

$$\mathsf{K}^{\mathsf{poly}(\log(n\lambda')),A}(f) \le 4\sqrt{\frac{2\gamma}{\tau}}n\lambda + \mathsf{poly}(n\tau).$$

Hence, using that A runs in time t, we have that

$$\mathsf{K}^{\mathsf{poly}(t\log(n\lambda'))}(f) \le |A| + 4\sqrt{\frac{2\gamma}{\tau}}n\lambda + \mathsf{poly}(n\tau),$$

which proves the lemma. □

We can then exploit the non-deterministic hardness of solving UNSAT to prove the following lemma.

Lemma V.3 (Lower Bound on NO Instances)**.** *There is a universal constant c such that the following holds. Assume that*

- UNSAT *on formulas of size $|\psi|$ requires non-deterministic circuits of size greater than $(n\lambda' t)^c$,*
- $\epsilon(|\psi|) < \frac{1}{(n\lambda t)^2}$*, and*
- n *is sufficiently large.*

Then, for every edge set E of a NO instance of Gap τ-Uniform Vertex Cover, there exists an unsatisfiable $|\psi|$-size formula ψ such that

$$\mathsf{K}^t(f_{\psi,E}) \ge \mathsf{K}^{\mathsf{poly}(t\log(n\lambda'))}(f) - 4\sqrt{\frac{2\gamma}{\tau}}n\lambda - \mathsf{poly}(n\tau).$$

Proof. Set

$$S = \mathsf{K}^{\mathsf{poly}(t\log(n\lambda'))}(f) - 4\sqrt{\frac{2\gamma}{\tau}}n\lambda - \mathsf{poly}(n\tau).$$

so that Lemma V.5 holds. Note that we always have that $S \le O(n\lambda)$.

We prove the contrapositive. Assume there exists an edge set E that is a NO instance of n-vertex Gap τ-Uniform Vertex Cover such that for every $|\psi|$-size unsatisfiable ψ,

$$\mathsf{K}^t(f_{\psi,E}) < S. \tag{4}$$

On the other hand, Lemma V.5 says that for every satisfiable ψ every t-time machine A that is .9-successful on $f_{\psi,E}$ has

$$|A| \ge S. \tag{5}$$

Hence, the following yields a randomized non-deterministic circuit for checking whether ψ is unsatisfiable:

Given ψ:

1) Non-deterministically guess an S-size Turing machine A
2) Using randomness, empirically estimate to additive error .01 (allowing failure probability at most $1/3$) the probability

$$\Pr[A\,(e, \{r_v\}_{v\in e}, \{c_v\}_{v\in e}\ , \pi) = b \text{ in } t\text{-time}]$$

where $b \leftarrow \{0,1\}$, $r_v \leftarrow [\lambda']$ for all $v \in e$, $c_v = b \oplus H(v, r_v)$ for all $v \in e$, and $\pi \leftarrow \mathsf{NIWI.Prove}(\varphi_x \vee \psi, w_{\varphi_x})$. Reject if this probability is less than .95. In order to do this empirical estimate efficiently, we hardcode in the the truth table of H and also hardcode in the lexicographically first witnesses to each circuit output by $\mathrm{Gen}(v, r_v)$ for all $v \in [n]$ and $r_v \in [\lambda']$. From the latter information, we can compute w_{φ_x} easily for any φ_x.

3) Accept (i.e., output that ψ is unsatisfiable.)

From (4) and the fact (Proposition V.4) that any A computing $f_{\psi,E}$ is 1-successful, we get that this circuit accepts every unsatisfiable ψ. From (5), the circuit rejects all satisfiable ψ.

Thus, we just need to argue for efficiency. Step 1 requires size $O(S) = \mathsf{poly}(n\lambda)$. Step 2 can be done in size $\mathsf{poly}(qn\lambda' t) = \mathsf{poly}(n\lambda' t)$. Thus, this circuit has size at most $\mathsf{poly}(n\lambda' t)$.

Finally, using Adleman's trick we can remove the randomness from this non-deterministic circuit to obtain a (standard) non-deterministic circuit of size at most $\mathsf{poly}(n\lambda' t|\psi|) = \mathsf{poly}(n\lambda' t)$. □

E. Proof of the Main Results

We need one additional lemma, which essentially states that, given a function with nearly maximum circuit complexity, one can construct a string with high time-bounded Kolmogorov complexity.

Lemma V.6. *For every oracle A, the following are equivalent.*

1) $\mathsf{P}^A/\mathsf{poly} \not\subseteq \mathsf{i.o.E}/\delta 2^n$ *for some constant* $\delta > 0$.
2) $\mathsf{P}^A/\mathsf{poly} \not\subseteq \mathsf{i.o.SIZE}(\delta 2^n/n)$ *for some constant* $\delta > 0$.

Proof. The high-level idea is this. (1) implies (2) because Turing machines can simulate circuits. (2) implies (1) because of the following argument. If one breaks the truth table of a high circuit complexity function f into small pieces, then at least one small piece must have have high time-bounded Kolmogorov complexity (relative to its length). If this were not then case, then one could build a small circuit for f.

Now we give the formal proof. Since any n-input circuit of size s can be encoded as a binary string of $O(s\log(s+n))$, it holds that $\mathsf{SIZE}(\delta 2^n/n) \subseteq \mathsf{E}/\alpha(n)$, where

$$\alpha(n) = O\left(\frac{\delta 2^n}{n} \cdot \log \frac{\delta 2^n}{n}\right) = O(\delta 2^n).$$

Thus, the first item implies the second item.

