Characterizing Average-Case Complexity of PH by Worst-Case Meta-Complexity

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

***Abstract*—We exactly characterize the average-case complexity of the polynomial-time hierarchy (PH) by the worst-case (meta-)complexity of $\mathrm{GapMINKT}^{\mathsf{PH}}$, i.e., an approximation version of the problem of determining if a given string can be compressed to a short PH-oracle efficient program. Specifically, we establish the following equivalence:**

$$\mathsf{DistPH} \subseteq \mathsf{AvgP} \ (\textbf{i.e., PH is easy on average}) \iff \mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}.$$

In fact, our equivalence is significantly broad: A number of statements on several fundamental notions of complexity theory, such as errorless and one-sided-error average-case complexity, sublinear-time-bounded and polynomial-time-bounded Kolmogorov complexity, and PH-computable hitting set generators, are all shown to be equivalent.

Our equivalence provides fundamentally new proof techniques for analyzing average-case complexity through the lens of *meta-complexity* of time-bounded Kolmogorov complexity and resolves, as immediate corollaries, questions of equivalence among different notions of average-case complexity of PH: low success versus high success probabilities (i.e., a hardness amplification theorem for DistPH against uniform algorithms) and errorless versus one-sided-error average-case complexity of PH.

Our results are based on a sequence of new technical results that further develops the proof techniques of the author's previous work on the non-black-box worst-case to average-case reduction and unexpected hardness results for Kolmogorov complexity (FOCS'18, CCC'20, ITCS'20, STOC'20). Among other things, we prove the following.

1) $\mathrm{GapMINKT}^{\mathsf{NP}} \in \mathsf{P}$ **implies** $\mathsf{P} = \mathsf{BPP}$**. At the core of the proof is a new black-box hitting set generator construction whose reconstruction algorithm uses few random bits, which also improves the approximation quality of the non-black-box worst-case to average-case reduction without using a pseudorandom generator.**
2) $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$ **implies** $\mathsf{DistPH} \subseteq \mathsf{AvgBPP} = \mathsf{AvgP}$**.**
3) **If** $\mathrm{MINKT}^{\mathsf{PH}}$ **is easy on a** $1/\mathsf{poly}(n)$**-fraction of inputs, then** $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$**. This improves the error tolerance of the previous non-black-box worst-case to average-case reduction.**

The full version of this paper is available on ECCC.

***Keywords*-average-case complexity; meta-complexity; Kolmogorov complexity; polynomial-time hierarchy; hitting set generator; hardness amplification; pseudorandomness**

I. Introduction

Two of the mysteries of complexity theory are the *average-case complexity* of PH and the computational complexity of computing time-bounded Kolmogorov complexity (referred to as *meta-complexity*). These questions originate from the 1980s, when Levin [1] laid the foundation of the average-case complexity theory and Ko [2] investigated the complexity of MINKT, which is the problem of computing time-bounded Kolmogorov complexity. The exact relationship among them has not been well understood. In this paper, we establish an interdisciplinary link between the two subareas of complexity theory.

Main Theorem (a short version; "DistPH-completeness" of $\mathrm{GapMINKT}^{\mathsf{PH}}$)**.**

$$\mathsf{DistPH} \subseteq \mathsf{AvgP} \iff \mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}.$$

This equivalence connects two fundamentally different notions. On the left, the statement $\mathsf{DistPH} \subseteq \mathsf{AvgP}$ means that PH is easy on *most* instances. On the right, the statement $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$ means that an efficient algorithm can compute the PH-oracle Kolmogorov complexity of *every* instance. In fact, our equivalence is much larger, and it connects a number of statements on several notions of complexity theory, including errorless and one-sided-error average-case complexity, time-bounded Kolmogorov complexity, and PH-computable hitting set generator.

Our equivalence not only is interdisciplinary but also has significant impacts on fundamental questions in each subarea. We review the average-case complexity theory and its open questions in the next section, as well as the notion of time-bounded Kolmogorov complexity in the subsequent section.

II. Average-Case Complexity Theory

In practice, the traditional analysis of an algorithm based on *worst-case* inputs can be misleading. It is often reported that modern SAT solvers can solve huge instances, notwithstanding the NP-completeness of SAT. The main source of the disagreement between the practical performance of an algorithm and the NP-completeness theory is that the latter notion is based on the *worst-case* analysis of an algorithm; however, we cannot generate worst-case inputs, and thus never encounter them in reality.

Pioneered by Levin [1], the theory of *average-case complexity* aims at analyzing the performance of algorithms with respect to random inputs sampled efficiently from some distribution. Specifically, for a language $L\colon \{0,1\}^* \to \{0,1\}$

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00014

and a family of distributions $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$,[1] the pair $(L, \mathcal{D})$ is called a *distributional problem*. The task of the distributional problem $(L, \mathcal{D})$ is to compute $L(x)$ given a random input $x \sim \mathcal{D}_n$ for each instance size $n \in \mathbb{N}$.[2] Following Levin's original notion [1], $(L, \mathcal{D})$ is said to be *polynomial-time-solvable on average* if there exists an algorithm A computing L such that, for some constant $\epsilon > 0$, for all large $n \in \mathbb{N}$, $\mathbb{E}_{x\sim\mathcal{D}_n}[t_A(x)^\epsilon] \le O(n)$, where $t_A(x)$ denotes the running time of A on input x. The class of distributional problems that are polynomial-time-solvable on average is denoted by AvgP.

For our purpose, it is useful to use the equivalent notion of an *errorless heuristic scheme*. An errorless heuristic scheme A for a distributional problem $(L, \mathcal{D})$ satisfies the following: (1) A takes an input $x \in \mathsf{supp}(\mathcal{D}_n) \subseteq \{0,1\}^*$ and an error parameter $\delta \in (0, 1)$ and runs in time $\mathsf{poly}(n, 1/\delta)$, (2) A is errorless, that is, $A(x, \delta) \in \{L(x), \perp\}$, and (3) $A(x, \delta) = L(x)$ holds with probability at least $1 - \delta$ over the choice of $x \sim \mathcal{D}_n$ for any $n \in \mathbb{N}$. It is known that $(L, \mathcal{D}) \in \mathsf{AvgP}$ if and only if $(L, \mathcal{D})$ admits an errorless heuristic scheme. (Roughly speaking, an errorless heuristic scheme outputs $\perp$ if it takes super-polynomial time to compute.) For a function $\delta\colon \mathbb{N} \to (0, 1)$, let $\mathsf{Avg}_\delta\mathsf{P}$ denote the class of distributional problems $(L, \mathcal{D})$ that admit an errorless heuristic algorithm with error parameter fixed to $\delta(n)$. Further details on average-case complexity can be found in the survey of Bogdanov and Trevisan [3].