To see the converse, let $f = \{f_n : \{0,1\}^n \to \{0,1\}\}$ be a function in $\mathsf{P}^A/\mathsf{poly} \setminus \mathsf{i.o.SIZE}(\delta 2^n/n)$. This assumption is equivalent to saying that for all large $n \in \mathbb{N}$, the function f_n can be computed by some A-oracle circuit of size $\mathsf{poly}(n)$ but cannot be computed by any circuit of size $\delta 2^n/n$. Fix an arbitrary constant c. Let $\epsilon, \delta' > 0$ be sufficiently small constants chosen later. (We will construct some function outside $\mathsf{DTIME}(2^{cn})/\delta' 2^n$.) Fix sufficiently large $n \in \mathbb{N}$. For every $z \in \{0,1\}^{(1-\epsilon)n}$, define $f_{n,z} \colon \{0,1\}^{\epsilon n} \to \{0,1\}$ as the function such that $f_{n,z}(x) = f_n(z,x)$.

We claim that there exists some z such that $\mathsf{K}^t(f_{n,z}) \geq \delta' 2^{\epsilon n}$ for $t := 2^{c\epsilon n} \leq 2^{n/2}$ (we choose $\epsilon \leq c/2$). This is sufficient, as $f_{n,z}$ can be computed by an A-oracle circuit of size $\mathsf{poly}(n)$. To see the claim, assume for contradiction that for every z, it holds that $\mathsf{K}^t(f_{n,z}) < \delta' 2^n$. That is, for every z, there exists $d_z \in \{0,1\}^{\delta' 2^n}$ such that $U^t(d_z, x) = f_{n,z}(x)$. For every $i \in [\delta' 2^{\epsilon n}]$, let α_i be the function that takes $z \in \{0,1\}^{(1-\epsilon)n}$ and outputs the i'th bit of d_z. Then, α_i can be implemented by a circuit of size

$$(1+o(1)) \cdot \frac{2^{(1-\epsilon)n}}{\log 2^{(1-\epsilon)n}} = (1+\epsilon+o(1)) \cdot \frac{2^{(1-\epsilon)n}}{n}.$$

Thus, given z as input, the advice string d_z can be computed by a circuit of size

$$\delta' 2^{\epsilon n} \cdot O\left(\frac{2^{(1-\epsilon)n}}{n}\right) = O\left(\frac{\delta' 2^n}{n}\right).$$

Now we construct a small circuit that computes f_n as follows: Given input $w \in \{0,1\}^n$, let z and x be the first $(1-\epsilon)n$ bits and the last ϵn bits of w, respectively. Compute d_z by the circuit of size $O(\delta' 2^n/n)$, and simulate and output $U^t(d_z, x)$. This algorithm can be implemented by a circuit of size

$$O\left(\delta' \frac{2^n}{n}\right) + \widetilde{O}(t) \leq \delta 2^n,$$

where the last inequality holds by choosing a sufficiently small $\delta' > 0$. This is a contradiction. □

Now we are ready to prove our main theorem.

Theorem V.7 (a restatement of Theorem I.1)**.** *Assume all of the following:*

1) *subexponentially-secure NIWIs exist,*
2) $\mathsf{coNP} \not\subseteq \mathsf{i.o.NSIZE}\left(2^{n^\epsilon}\right)$ *for some constant* $\epsilon > 0$*, and*
3) $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly} \not\subseteq \mathsf{i.o.SIZE}(\delta 2^n/n)$ *for some constant* $\delta > 0$.

Then, for any constant $g \geq 1$ and any polynomial t, it is NP*-hard under deterministic quasipolynomial-time non-adaptive reductions to solve the following promise problem Π:*

Input: *the truth table of* $f \colon \{0,1\}^n \to \{0,1\}$ *and a size parameter* $S \leq 2^n/n$.
Yes: $\mathsf{CC}(f) \leq S$ *and* $\mathsf{K}^{O(S\log S)}(f) \leq S\log S$.
No: $\mathsf{CC}(f) > S \cdot g$ *and* $\mathsf{K}^{t(S)}(f) > S\log S \cdot g$.

In particular, both MCSP *and* K^t *are* NP*-hard to approximate to within a constant factor.*

Proof. By Lemma V.6, the third assumption implies that for any polynomial t, there exists a function $f = \{f_m : \{0,1\}^m \to \{0,1\}\}_{m\in\mathbb{N}}$ such that f_m is computable by a SAT-oracle circuit C^{SAT} of size $\mathsf{poly}(m)$ and $\mathsf{K}^{t(2^m)}(f_m) \geq$

$\delta 2^m$. It follows from Proposition IV.2 that f_m is (Q, q)-somewhat $\mathsf{NP} \cap \mathsf{coNP}$-computable by some Q-size circuit $\mathrm{Gen}_{C^{\mathsf{SAT}}}$, where $Q = 2^m/m^5$ and $q = \mathsf{poly}(m)$. We may regard $f_m \colon [n] \times [\lambda] \to \{0,1\}$, where $2^m = n \cdot \lambda$ for parameters n and λ chosen later.[8]

Our final reduction builds on the reduction presented in Section V-A. Let $([n], E)$ be an instance of Gap τ-Uniform Vertex Cover. We enumerate all the SAT-oracle circuits C^{SAT} of size $\mathsf{poly}(m)$, and output "Yes" if and only if for any SAT-oracle m-input circuit C^{SAT} of size $\mathsf{poly}(m)$, all the queries $(f_{\psi,E}, S)$ of the reduction of Section V-A given the generator $\mathrm{Gen}_{C^{\mathsf{SAT}}}$ are answered with "Yes" by the oracle that solves Π, where $f_{\psi,E}$ is the function defined in Section V-A. Note that this reduction can be implemented in time $2^{\mathsf{poly}(m)}$, where $m \leq \mathsf{poly}(\log n)$. We will choose the parameters during the analysis of this reduction.

To see the completeness, let E be the edge set of a YES instance of Gap τ-Uniform Vertex Cover. By Lemma V.2,

$$\mathsf{CC}(f_{\psi,E}) \leq O\left(\frac{\gamma \cdot n \cdot \lambda}{\tau \log \lambda} + \tau^3 Q\right) + \mathsf{poly}(n^\tau)$$

Since $\tau \leq \log n \leq m$, we have $\tau^3 Q \leq 2^m/m^2 \leq n \cdot \lambda/(\log \lambda)^2$. Let c be a sufficiently large constant. We set $\lambda = n^{c\tau}$. Then we have $\mathsf{poly}(n^\tau) \leq \lambda/(\log \lambda)^2$ by choosing a sufficiently large c. Since $\tau \leq \log \lambda$, it follows that

$$\begin{aligned}\mathsf{CC}(f_{\psi,E}) &\leq O\left(\frac{\gamma \cdot n \cdot \lambda}{\tau \log \lambda} + \tau^3 Q\right) + \mathsf{poly}(n^\tau) \\ &\leq O\left(\frac{n\lambda}{\tau \log \lambda}\right) = S,\end{aligned}$$

where we define S so that this equality holds. This further implies that

$$\mathsf{K}^{O(n\lambda)}(f_{\psi,E}) \leq O\left(\frac{n\lambda}{\tau \log n} \log\left(\frac{n\lambda}{\tau \log n}\right)\right) \leq O\left(\frac{n\lambda}{\tau}\right).$$

Now we argue for soundness. Let E be the edge set of a NO instance of Gap τ-Uniform Vertex Cover. Let $t = t(2^m)$. Let C^{SAT} be the SAT-oracle circuit of size $\mathsf{poly}(m)$ that computes f_m such that $\mathsf{K}^{\mathsf{poly}(t)}(f_m) \geq \delta 2^m = \delta n \lambda$, where poly is some polynomial chosen later. It suffices to claim that the reduction of Section V-A given $\mathrm{Gen}_{C^{\mathsf{SAT}}}$ queries the function $f_{\psi,E}$ that is answered with "No" for some unsatisfiable formula ψ. Let c_0 be the universal constant of Lemma V.3. To satisfy the first assumption of Lemma V.3, we choose $|\psi|$ so that $2^{|\psi|^\epsilon} \geq \left(n\lambda^{O(\tau^2)}t\right)^{c_0}$. For all sufficiently large n, Lemma V.3 implies that

$$\begin{aligned}\mathsf{K}^t(f_{\psi,E}) &\geq \mathsf{K}^{\mathsf{poly}(t \log(n\lambda'))}(f) - 4\sqrt{\frac{2\gamma}{\tau}} n\lambda - \mathsf{poly}(n\tau) \\ &\geq \delta n\lambda - \frac{\delta}{2} \cdot n\lambda \\ &= \frac{\delta}{2} \cdot n\lambda,\end{aligned}$$

where the second inequality holds by choosing a sufficiently large $\tau \geq O(\delta^{-2})$. This further implies that

$$\mathsf{CC}(f_{\psi,E}) \geq \Omega\left(\delta \cdot \frac{n\lambda}{\log(n\lambda)}\right) \geq \Omega(\tau\delta \cdot S) > g \cdot S,$$

where the last inequality holds by choosing a sufficiently large constant τ. □

[8] Strictly speaking, we choose $2^{m-1} \leq n\lambda \leq 2^m$, and the truth table of length $n\lambda$ is constructed from the first $n\lambda$ bits of the truth table of f_m.

Remark V.8. *Choosing $\tau := (\log n)^\epsilon$ for a small constant $\epsilon > 0$, the inapproximability factor of Theorem V.7 can actually be improved to $(\log |f|)^{\Omega(1)}$. Moreover, the third assumption can be weakened to $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly} \not\subseteq \mathsf{i.o.SIZE}(2^n/n^{1+\Omega(1)})$.*

We may remove the third assumption using a case analysis.

Corollary V.9 (The formal version of Corollary I.3)**.** *Assume that*

1) *subexponentially-secure NIWIs for SAT exist, and*
2) *coNP requires subexponential-size non-deterministic circuits almost everywhere.*

Then, for every function $f = \{f_n : \{0,1\}^n \to \{0,1\}\}_{n \in \mathbb{N}}$ in NP and every polynomial p, there exist infinitely many $n \in \mathbb{N}$ and an n-input subexponential-size non-adaptive oracle circuit C such that

$$\Pr_{x \leftarrow \{0,1\}^n}\left[f_n(x) = C^{\mathsf{MCSP}}(x)\right] \geq 1 - \frac{1}{p(n)}.$$

Proof. A high-level idea is to consider two cases: whether $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly} \not\subseteq \mathsf{i.o.SIZE}(2^n/2n)$ or not. If it is, we conclude from Theorem I.1 that MCSP is NP-hard (in the worst case). Otherwise, we will show that an MCSP oracle can be used to break the security of a pseudorandom generator based on a candidate hard function $f \in \mathsf{NP}$, from which we obtain an average-case MCSP-oracle circuit to compute f. Details follow.