What kind of distributions should we consider? It is reasonable to focus on the distribution from which one can generate a random instance efficiently. A family of distributions $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ is said to be *polynomial-time samplable* if there exists a randomized polynomial-time algorithm that, on input 1^n, outputs a string that is distributed according to $\mathcal{D}_n$. The class of polynomial-time-samplable distributions is denoted by PSamp. For a complexity class $\mathfrak{C}$, let $\mathsf{Dist}\mathfrak{C}$ denotes the class $\mathfrak{C} \times \mathsf{PSamp}$ of distributional problems.

The fundamental questions of average-case complexity are whether $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ and its relationship with $\mathsf{DistNP} \subseteq \mathsf{Avg}_\delta\mathsf{P}$ holds for different choices of parameters δ. It is evident that $\mathsf{DistNP} \subseteq \mathsf{Avg}_\delta\mathsf{P}$ implies $\mathsf{DistNP} \subseteq \mathsf{Avg}_{\delta'}\mathsf{P}$ if $\delta \le \delta'$. However, it is a fundamental open question to prove the converse, depending on the choice of the error parameter δ. For example, in the extreme case of $\delta(n) = 2^{-n}$, an errorless heuristic algorithm is equivalent to a worst-case solver, whose relationship with an errorless heuristic scheme is open.

Open Question II.1 (Worst-case versus average-case complexity of NP)**.** Does $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ imply $\mathsf{DistNP} \subseteq \mathsf{Avg}_{2^{-n}}\mathsf{P}$ ($\Leftrightarrow \mathsf{NP} = \mathsf{P}$)?

This is one of the central questions in complexity theory, particularly because of its relationship to cryptography. In terms of Impagliazzo's five possible worlds, the open question corresponds to excluding Heuristica from the possible worlds (cf. [4]). In another regime of parameters, the question is referred to as *hardness amplification*.

Open Question II.2 (Hardness amplification for NP)**.**

$$\text{Does } \mathsf{DistNP} \subseteq \mathsf{Avg}_{1-1/\mathsf{poly}(n)}\mathsf{P} \text{ imply } \mathsf{DistNP} \subseteq \mathsf{AvgP}?$$

The history of hardness amplification dates back to Yao's XOR Lemma (cf. [5]). It is however relatively recently that Bogdanov and Safra [6] initiated the study of hardness amplification in the context of *errorless* average-case complexity, and made progress towards Open Question II.2 by showing that $\mathsf{DistNP} \subseteq \mathsf{Avg}_{1-(\log n)^{-1/10+o(1)}}\mathsf{P}$ implies $\mathsf{DistNP} \subseteq \mathsf{AvgP}$.[3]

Another natural question is whether two-sided-error average-case complexity of NP is equivalent to errorless average-case complexity, as raised in [4].

Open Question II.3 (Two-sided-error versus errorless average-case complexity of NP)**.**

$$\text{Does } \mathsf{DistNP} \subseteq \mathsf{HeurP} \text{ imply } \mathsf{DistNP} \subseteq \mathsf{AvgP}?$$

Here, HeurP denotes the class of distributional problems that admit a two-sided-error heuristic scheme. In fact, using the search-to-decision reduction for NP [8], a two-sided-error heuristic scheme for NP can be converted to a "one-sided-error" heuristic scheme for NP that always rejects NO instances and accepts most YES instances. In light of this, Open Question II.3 is morally equivalent to the following question.

Open Question II.4 (One-sided-error versus errorless average-case complexity of NP)**.**

$$\text{Does } \mathsf{DistNP} \subseteq \mathsf{Avg}^1\mathsf{P} \text{ imply } \mathsf{DistNP} \subseteq \mathsf{AvgP}?$$

Here, $\mathsf{Avg}^1\mathsf{P}$ denotes the class of distributional problems that admit a one-sided-error heuristic scheme. Specifically, we say that $(L, \mathcal{D}) \in \mathsf{Avg}^1\mathsf{P}$ if there exists an algorithm A such that (1) $A(x, \delta)$ runs in time $\mathsf{poly}(n, 1/\delta)$, (2) $L(x) = 0$ implies $A(x, \delta) = 0$, and (3) $A(x, \delta) = L(x)$ with probability at least $1 - \delta$ over the choice of $x \sim \mathcal{D}_n$.

So far we have explained the case of DistNP. However, all corresponding questions are open even for the Polynomial-time Hierarchy (PH). Recall that PH is a generalization of NP and is defined as $\bigcup_{k\in\mathbb{N}} \Sigma^{\mathrm{p}}_k$, where the k-th level Σ^{p}_k of PH is defined as $\mathsf{NP}^{\Sigma^{\mathrm{p}}_{k-1}}$ and $\Sigma^{\mathrm{p}}_0 := \mathsf{P}$. For example, the following is an easier question than Open Question II.1.

[1] We identify a language $L \subseteq \{0,1\}^*$ with its characteristic function $L\colon \{0,1\}^* \to \{0,1\}$.

[2] The support of $\mathcal{D}_n$ is a subset of $\{0,1\}^*$, and it is not required to be a subset of $\{0,1\}^n$.

[3] While their reduction is randomized, the reduction can be derandomized by using the pseudorandom generator of [7].

Open Question II.5 (Worst-case versus average-case complexity of PH)**.**

$$\text{Does } \mathsf{DistPH} \subseteq \mathsf{AvgP} \text{ imply } \mathsf{PH} = \mathsf{P} \ (\Leftrightarrow \mathsf{NP} = \mathsf{P})?$$

The landscape of average-case complexity is summarized in Fig. 1. The depicted implications are trivial facts, and the converse directions correspond to the open questions mentioned above.

As a part of our equivalence, we resolve the PH analogues of Open Questions II.2 and II.4 simultaneously.[4] Specifically, we show that, if there exists a *one-sided-error* algorithm for DistPH that succeeds with probability at least $1/\mathsf{poly}(n)$, then there exists an *errorless* heuristic scheme for DistPH.

Theorem II.6. *For any constant $c > 0$, if $\mathsf{DistPH} \subseteq \mathsf{Avg}^1_{1-n^{-c}}\mathsf{P}$, then $\mathsf{DistPH} \subseteq \mathsf{AvgP}$.*

Our proof techniques are fundamentally different from previous techniques. All previous proof techniques of hardness amplification that we are aware of use a hardness amplification procedure "$\mathrm{Amp}(f)$" (e.g., [5, 6, 9–16]). Specifically, a given function f is mapped to a candidate hard function $\mathrm{Amp}(f)$; typically, the function is defined as $\mathrm{Amp}(f)(x_1, \ldots, x_k) := f(x_1) \oplus \cdots \oplus f(x_k)$ as in Yao's XOR Lemma. Then, one designs an efficient oracle algorithm R that, given access to an oracle that solves $\mathrm{Amp}(f)$ on a small fraction of inputs, solves f on a large fraction of inputs.

The proof strategy based on $\mathrm{Amp}(f)$ is often referred to as *black-box*: The oracle algorithm R takes as oracle an *arbitrary* (even inefficient) heuristic algorithm for $\mathrm{Amp}(f)$ and solves f on average. Such a proof technique based on a black-box hardness amplification procedure is so general that there are a number of significant limits (e.g., [17–22]). For example, Viola [17] showed that the worst-case-to-average-case equivalence of PH (i.e., Open Question II.5) cannot be proved by using a black-box hardness amplification procedure.

Note that, in order to prove a hardness amplification theorem, it suffices to design a "*non-black-box*" oracle algorithm R that is successful only when the oracle is efficient. Here, a reduction is referred to as *non-black-box* if the proof of the correctness of the reduction uses the efficiency of the oracle in an essential manner.

The approach of this work is not black-box,[5] and our proof is based on new understanding of average-case complexity through the *meta-complexity* of time-bounded Kolmogorov complexity. In order to connect average-case complexity theory and meta-complexity theory, we employ the proof techniques of the author's previous work on a non-black-box worst-case-to-average-case reduction [23], which overcame another fundamental limit of black-box reductions that was presented by Feigenbaum and Fortnow [21] and Bogdanov and Trevisan [22]. We next review the notion of meta-complexity.

[4]By using the fact that PH is closed under complement, it is easy to prove that $\mathsf{DistPH} \subseteq \mathsf{Avg}^1\mathsf{P}$ iff $\mathsf{DistPH} \subseteq \mathsf{AvgP}$. However, the same argument does not work when the failure probability is large.

[5]We do not know whether our hardness amplification result (Theorem II.6) itself is subject to some black-box barrier results, such as [17, 18]. See also Section V-D.

III. Meta-Complexity of Time-Bounded Kolmogorov Complexity

Kolmogorov complexity enables quantifying how much a string is complex. Informally, the *t-time-bounded Kolmogorov complexity* of a finite string $x \in \{0,1\}^*$ is defined as the minimum size of a program that outputs x in t steps. For a formal definition, we need to clarify the meaning of "the size of a program" by fixing a particular interpreter. We fix an efficient universal Turing machine U. The t-time-bounded Kolmogorov complexity $\mathrm{K}^t(x)$ of x is formally defined as follows. (We adopt the definition that is meaningful even for $t \leq |x|$.)

Definition III.1. *For a string $x \in \{0,1\}^*$, an oracle $A \subseteq \{0,1\}^*$, and a time bound $t \in \mathbb{N} \cup \{\infty\}$,*

$$\mathrm{K}^{t,A}(x) := \min\{\, |d| \mid U^{d,A}(i) \text{ outputs } x_i \text{ in time } t \text{ for each } i \in [|x|+1] \,\}.$$

Here, $U^{d,A}(i)$ means the output of the universal Turing machine given random access to d and A and i as input; x_i denotes the ith bit of x if $i \leq |x|$ and $\perp$ otherwise. We omit the superscript A if $A = \varnothing$, and the superscript t if $t = \infty$, respectively.

Note that time-bounded Kolmogorov complexity itself asks the *complexity* of a shortest program to print a given string. Stepping back, one can ask the *complexity* of computing time-bounded Kolmogorov complexity—what is called *meta-complexity* of time-bounded Kolmogorov complexity. Although the origin of meta-complexity can be traced back to the Russian study of 1950s [24], its importance was identified only recently, especially through the study of the Minimum Circuit Size Problem (MCSP [25]).[6]

Among the early studies on meta-complexity of time-bounded Kolmogorov complexity, Ko [2] introduced and investigated MINKT, which is the problem of deciding, given $(x, 1^t, 1^s)$ as input, whether there is a program of size at most s that prints x in time t, as well as its approximation version which we denote by $\mathrm{GapMINKT}$. The problem $\mathrm{GapMINKT}$ asks for approximating $\mathrm{K}^t(x)$ within an additive error of $O(\log(|x| + t))$. Formally:

Definition III.2. *For a function $\tau\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ and an oracle $A \subseteq \{0,1\}^*$, $\mathrm{Gap}_\tau\mathrm{MINKT}^A$ is defined as the promise*

[6]The reader is referred to the survey of Allender [26] for more details on meta-complexity and MCSP.

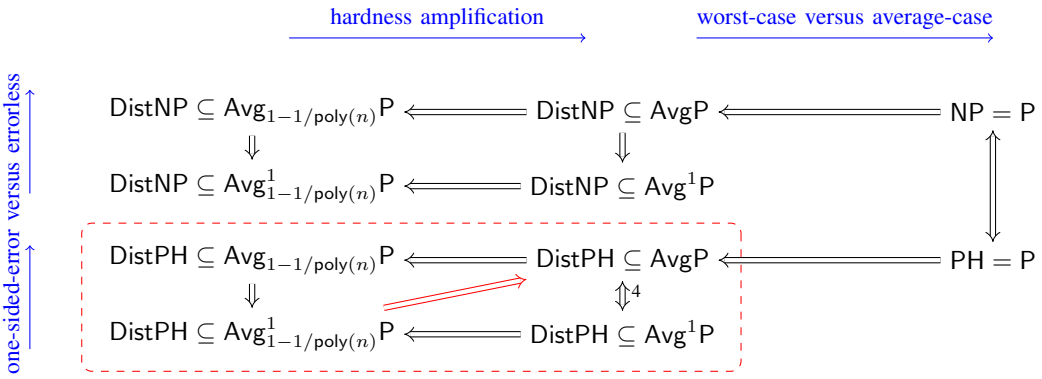

Figure 1. The relationships between average-case complexity of NP and PH under various notions. All implications depicted (except for our results) are trivial. The converse directions are fundamental open questions. A section of our equivalence is the four statements enclosed by the rectangle, which resolves the open questions regarding the average-case complexity of PH.