Let $\epsilon > 0$ be an arbitrary small constant. Fix $n \in \mathbb{N}$. Define $m = m(n) := n^\epsilon$. As in Lemma III.5, we define $g := f^{\oplus k}$, i.e., the k-wise XOR function of f, where $k = O(m(n)/p(n))$. Let $\mathrm{NW}^g \colon \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^{2^m}$ be the Nisan–Wigderson generator [91] instantiated with a candidate hard function g and the combinatorial design $\{S_i\}_{i \in [2^m]}$ based on polynomials. For each seed z, we regard $\mathrm{NW}^g(z)$ as the truth table of a function $h_z \colon \{0,1\}^m \to \{0,1\}$ on m-bit inputs. As observed in [12], h_z can be computed by a g-oracle circuit of size $\mathsf{poly}(n)$ because the i'th set S_i in the combinatorial design can be computed by a polynomial-size circuit given i as input. In particular, h_z can be computed by a polynomial-size SAT-oracle circuit.

Consider the case where for all sufficiently large $n \in \mathbb{N}$, it holds that $\mathsf{CC}(\mathrm{NW}^g(z)) \geq \frac{2^m}{2m}$ for some z. In this case, the function h_z defined by its truth table $\mathrm{NW}^g(z)$ has circuit complexity at least $2^m/2m$, and moreover h_z can be computed by a polynomial-size SAT-oracle circuit. Thus, we obtain $\mathsf{P}^{\mathsf{NP}}/\mathsf{poly} \not\subseteq \mathsf{i.o.SIZE}(2^n/2n)$. By Theorem I.1, MCSP is NP-hard under quasi-polynomial-time non-adaptive reductions, and we are done.

Now consider the other case: Assume that for infinitely many $n \in \mathbb{N}$, it holds that $\mathsf{CC}(\mathrm{NW}^g(z)) < 2^m/2m$ for all z. Since $\mathsf{CC}(w) \geq 2^m/2m$ with high probability over a uniformly random $w \in \{0,1\}^{2^m}$, this implies that MCSP is a statistical test that distinguishes the output distribution of $\mathrm{NW}^g(\cdot)$ from the uniform distribution. By the reconstruction property of the Nisan–Wigderson pseudorandom generator, there exists an oracle circuit C_0 of size $2^{O(m)}$ such that

$$\Pr_{y \leftarrow \{0,1\}^{\mathsf{poly}(n)}}\left[g(y) = C_0^{\mathsf{MCSP}}(y)\right] \geq \frac{1}{2} + 2^{-m}.$$

By the property of hardness amplification (Yao's XOR lemma; see the proof of Lemma III.5), we obtain an oracle circuit C of size $2^{O(m)} = 2^{O(n^\epsilon)}$ such that

$$\Pr_{x \leftarrow \{0,1\}^n}\left[f(x) = C^{\mathsf{MCSP}}(x)\right] \geq 1 - \frac{1}{p(n)},$$

as desired. □

APPENDIX

In this section, we justify one of our assumptions under the random oracle model.

Theorem A.1. *Let $\mathcal{O}$ be a random oracle. Then,*

$$\Pr_{\mathcal{O}}\left[\mathsf{NP}^{\mathcal{O}} \text{ is not in } \mathsf{DTIME}[2^{n^2}]^{\mathcal{O}}/(\delta 2^n) \text{ infinitely often}\right] = 1$$

for some constant[9] *$\delta > 0$.*

Lemma A.2 (Borel–Cantelli)**.** *Let E_n be a sequence of events. If $\sum_n \Pr[E_n]$ converges, then*

$$\Pr[E_n \text{ occurs for at most finitely many } n] = 1$$

Proof of Theorem A.1. Without loss of generality, we can view the oracle $\mathcal{O}$ as taking as input pairs (x, y) where $|y| = |x|^3$ and outputting a string of length $|x|^3 + 1$. Our hard language will be the following:

$$L^{\mathcal{O}} = \{x : \mathcal{O}(x, y) \text{ outputs all zeroes for some } y\}.$$

It is easy to see that $L^{\mathcal{O}} \in \mathsf{NP}^{\mathcal{O}}$.

It remains to show the lower bound on $L^{\mathcal{O}}$. We make the following claim.

Claim A.3. *There is a constant n_0 such that the following holds. Let A be an oracle circuit that makes $q = 2^{2n^2}$ queries. If $n \geq n_0$, then*

$$\Pr_{\mathcal{O}}[A^{\mathcal{O}} \text{ computes } L^{\mathcal{O}} \text{ on all inputs of length } n] \leq (.9)^{2^n}.$$

Assuming Claim A.3 holds, we prove the desired lower bound. Let $A_1, A_2, A_3, \ldots$ be an enumeration of all (uniform) algorithms that run in time also take an advice string as input. Let E_n be the event that there exists an $i \in [n]$ and an advice string $\alpha \in \{0,1\}^{\delta 2^n}$ such that all the following holds:

- $A^{\mathcal{O}}(x, \alpha) = L^{\mathcal{O}}(x)$ for all $x \in \{0,1\}^n$, and
- $A^{\mathcal{O}}(x, \alpha)$ makes at most 2^{2n^2} oracle queries for all $x \in \{0,1\}^n$.

[9]We have not attempted to calculate or optimize for this value.

By Claim A.3 and a union bound, we have that

$$\Pr_{\mathcal{O}}[E_n] \leq n 2^{\delta 2^n} (.9)^{2^n} \leq O\left(\frac{1}{n^2}\right),$$

by setting δ to be a sufficiently small constant. Since $\sum_n \Pr[E_n] < \infty$, the Borel–Cantelli lemma (Lemma A.2) implies that

$$\Pr_{\mathcal{O}}[\text{ infinitely many } E_n \text{ occurs }] = 0.$$

By the definition of E_n, this implies that

$$\Pr_{\mathcal{O}}\left[L^{\mathcal{O}} \text{ is in } \mathsf{DTIME}[2^{n^2}]^{\mathcal{O}}/(\delta 2^n) \text{ infinitely often}\right] = 0.$$

Thus, it just remains to prove Claim A.3. To do so, we introduce some notation. Let $\tilde{\mathcal{O}}$ denote a partial function with the same input/output behavior as $\mathcal{O}$ (the difference is that $\tilde{\mathcal{O}}$ can output $\star$ for "undefined" values). We say that $\mathcal{O}$ *agrees* with $\tilde{\mathcal{O}}$ if they agree on all values where they are defined. We say that $\tilde{O}$ is *good for* x if both of the following hold:

- $\tilde{\mathcal{O}}(x, y)$ is not all zeroes (i.e., $\tilde{\mathcal{O}}(x, y) \notin \{0\}^*$) for any $y \in \{0,1\}^{|x|^3}$, and
- $\tilde{\mathcal{O}}(x, y) \neq \star$ for at most $q2^n$ many $y \in \{0,1\}^{|x|^3}$.

We make an intermediary claim.

Claim A.4. *Let $b \in \{0,1\}$. If $\tilde{\mathcal{O}}$ is good for $x \in \{0,1\}^n$ and $n \geq 10$ and $q \leq 2^{2n^2}$, then*

$$\Pr_{\mathcal{O}}\left[L^{\mathcal{O}}(x) = b \,\middle|\, \mathcal{O} \text{ agrees with } \tilde{\mathcal{O}}\right] \leq 3/4.$$

Proof. Since $\tilde{\mathcal{O}}$ is good for x, a union bound implies

$$\Pr_{\mathcal{O}}\left[L^{\mathcal{O}}(x) = 1 \,\middle|\, \mathcal{O} \text{ agrees with } \tilde{\mathcal{O}}\right] \leq 2^{-n^3-1} 2^{n^3} = 1/2$$

On the other hand, we also have

$$\begin{aligned}
&\Pr_{\mathcal{O}}\left[L^{\mathcal{O}}(x) = 0 \,\middle|\, \mathcal{O} \text{ agrees with } \tilde{\mathcal{O}}\right] \\
&\leq (1 - 2^{-n^3-1})^{2^{n^3} - q2^n} \\
&\leq e^{-1/2 + 2^{3n^2 - n^3 - 1}} \leq 3/4
\end{aligned}$$

when $n \geq 10$. □

Inductively applying Claim A.4 yields the following intermediary claim.

Claim A.5. *Let $f\colon \{0,1\}^n \to \{0,1\}$. Let $V \subseteq \{0,1\}^n$. If $\tilde{\mathcal{O}}$ is good for every element of V and $n \geq 10$ and $q \leq 2^{2n^2}$, then*

$$\Pr_{\mathcal{O}}\left[L^{\mathcal{O}}(x) \text{ computes } f(x) \,\middle|\, \mathcal{O} \text{ agrees with } \tilde{\mathcal{O}}\right] \leq (3/4)^{|V|}.$$

Proof. We use the lazy sampling technique. Imagine slowly revealing $\mathcal{O}$ as follows. We start the original $\tilde{\mathcal{O}}$. Then we reveal $\mathcal{O}$ on all inputs of the form (x, y) where x is the lexicographically first element of V. By Claim A.4, the probability that $f(x) = L^{\mathcal{O}}(x)$ is at most $3/4$.

Now remove x from V and consider the new partial function $\tilde{\mathcal{O}}$ that now includes the values we have revealed. Observe that we have preserved the invariant that (the updated) $\tilde{\mathcal{O}}$ is good

for every element of (the updated) V. This is because we only revealed values that began with the x we removed, and because the definition of good for x' only cares about inputs that start with x'. Repeating this argument until V is empty yields the bound. □

Finally, we prove Claim A.3

Proof of Claim A.3. We will again use the lazy sampling technique. Imagine we go over every $x \in \{0,1\}^n$ and reveal all the values of $\mathcal{O}$ that $A^{\mathcal{O}}(x)$ queries and set $f\colon \{0,1\}^n \to \{0,1\}$ to the function that $A^{\mathcal{O}}$ computes. In total, this reveals at most $2^n q$ values of $\mathcal{O}$. Let $\tilde{\mathcal{O}}$ be the corresponding partial function. Since so little values are defined, the only way for $\tilde{\mathcal{O}}$ not to be good for an $x \in \{0,1\}^n$ is if we revealed an $\mathcal{O}(x,y)$ equal to all zeroes.

The number of x for which this occurs is at most the number B of all zero outputs we have revealed. Observe that B is the sum of $q2^n \leq 2^{3n^2}$ independent Bernoulli random variables each with expectation 2^{-n^3-1}. So B has expectation at most $2^{-n^3+3n^2-1}$. Hence, by a Chernoff Bound, the probability that $|B| \geq 2^n/2$ is at most

$$\exp(-2^n/3)$$

for sufficiently large n.