problem $(\Pi_{\text{YES}}, \Pi_{\text{NO}})$ *such that*

$$\Pi_{\text{YES}} := \{\, (x, 1^t, 1^s) \mid \mathrm{K}^{t,A}(x) \le s \,\},$$
$$\Pi_{\text{NO}} := \{\, (x, 1^t, 1^s) \mid \mathrm{K}^{\tau(|x|,t),A}(x) > s + \log \tau(|x|, t) \,\}.$$

We write $\mathrm{GapMINKT}^A \in \mathsf{P}$ *if there exists some polynomial* τ *such that* $\mathrm{Gap}_\tau\mathrm{MINKT}^A \in \mathsf{P}$. *We define* $\mathrm{MINKT}^A := (\Pi_{\text{YES}}, \{0,1\}^* \setminus \Pi_{\text{YES}})$.

We extend this definition to $\mathrm{GapMINKT}^{\mathfrak{C}}$ for a complexity class $\mathfrak{C}$. We could have simply defined $\mathrm{GapMINKT}^{\mathsf{PH}}$ as $\mathrm{GapMINKT}^A$ if we had a PH-complete problem A. However, this definition is problematic: If there exists a PH-complete problem A, then we must have $\mathsf{PH} \le^p_{\mathrm{m}} A \in \Sigma^{\mathrm{p}}_k$ for some constant $k \in \mathbb{N}$, which implies the unlikely consequence that the polynomial-time hierarchy collapses. We address this issue by introducing the definition of $\mathrm{GapMINKT}^{\mathfrak{C}}$ that does not depend on a complete problem, where $\mathfrak{C}$ is an arbitrary complexity class.

Definition III.3. *For a complexity class* $\mathfrak{C}$, *we regard* $\mathrm{GapMINKT}^{\mathfrak{C}}$ *as a family of problems* $\{\, \mathrm{GapMINKT}^A \mid A \in \mathfrak{C} \,\}$. *We write* $\mathrm{GapMINKT}^{\mathfrak{C}} \in \mathsf{P}$ *if* $\mathrm{GapMINKT}^A \in \mathsf{P}$ *for any oracle* $A \in \mathfrak{C}$. *Similarly, we define* $\mathrm{MINKT}^{\mathfrak{C}} := \{\, \mathrm{MINKT}^A \mid A \in \mathfrak{C} \,\}$.

One of the central questions on GapMINKT is to classify its complexity. It is easy to observe that $\mathrm{GapMINKT} \in \mathsf{NP}$, and more generally, $\mathrm{GapMINKT}^A \in \mathsf{NP}^A$ for any oracle A. Thus, the central question is to prove its NP-hardness, which would completely classify the complexity of GapMINKT as an "NP-complete" problem.[7]

Open Question III.4. Does $\mathrm{GapMINKT} \in \mathsf{P}$ imply $\mathsf{P} = \mathsf{NP}$?

[7]We often use the weak notion of hardness and completeness. For example, we say that a problem L is "NP-complete" if $L \in \mathsf{P}$ implies $\mathsf{P} = \mathsf{NP}$ and $L \in \mathsf{NP}$. This property is implied by the standard NP-completeness under polynomial-time reductions, but the converse is not necessarily true.

Somewhat surprisingly, the following easier question is open as well.

Open Question III.5 ("NP-hardness" of $\mathrm{GapMINKT}^{\mathsf{PH}}$)**.** Does $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$ imply $\mathsf{P} = \mathsf{NP}$?

Readers unfamiliar with meta-complexity may wonder why it is not trivial that NP is reducible to $\mathrm{GapMINKT}^{\mathsf{NP}}$, or more generally, that A is reducible to $\mathrm{GapMINKT}^A$ for any oracle A. Intuitively, $\mathrm{GapMINKT}^A$ seems to be more difficult than computing A. Unfortunately, there is a gap between this intuition and actually constructing a reduction from A to $\mathrm{GapMINKT}^A$: The meta-complexity of $\mathrm{GapMINKT}^A$ refers to the complexity of minimizing the size of an A-oracle program; it is not clear whether the task of computing A can be converted to the task of minimizing the size of an A-oracle program. In fact, this gap—which seems to be intuitively small—is the only missing piece for establishing the equivalence between worst-case and average-case complexity of PH.

Corollary III.6 (of Main Theorem)**.** *Open Question III.5 is* equivalent *to Open Question II.5. That is,* "NP-*hardness*" *of* $\mathrm{GapMINKT}^{\mathsf{PH}}$ *is equivalent to establishing the equivalence between worst-case and average-case complexity of* PH.

One of our main technical contributions is to partially close the gap between the tasks of computing A and minimizing an A-oracle program: We will show that $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$ implies $\mathsf{DistPH} \subseteq \mathsf{AvgP}$, which can be regarded as "DistPH-hardness" of $\mathrm{GapMINKT}^{\mathsf{PH}}$. In fact, our results classify the complexity of $\mathrm{GapMINKT}^{\mathsf{PH}}$ as a "DistPH-complete" problem in the following sense: $\mathsf{DistPH} \subseteq \mathsf{AvgP} \iff \mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$, thereby resolving the fundamental open question of the complexity of $\mathrm{GapMINKT}^{\mathsf{PH}}$.

A. Monotonicity

There is another counterintuitive property of meta-complexity that is not well understood—the *monotonicity* of meta-complexity. One might think that it is trivial that $\mathrm{GapMINKT}^{\mathsf{SAT}} \in \mathsf{P}$ implies $\mathrm{GapMINKT} \in \mathsf{P}$. However, this was not known before this work.[8]

More generally, one might guess that $\mathrm{GapMINKT}$ should be reducible to $\mathrm{GapMINKT}^A$ for any oracle A via, for instance, the identity reduction that maps an instance to itself. While the identity reduction can map any YES instance of $\mathrm{GapMINKT}$ to a YES instance of $\mathrm{GapMINKT}^A$, it does not necessarily map a NO instance of $\mathrm{GapMINKT}$ to a NO instance of $\mathrm{GapMINKT}^A$. The identity reduction actually works under the notion of the average-case complexity of $\mathrm{GapMINKT}$ [27], but, in terms of worst-case complexity, the reduction may not be correct. In fact, any deterministic reduction does not work: there exists some oracle A such that MCSP is not reducible to MCSP^A via any deterministic polynomial-time Turing reduction unless $\mathrm{MCSP} \in \mathsf{P}$ [28].