On the other hand if $|B| \leq 2^n/2$, then by Claim A.5, the probability that f computes $L^{\mathcal{O}}$ on inputs of length n is at most

$$(3/4)^{2^n/2}.$$

Thus, we have that the probability f computes $L^{\mathcal{O}}$ on inputs of length n is at most

$$\max\{\exp(-2^n/3), (3/4)^{2^n/2}\} \leq (.9)^{2^n}$$

for sufficiently large n. □

□

Acknowledgment

We deeply thank Marshall Ball and Yael Tauman Kalai for helpful discussion.

References

[1] S. A. Cook, "The Complexity of Theorem-Proving Procedures," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 1971, pp. 151–158.

[2] L. A. Levin, "Universal sequential search problems," *Problemy Peredachi Informatsii*, vol. 9, no. 3, pp. 115–116, 1973.

[3] V. Kabanets and J. Cai, "Circuit minimization problem," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2000, pp. 73–79.

[4] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 1986, pp. 174–187.

[5] V. R. Pratt, "Every Prime has a Succinct Certificate," *SIAM J. Comput.*, vol. 4, no. 3, pp. 214–220, 1975.

[6] G. S. Frandsen and P. B. Miltersen, "Reviewing bounds on the circuit size of the hardest functions," *Inf. Process. Lett.*, vol. 95, no. 2, pp. 354–357, 2005.

[7] N. Mazor and R. Pass, "The Non-Uniform Perebor Conjecture for Time-Bounded Kolmogorov Complexity Is False," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2024, pp. 80:1–80:20.

[8] S. Hirahara, R. Ilango, and R. R. Williams, "Beating Brute Force for Compression Problems," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2024, pp. 659–670.

[9] E. Allender and B. Das, "Zero knowledge and circuit minimization," *Inf. Comput.*, vol. 256, pp. 2–8, 2017.

[10] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *SIAM J. Comput.*, vol. 18, no. 1, pp. 186–208, 1989.

[11] M. Tompa and H. Woll, "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 1987, pp. 472–482.

[12] M. L. Carmosino, R. Impagliazzo, V. Kabanets, and A. Kolokolova, "Learning Algorithms from Natural Proofs," in *Proceedings of the Conference on Computational Complexity (CCC)*, 2016, pp. 10:1–10:24.

[13] R. Impagliazzo and L. A. Levin, "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 1990, pp. 812–821.

[14] Y. Liu and R. Pass, "On One-way Functions and Kolmogorov Complexity," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2020, pp. 1243–1254.

[15] A. A. Razborov and S. Rudich, "Natural Proofs," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 24–35, 1997.

[16] I. C. Oliveira and R. Santhanam, "Hardness Magnification for Natural Problems," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 65–76.

[17] S. Hirahara, "Non-Disjoint Promise Problems from Meta-Computational View of Pseudorandom Generator Constructions," in *Proceedings of the Computational Complexity Conference (CCC)*, 2020, pp. 20:1–20:47.

[18] Y. Liu and R. Pass, "Characterizing Derandomization Through Hardness of Levin-Kolmogorov Complexity," in *Proceedings of the Computational Complexity Conference (CCC)*, 2022, pp. 35:1–35:17.

[19] S. Hirahara, "Non-Black-Box Worst-Case to Average-Case Reductions Within NP," *SIAM J. Comput.*, vol. 52, no. 6, pp. S18–349, 2023.

[20] R. Impagliazzo, "A Personal View of Average-Case Complexity," in *Proceedings of the Structure in Complexity Theory Conference*, 1995, pp. 134–147.

[21] R. Santhanam, "Pseudorandomness and the Minimum Circuit Size Problem," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2020, pp. 68:1–68:26.

[22] S. Hirahara, "Characterizing Average-Case Complexity of PH by Worst-Case Meta-Complexity," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2020, pp. 50–60.

[23] ——, "Average-case hardness of NP from exponential worst-case hardness assumptions," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2021, pp. 292–302.

[24] S. Hirahara and M. Nanashima, "On Worst-Case Learning in Relativized Heuristica," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2021, pp. 751–758.

[25] ——, "Finding Errorless Pessiland in Error-Prone Heuristica," in *Proceedings of the Computational Complexity Conference (CCC)*, 2022, pp. 25:1–25:28.

[26] S. Hirahara, "Symmetry of Information from Meta-Complexity," in *Proceedings of the Computational Complexity Conference (CCC)*, 2022, pp. 26:1–26:41.

[27] L. Chen, S. Hirahara, and N. Vafa, "Average-case Hardness of NP and PH from Worst-case Fine-grained Assumptions," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2022, pp. 67:1–67:17.

[28] H. Goldberg, V. Kabanets, Z. Lu, and I. C. Oliveira, "Probabilistic Kolmogorov Complexity with Applications to Average-Case Complexity," in *Proceedings of the Computational Complexity Conference (CCC)*, 2022, pp. 16:1–16:60.

[29] H. Goldberg and V. Kabanets, "A Simpler Proof of the Worst-Case to Average-Case Reduction for Polynomial Hierarchy via Symmetry of Information," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 007, 2022.

[30] ——, "Improved Learning from Kolmogorov Complexity," in *Proceedings of the Computational Complexity Conference (CCC)*, 2023, pp. 12:1–12:29.

[31] H. Ren and R. Santhanam, "Hardness of KT Characterizes Parallel Cryptography," in *Proceedings of the Computational Complexity Conference (CCC)*, 2021, pp. 35:1–35:58.