Nevertheless, we establish the following monotonicity property of meta-complexity.

Theorem III.7. *Let A and B be oracles such that A is NP-hard and $B \leq_{\mathrm{T}}^{p} A$. Then, $\mathrm{GapMINKT}^A \in \mathsf{P}$ implies $\mathrm{GapMINKT}^B \in \mathsf{P}$*

The proof of Theorem III.7 is based on non-black-box reductions, thereby bypassing the impossibility result of [28]. At the core of the proof is the randomized non-black-box worst-case-to-average-case reduction of [23, 29]. In order to derandomize it, we again use a (deterministic) non-black-box worst-case-to-average-case reduction to construct a nearly optimal pseudorandom generator, by making use of the NP-hardness of the oracle A.

IV. Our Results

Thus far, we have explained the significance of our results mainly within each subarea of complexity theory. We now guide the reader to our interdisciplinary equivalence that connects average-case complexity, meta-complexity of time-bounded Kolmogorov complexity, and more. Since the number of equivalent statements is large, we will explain one by one while presenting some ideas of the proofs.

Theorem IV.1 (Main results)**.** *The following (Items 1 to 12) are equivalent.*

1) $\mathsf{DistPH} \subseteq \mathsf{AvgP}$.

2) $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$.

Our equivalence is considerably robust with respect to minor changes. For example, recall that $\mathrm{GapMINKT}^{\mathsf{PH}}$ is the family of problems $\mathrm{GapMINKT}^A$ for each oracle $A \in \mathsf{PH}$. Although this is a convenient notion, we do not have to consider every oracle $A \in \mathsf{PH}$; alternatively, it suffices to consider a complete problem for each level of PH. Let $\Sigma_k\mathsf{SAT}$ denote the canonical complete problem for Σ_k^{p}. Then the following is equivalent as well.

3) $\mathrm{GapMINKT}^{\Sigma_k\mathsf{SAT}} \in \mathsf{P}$ *for any constant* $k \in \mathbb{N}$.

The significance of our equivalence is that the robustness of meta-complexity can be transferred to average-case complexity, the latter of which is often not resilient to modifications of success probability or the notion of errorless to one-sided-error. The equivalence enables us to reduce the success probability of an errorless heuristic algorithm to $1/\mathsf{poly}(n)$, which establishes a hardness amplification theorem for PH against uniform algorithms.

4) $\mathsf{DistPH} \subseteq \mathsf{Avg}_{1-n^{-c}}\mathsf{P}$ *for some constant* $c > 0$.

Furthermore, the equivalence extends to one-sided-error heuristic algorithms, thereby equating the errorless and one-sided-error average-case complexity of PH. We are unaware of any existing proof techniques that can yield such an equivalence.[9]

5) $\mathsf{DistPH} \subseteq \mathsf{Avg}^1_{1-n^{-c}}\mathsf{P}$ *for some constant* $c > 0$.

How do we establish the equivalence? Essential to our proof is to identify $\mathrm{MINKT}^{\mathsf{PH}} \times \mathsf{PSamp}$ as a natural "DistPH-complete" family of distributional problems.[10] Here, we say that a class $\mathfrak{C}$ of distributional problems is "DistPH-complete" if $\mathfrak{C} \subseteq \mathsf{DistPH}$, and $\mathfrak{C} \subseteq \mathsf{AvgP}$ implies $\mathsf{DistPH} \subseteq \mathsf{AvgP}$. We show that $\mathrm{MINKT}^{\mathsf{PH}} \times \mathsf{PSamp}$ is "DistPH-hard" even if the success probability of an errorless heuristic algorithm is $1/\mathsf{poly}(n)$.

6) $\mathsf{Dist}(\mathrm{MINKT}^{\mathsf{PH}}) := \mathrm{MINKT}^{\mathsf{PH}} \times \mathsf{PSamp} \subseteq \mathsf{Avg}_{1-n^{-c}}\mathsf{P}$ *for some constant* $c > 0$.

The reason why we are able to show the equivalence between one-sided-error and errorless average-case complexity is that $\mathsf{Dist}(\mathrm{MINKT}^{\mathsf{PH}})$ is "DistPH-hard" in an even stronger sense. If $\mathrm{coMINKT}^{\mathsf{PH}}$ admits a *one-sided-error* heuristic algorithm, then DistPH admits an *errorless* heuristic algorithm. Here, $\mathsf{co}\mathfrak{C}$ denotes the complement of $\mathfrak{C}$ for a class $\mathfrak{C}$.

7) $\mathrm{coMINKT}^{\mathsf{PH}} \times \mathsf{PSamp} \subseteq \mathsf{Avg}^1_{1-n^{-c}}\mathsf{P}$ *for some constant* $c > 0$.

Next, we explain how to resolve the issue of monotonicity. As mentioned before, the meta-complexity of $\mathrm{GapMINKT}^A$ is not necessarily monotone increasing with respect to A. In our previous work [29], we introduced the notion of "non-disjoint" promise problems so that the

[8] In contrast, it is easy to observe that $\mathrm{GapMINKT}^{\mathsf{NP}} \in \mathsf{P}$ implies $\mathrm{GapMINKT} \in \mathsf{P}$.

[9] The standard techniques of error-correcting codes yield such an equivalence for high complexity classes such as PSPACE and EXP. For example, two-sided-error average-case and worst-case complexity of PSPACE are equivalent [30, 31], and so is the one-sided-error average-case complexity. However, the proof technique is not applicable to PH [17]. We also mention that there is a simple argument that works if the failure probability is small.

[10] We mention that it is easy to construct an *artificial* DistPH-complete family of distributional problems [32].

monotonicity can be incorporated into the definition of a problem itself.

Specifically, let $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K})$ denote (a family of) the promise problems whose YES instances are those of $\mathrm{GapMINKT}^{\mathsf{PH}}$ and NO instances are those of $\mathrm{GapMINKT}$. This is not a standard promise problem in the sense that, under the plausible assumption that $\mathsf{E}^{\mathsf{NP}} \neq \mathsf{E}$, there exists an instance that is simultaneously a YES and NO instance, and thus $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K})$ is not a disjoint pair of languages; in this case, any algorithm—not only a polynomial-time algorithm but also literally *any* algorithm—cannot solve $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K})$. Nevertheless, under the assumption that $\mathsf{DistPH} \subseteq \mathsf{AvgP}$, there exists a polynomial-time algorithm that solves the "non-disjoint" promise problem.

8) $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K}) \in \mathsf{P}$.

The mathematical properties of the meta-complexity of $\mathrm{Gap}(\mathrm{K}^{A}$ vs $\mathrm{K})$ are better and more intuitive than $\mathrm{GapMINKT}^{A}$. For example, it is not hard to see that there is a many-one reduction from $\mathrm{Gap}(\mathrm{K}^{A}$ vs $\mathrm{K})$ to $\mathrm{Gap}(\mathrm{K}^{B}$ vs $\mathrm{K})$ for any oracles $A \leq^{p}_{\mathrm{T}} B$, which serves as a key property for proving the monotonicity of meta-complexity (Theorem III.7). Moreover, the identity map reduces $\mathrm{GapMINKT}^{\mathsf{PH}}$ to $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K})$, and thus the latter problem is a harder problem, which explains the implication from Item 8 to 2.

The question is—how can we show that there exists a *polynomial-time algorithm* that can solve the "non-disjoint" promise problem which we believe *no algorithm* can solve? A short answer is that $\mathrm{MINKT}^{\mathsf{PH}}$ is inherently a *meta-computational* problem that encodes a computation as its instance.[11] This enables us to show that, under the assumption that $\mathrm{GapMINKT}^{\mathsf{PH}} \in \mathsf{P}$, for any oracle $A \in \mathsf{PH}$, there exists a polynomial τ_A such that $\mathrm{K}^{\tau_A(|x|,t)}(x) \leq \mathrm{K}^{t,A}(x) + \log \tau_A(|x|,t)$ for any $x \in \{0,1\}^*$ and any $t \in \mathbb{N}$, in which case $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K})$ is indeed a *disjoint* promise problem.

One of the key components of the proof is to bridge the gap from the one-sided-error average-case complexity of $\mathsf{coMINKT}^{\mathsf{PH}} \times \mathsf{PSamp}$ to the worst-case meta-complexity of $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}}$ vs $\mathrm{K})$, that is, the implication from Item 7 to Item 8. The gap can be closed by using the proof techniques of the *non-black-box* worst-case-to-average-case reductions of [23, 29]. Unfortunately, the previous worst-case-to-average-case reductions are not error-tolerant, and require the success probability of a one-sided-error heuristic algorithm to be at least $1 - 1/\mathsf{poly}(n)$. We need to reduce the requirement of the success probability to $1/\mathsf{poly}(n)$.

One of the technical contributions of this work is to make the reductions error-tolerant. The main bottleneck of the previous reductions is the existence of the time parameter t: In the previous reductions, an instance $(x, 1^t, 1^s)$ was reduced to some instance $(x', 1^{t'}, 1^{s'})$, where $t' = \mathsf{poly}(n, t)$ and $n = |x|$. Because we require solving $\mathrm{GapMINKT}^{\mathsf{PH}}$ for *every* time parameter $t \in \mathbb{N}$, it was also required that a heuristic algorithm solves $(x', 1^{t'}, 1^{s'})$ for *every* time parameter t', which can be ensured if the success probability of the heuristic algorithm is assumed to be at least $1 - 1/\mathsf{poly}(n)$.

A new insight of this work is that the time bound can be fixed to $t := n^{\gamma}$, where $\gamma > 0$ is an arbitrary constant and n is the length of x. For a function $t\colon \mathbb{N} \to \mathbb{N}$, let $\mathrm{GapMINKT}^{\mathsf{PH}}[t = t(n)]$ denote the version of $\mathrm{GapMINKT}^{\mathsf{PH}}$, where the time bound is fixed to $t(|x|)$ on input $(x, 1^s)$. The following is equivalent, for any constant $\gamma > 0$.

9) $\mathrm{GapMINKT}^{\mathsf{PH}}[t = n^{\gamma}] \in \mathsf{P}$.

For example, one can regard $\mathrm{GapMINKT}^{\mathsf{PH}}[t = n^{1/10}]$ or $\mathrm{GapMINKT}^{\mathsf{PH}}[t = n^{100}]$ as a problem that characterizes the average-case complexity of PH. We mention in passing that $\mathrm{GapMINKT}^{\mathsf{PH}}[t = n^{\gamma}]$ for $\gamma \in (0,1)$ can be regarded as a *sublinear*-time-bounded Kolmogorov complexity, which is reminiscent of MKTP [33, 34]. Here, MKTP is the problem of, given an input x, computing the trade-off $\mathrm{KT}(x) := \min\{\, \mathrm{K}^t(x) + t \mid t \in \mathbb{N} \,\}$ between a description length and a (sublinear-)time bound.

We have explained the equivalence that connects average-case complexity and meta-complexity of time-bounded Kolmogorov complexity. Our equivalence even extends to the non-existence of a hitting set generator, which is one of the fundamental notions of complexity theory. Recall that a family of functions $H = \{H_n : \{0,1\}^{s(n)} \to \{0,1\}^n\}_{n \in \mathbb{N}}$ is said to be a *hitting set generator* (HSG) secure against a complexity class $\mathfrak{C}$ if, for every $C \in \mathfrak{C}$, for infinitely many $n \in \mathbb{N}$, $\Pr_{w \sim \{0,1\}^n}[C(w) = 1] \geq 1/4$ implies that $C(H_n(z)) = 1$ for some $z \in \{0,1\}^{s(n)}$.

As observed in [35], it is not hard to see that the existence of a PH-computable hitting set generator implies $\mathsf{DistPH} \not\subseteq \mathsf{AvgP}$. Surprisingly, we establish the converse direction.

10) There exists no hitting set generator $H = \{H_n : \{0,1\}^{n-1} \to \{0,1\}^n\}_{n \in \mathbb{N}}$ *computable in polynomial time with* PH *oracle that is secure against* P*, and* $\mathsf{P} = \mathsf{ZPP}$.[12]

We note that the notion of hitting set generator considered here is *cryptographic* as opposed to *complexity-theoretic*. The latter notion is suitable for derandomizing one-sided-error randomized algorithms, and allows the computational resource for computing a hitting set generator to be larger than its adversary. In contrast, we require that a hitting set generator H is computable in a *fixed polynomial time* (with PH oracle) and H is secure against an *arbitrary polynomial-time* adversary.

[11] More specifically, the instances encode some relationships among complexity classes.