[32] Y. Liu and R. Pass, "On the Possibility of Basing Cryptography on EXP$\neq$ BPP," in *Proceedings of the International Cryptology Conference (CRYPTO)*, 2021, pp. 11–40.

[33] R. Ilango, H. Ren, and R. Santhanam, "Robustness of average-case meta-complexity via pseudorandomness," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2022, pp. 1575–1583.

[34] E. Allender, M. Cheraghchi, D. Myrisiotis, H. Tirumala, and I. Volkovich, "One-Way Functions and a Conditional Variant of MKTP," in *Proceedings of the Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2021, pp. 7:1–7:19.

[35] Y. Liu and R. Pass, "On One-Way Functions from NP-Complete Problems," in *Proceedings of the Computational Complexity Conference (CCC)*, 2022, pp. 36:1–36:24.

[36] S. Hirahara and R. Santhanam, "Excluding PH Pessiland," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2022, pp. 85:1–85:25.

[37] Y. Liu and R. Pass, "On One-Way Functions and Sparse Languages," in *Proceedings of the Theory of Cryptography Conference (TCC)*, 2023, pp. 219–237.

[38] ——, "One-Way Functions and the Hardness of (Probabilistic) Time-Bounded Kolmogorov Complexity w.r.t. Samplable Distributions," in *Proceedings of the International Cryptology Conference (CRYPTO)*, 2023, pp. 645–673.

[39] S. Hirahara, R. Ilango, Z. Lu, M. Nanashima, and I. C. Oliveira, "A Duality between One-Way Functions and Average-Case Symmetry of Information," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2023, pp. 1039–1050.

[40] S. Hirahara, "Capturing One-Way Functions via NP-Hardness of Meta-Complexity," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2023, pp. 1027–1038.

[41] S. Hirahara and M. Nanashima, "Learning in Pessiland via Inductive Inference," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2023, pp. 447–457.

[42] S. Hirahara, Z. Lu, and M. Nanashima, "Optimal Coding for Randomized Kolmogorov Complexity and Its Applications," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2024, pp. 369–378.

[43] M. Nanashima, "On Basing Auxiliary-Input Cryptography on NP-Hardness via Nonadaptive Black-Box Reductions," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2021, pp. 29:1–29:15.

[44] S. Hirahara and M. Nanashima, "One-Way Functions and Zero Knowledge," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2024, pp. 1731–1738.

[45] Z. Lu and R. Santhanam, "Impagliazzo's Worlds Through the Lens of Conditional Kolmogorov Complexity," in *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, 2024, pp. 110:1–110:17.

[46] M. Ball, Y. Liu, N. Mazor, and R. Pass, "Kolmogorov Comes to Cryptomania: On Interactive Kolmogorov Complexity and Key-Agreement," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2023, pp. 458–483.

[47] Y. Liu, N. Mazor, and R. Pass, "On White-Box Learning and Public-Key Encryption," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2025, pp. 73:1–73:22.

[48] W. J. Masek, "Some NP-complete set covering problems," *Unpublished manuscript*, 1979.

[49] E. Allender, L. Hellerstein, P. McCabe, T. Pitassi, and M. E. Saks, "Minimizing Disjunctive Normal Form Formulas and AC^0 Circuits Given a Truth Table," *SIAM J. Comput.*, vol. 38, no. 1, pp. 63–84, 2008.

[50] S. Hirahara, I. C. Oliveira, and R. Santhanam, "NP-hardness of Minimum Circuit Size Problem for OR-AND-MOD Circuits," in *Proceedings of the Computational Complexity Conference (CCC)*, 2018, pp. 5:1–5:31.

[51] R. Ilango, "Approaching MCSP from Above and Below: Hardness for a Conditional Variant and $\mathsf{AC}^0[p]$," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2020, pp. 34:1–34:26.

[52] ——, "Constant Depth Formula and Partial Function Versions of MCSP are Hard," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2020, pp. 424–433.

[53] R. Ilango, B. Loff, and I. C. Oliveira, "NP-Hardness of Circuit Minimization for Multi-Output Functions," in *Proceedings of the Computational Complexity Conference (CCC)*, 2020, pp. 22:1–22:36.

[54] R. Ilango, "The Minimum Formula Size Problem is (ETH) Hard," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2021, pp. 427–432.

[55] S. Hirahara, "NP-Hardness of Learning Programs and Partial MCSP," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2022, pp. 968–979.

[56] Y. Huang, R. Ilango, and H. Ren, "NP-Hardness of Approximating Meta-Complexity: A Cryptographic Approach," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2023, pp. 1067–1075.

[57] R. Ilango, "SAT Reduces to the Minimum Circuit Size Problem with a Random Oracle," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2023, pp. 733–742.

[58] N. Mazor and R. Pass, "Gap MCSP Is Not (Levin) NP-Complete in Obfustopia," in *Proceedings of the Computational Complexity Conference (CCC)*, 2024, pp. 36:1–36:21.

[59] C. D. Murray and R. R. Williams, "On the (Non) NP-Hardness of Computing Circuit Complexity," *Theory of Computing*, vol. 13, no. 1, pp. 1–22, 2017.

[60] M. Saks and R. Santhanam, "On Randomized Reductions to the Random Strings," in *Proceedings of the Computational Complexity Conference (CCC)*, 2022, pp. 29:1–29:30.