[12] For some technical reason, we include in Item 10 the mild derandomization assumption that $\mathsf{P} = \mathsf{ZPP}$. In particular, Item 10 is equivalent to Items 1 to 9 under the assumption that $\mathsf{P} = \mathsf{ZPP}$.

One of the central questions about cryptographic hitting set generators is whether one can extend its seed. It is well known that a cryptographic pseudorandom generator $G = \{G_n : \{0,1\}^{n-1} \to \{0,1\}^n\}_{n\in\mathbb{N}}$ that extends its seed by one bit can be stretched to $\mathsf{poly}(n)$ bits. However, the corresponding question on a hitting set generator is open, as raised by Rudich [36]. We make the first progress towards resolving the question, by showing that the seed of a PH-computable HSG can be extended by 1 bit (¬Item 10) if and only if it can be extended by $O(\log n)$ bits (¬Item 11).

11) *For some constant $c > 0$, there exists no* PH*-computable hitting set generator $H = \{H_n : \{0,1\}^{n-c\log n} \to \{0,1\}^n\}_{n\in\mathbb{N}}$ that is secure against* P*, and* $\mathsf{P} = \mathsf{ZPP}$.

Another natural question regarding a polynomial-time-computable HSG is whether it is equivalent to a sublinear-time-computable HSG. Specifically, for any constant $\gamma > 0$, we say that an HSG $H = \{H_n : \{0,1\}^{s(n)} \to \{0,1\}^n\}_{n\in\mathbb{N}}$ is computable in time n^γ if, given random access to $z \in \{0,1\}^{s(n)}$ and an index $i \in [n]$, one can compute the ith bit of $H_n(z)$ in time n^γ. We show that the following is equivalent for any constant $\gamma > 0$.

12) *For some constant $c > 0$, for any constant $k \in \mathbb{N}$, there exists no hitting set generator $H = \{H_n : \{0,1\}^{n-c\log n} \to \{0,1\}^n\}_{n\in\mathbb{N}}$ computable in time n^γ with $\Sigma_k\mathsf{SAT}$ oracle that is secure against* P*, and* $\mathsf{P} = \mathsf{ZPP}$.

For example, the non-existence of a PH-oracle $n^{1/10}$-computable HSG (for $\gamma := 1/10$) and that of a PH-oracle n^{100}-computable HSG (for $\gamma := 100$) are equivalent. This is a rather counterintuitive result: Naively, one can imagine that, if the time bound $n^{1/10}$ is increased to n^{100}, more strings can be computed, and thus a hitting set generator should become more secure. As a consequence, one might guess that Item 12 should not be equivalent to the non-existence of PH-computable hitting set generators. This intuition turns out to be not correct.

Instead, it is instructive to consider Item 12 as an HSG analogue of the pseudorandom function generator construction of Goldreich, Goldwasser, and Micali [37], from which it follows that any $\mathsf{poly}(n)$-time computable PRG can be converted to an n^γ-time computable PRG, where n denotes the output length of PRGs. ◊

Fig. 2 summarizes some of the important statements and our proof strategies. On the top half of the figure are the statements on average-case complexity. On the bottom half of the figure are the statements on worst-case meta-complexity. The essential steps in our proof are to connect these fundamentally different statements.

One crucial step that brings us from the average-case-complexity world to the worst-case meta-complexity world is the following, which provides the non-black-box error-tolerant worst-case-to-average-case reduction and improves [23, 29].

Theorem IV.2 (Item 7 ⇒ 8)**.** *Let $c > 0$ be any constant and A be any* NP*-hard oracle.*[13] *If $\{\mathrm{coMINKT}^A\} \times \mathsf{PSamp} \subseteq \mathsf{Avg}^1_{1-n^{-c}}\mathsf{P}$, then $\mathrm{Gap}(\mathrm{K}^A \text{ vs } \mathrm{K}) \in \mathsf{P}$.*

Due to the barrier of Bogdanov and Trevisan [22], this step cannot be regarded as a (black-box) worst-case-to-average-case reduction (see [23, 35] for detailed discussion); thereby it crosses the boundary from the average-case world to the worst-case meta-complexity world.

Another crucial step that brings us from the worst-case meta-complexity world to the average-complexity world is the following, which establishes "DistPH-hardness" of $\mathrm{GapMINKT}^{\mathsf{PH}}$ by building on the ideas developed in [38, 39].

Theorem IV.3 (Item 3 ⇒ 1)**.** *Let A be any Σ_k^{p}-hard problem for some $k \in \mathbb{N}$. If $\mathrm{GapMINKT}^A \in \mathsf{P}$, then $\mathsf{Dist}\Sigma_k^{\mathrm{p}} \subseteq \mathsf{AvgP}$.*

Since our hardness amplification theorem for PH (Theorem II.6) is a statement purely on average-case complexity, it is natural to ask whether we can simplify our proof to provide a purely average-case complexity-theoretic argument. Note that Theorem IV.3 provides an "average-case-to-worst-case" reduction. If we could interpret this reduction as an average-case-to-average-case reduction, then we would have obtained an average-case complexity-theoretic proof easily. Surprisingly, the reduction of Theorem IV.3 *cannot* be regarded as an average-case-to-average-case reduction for any $k \geq 2$, and it is essential to cross the boundary from the worst-case meta-complexity world to the average-case world. It is this interplay between average-case complexity and worst-case meta-complexity that enables resolving the fundamental open questions. The details are explained in the full version of the paper under the name of the "$\mathsf{S}_2^{\mathrm{p}}$-barrier".

Both of Theorems IV.2 and IV.3 make use of the existence of a (complexity-theoretic) pseudorandom generator. In fact, one of our main technical contributions is to prove $\mathsf{P} = \mathsf{BPP}$ under the assumptions that any one of Items 1 to 12 holds. For example:

Theorem IV.4 (BPP-hardness)**.** *Let A be any* NP*-hard oracle. If $\mathrm{GapMINKT}^A \in \mathsf{P}$, then $\mathsf{P} = \mathsf{BPP}$.*

Fig. 3 summarizes the relationship among statements on different levels of PH.

A. Meta-Complexity is Indispensable for Average-Case Complexity

Our results have important consequences to the open questions mentioned before.

[13] The NP-hardness of A is used to construct an explicit pseudorandom generator of logarithmic seed length secure against linear-sized circuits. In particular, under the plausible assumption that $\mathsf{E} \not\subseteq \bigcap_{\epsilon>0} \mathsf{i.o.SIZE}(2^{\epsilon n})$ [11], Theorem IV.2 holds for any oracle A.