[61] S. Hirahara and O. Watanabe, "Limits of Minimum Circuit Size Problem as Oracle," in *Proceedings of the Conference on Computational Complexity (CCC)*, 2016, pp. 18:1–18:20.

[62] K. Ko, "On the Complexity of Learning Minimum Time-Bounded Turing Machines," *SIAM J. Comput.*, vol. 20, no. 5, pp. 962–986, 1991.

[63] H. Ren and R. Santhanam, "A Relativization Perspective on Meta-Complexity," in *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, 2022, pp. 54:1–54:13.

[64] J. M. Hitchcock and A. Pavan, "On the NP-Completeness of the Minimum Circuit Size Problem," in *Proceedings of the Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS)*, 2015, pp. 236–245.

[65] M. Saks and R. Santhanam, "Circuit Lower Bounds from NP-Hardness of MCSP Under Turing Reductions," in *Proceedings of the Computational Complexity Conference (CCC)*, 2020, pp. 26:1–26:13.

[66] E. Allender and S. Hirahara, "New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems," *TOCT*, vol. 11, no. 4, pp. 27:1–27:27, 2019.

[67] B. Barak, S. J. Ong, and S. P. Vadhan, "Derandomization in Cryptography," *SIAM J. Comput.*, vol. 37, no. 2, pp. 380–400, 2007.

[68] C. Yap, "Some Consequences of Non-Uniform Conditions on Uniform Classes," *Theor. Comput. Sci.*, vol. 26, pp. 287–300, 1983.

[69] L. Chen, S. Hirahara, and H. Ren, "Symmetric Exponential Time Requires Near-Maximum Circuit Size," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2024, pp. 1990–1999.

[70] Z. Li, "Symmetric Exponential Time Requires Near-Maximum Circuit Size: Simplified, Truly Uniform," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2024, pp. 2000–2007.

[71] J. Cai, "$\mathsf{S}_2^p \subseteq \mathsf{ZPP}^{\mathsf{NP}}$," *J. Comput. Syst. Sci.*, vol. 73, no. 1, pp. 25–35, 2007.

[72] J. Li and T. Yang, "3.1n - $o(n)$ circuit lower bounds for explicit functions," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2022, pp. 1180–1193.

[73] S. Rudich, "Super-bits, Demi-bits, and NP/qpoly-natural Proofs," in *Proceedings of the Randomization and Approximation Techniques in Computer Science (RANDOM/APPROX)*, 1997, pp. 85–93.

[74] R. Impagliazzo and M. Naor, "Efficient Cryptographic Schemes Provably as Secure as Subset Sum," *J. Cryptology*, vol. 9, no. 4, pp. 199–216, 1996.

[75] A. C. Yao, "Theory and application of trapdoor functions," in *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, 1982, pp. 80–91.

[76] O. Goldreich and Y. Oren, "Definitions and Properties of Zero-Knowledge Proof Systems," *J. Cryptol.*, vol. 7, no. 1, pp. 1–32, 1994.

[77] B. Kuykendall and M. Zhandry, "Towards Non-interactive Witness Hiding," in *Proceedings of the Theory of Cryptography Conference (TCC)*, 2020, pp. 627–656.

[78] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger, "Power from Random Strings," *SIAM J. Comput.*, vol. 35, no. 6, pp. 1467–1493, 2006.

[79] I. Dinur, V. Guruswami, S. Khot, and O. Regev, "A New Multilayered PCP and the Hardness of Hypergraph Vertex Cover," *SIAM J. Comput.*, vol. 34, no. 5, pp. 1129–1146, 2005.

[80] L. Fortnow and M. Sipser, "Are There Interactive Protocols for CO-NP

Languages?" *Inf. Process. Lett.*, vol. 28, no. 5, pp. 249–251, 1988.
[81] U. Feige and A. Shamir, "Witness Indistinguishable and Witness Hiding Protocols," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 1990, pp. 416–426.
[82] C. Dwork and M. Naor, "Zaps and Their Applications," *SIAM J. Comput.*, vol. 36, no. 6, pp. 1513–1543, 2007.
[83] N. Bitansky and O. Paneth, "ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation," in *Proceedings of the Theory of Cryptography Conference (TCC)*, 2015, pp. 401–427.
[84] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.
[85] J. Groth, R. Ostrovsky, and A. Sahai, "New Techniques for Noninteractive Zero-Knowledge," *J. ACM*, vol. 59, no. 3, pp. 11:1–11:35, 2012.
[86] A. Golovnev, R. Ilango, R. Impagliazzo, V. Kabanets, A. Kolokolova, and A. Tal, "$AC^0[p]$ Lower Bounds Against MCSP via the Coin Problem," in *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, 2019, pp. 66:1–66:15.
[87] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson, "Uniform Direct Product Theorems: Simplified, Optimized, and Derandomized," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1637–1665, 2010.
[88] D. Uhlig, "On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements," *Mathematical Notes of the Academy of Sciences of the USSR*, vol. 15, no. 6, pp. 558–562, 1974.
[89] ——, "Networks Computing Boolean Functions for Multiple Input Values," in *Poceedings of the London Mathematical Society Symposium on Boolean Function Complexity*. USA: Cambridge University Press, 1992, p. 165–173.
[90] I. Wegener, *The complexity of Boolean functions*. Wiley-Teubner, 1987.
[91] N. Nisan and A. Wigderson, "Hardness vs Randomness," *J. Comput. Syst. Sci.*, vol. 49, no. 2, pp. 149–167, 1994.