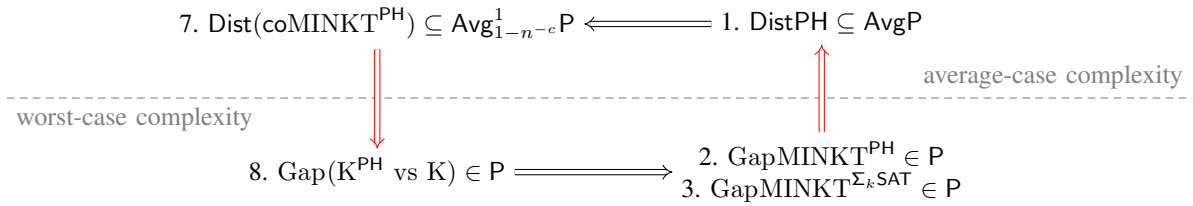

Figure 2. Some of the statements in the equivalence of the main result. The main technical implications are highlighted in red.

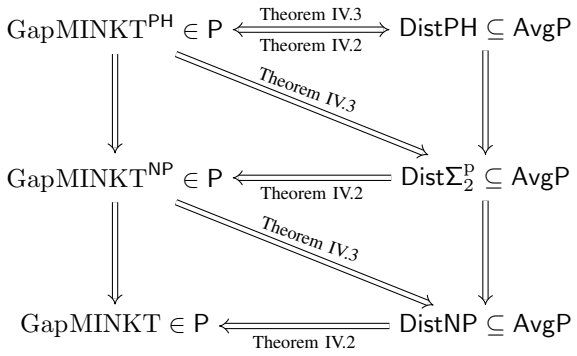

Figure 3. The relationships between GapMINKT$^{\mathfrak{C}}$ for $\mathfrak{C} \in \{\mathsf{P}, \mathsf{NP}, \mathsf{PH}\}$ and average-case complexity.

As mentioned in Open Question III.4, the fundamental open question of GapMINKT is to prove its "NP-completeness". In our previous work [23, 29], we showed that Open Question III.4 is sufficient for equating the worst-case and average-case complexity of NP (i.e., Open Question III.4 implies Open Question II.1). The results of [23] overcame the significant barrier of Bogdanov and Trevisan [22]: No non-adaptive black-box reduction can reduce NP-complete problems to DistNP (unless PH collapses) nor reduce GapMINKT to DistNP (unless GapMINKT $\in$ coNP/poly); in contrast, the proof techniques of [23] are non-black-box. However, it is possible that *adaptive* reductions can also bypass the barrier of [22], and it was not clear at all whether proving NP-hardness of meta-computational problems is really necessarily for resolving Open Question II.1.

The results of this work indicate that we cannot resolve Open Question II.1 without resolving "NP-hardness" of GapMINKT$^{\mathsf{PH}}$ (Open Question III.5); therefore, *meta-complexity is indispensable for studying average-case complexity*. To summarize, using informal notations such as NP $\leq$ DistPH (meaning that DistPH $\subseteq$ AvgP $\implies$ NP = P), we have the following relationships among open questions.

Corollary IV.5. *Open Question III.4* (NP $\leq$ GapMINKT) $\implies$ *Open Question II.1* (NP $\leq$ DistNP) $\implies$ *Open Question II.5* (NP $\leq$ DistPH) $\iff$ *Open Question III.5* (NP $\leq$ GapMINKT$^{\mathsf{PH}}$).

B. A New Approach Towards Hardness Amplification for NP

There are oracles relative to which Open Questions II.1 and III.4 do not hold, as constructed by Ko [2] and Impagliazzo [40], respectively. Therefore, we need to develop a non-relativizing proof technique in order to resolve these open questions. In light of this, we propose an open question that is less challenging but still has an important consequence:

Open Question IV.6 ("DistNP-hardness" of GapMINKT)**.** Does GapMINKT $\in$ P implies DistNP $\subseteq$ AvgP?

Open Question IV.6 provides a completely new approach towards improving the hardness amplification result of Bogdanov and Safra [6] from $1/(\log n)^{1/10}$ to $1/\mathsf{poly}(n)$. Specifically:

Corollary IV.7 (of Theorem IV.2)**.** *Open Question IV.6 implies Open Question II.2 (i.e., the hardness amplification theorem for* NP*).*

Note that Theorem IV.3 shows "DistNP-hardness" of GapMINKT$^{\mathsf{NP}}$; Open Question IV.6 asks whether the NP-oracle can be eliminated. It should be also noted that Open Question IV.6 would classify the complexity of GapMINKT as a "DistNP-complete" problem in light of Theorem IV.2.

C. Practically Generating Hard NP Instances?

We note that the error-tolerant worst-case-to-average-case reduction could be used to practically generate hard NP instances. Specifically, under the plausible assumptions that there exist (cryptographic and complexity-theoretic) pseudo-random generators, a $(1 - 1/\mathsf{poly}(n))$-fraction of instances of MINKT require super-polynomial time to solve.

Corollary IV.8 (of Theorem IV.2)**.** *Assume that* $\mathsf{E} \not\subseteq \bigcap_{\epsilon>0} \mathsf{i.o.SIZE}(2^{\epsilon n})$ *and* GapMINKT $\notin$ P*. Let* $c > 0$ *be any constant. Then, for any algorithm* M *that solves* MINKT$[t = n^{1/10}, s = n - (c+1)\log n]$ *and any polynomial* p*, for infinitely many* $n \in \mathbb{N}$*, with probability at least* $1 - n^{-c}$ *over the choice of* $x \sim \{0,1\}^n$*,* $t_M(x) \geq p(n)$ *holds, where* $t_M(x)$ *denotes the running time of* M *on input* x*.*

The assumption that $\mathrm{GapMINKT} \notin \mathsf{P}$ is relatively mild. For example, any one of the following assumptions implies $\mathrm{GapMINKT} \notin \mathsf{P}$: $\mathsf{SZK} \not\subseteq \mathsf{BPP}$ [41], the existence of cryptographic pseudorandom generators, the existence of (auxiliary-input) one-way functions [42], unsolvability of Random 3SAT in polynomial time [27], or more generally, the existence of a cryptographic hitting set generator [23].

V. Related Work

We present several previous works that are closely related to this work as well as impacts of our results to theirs.

A. From Average-Case Complexity to Meta-Complexity

In the area of meta-complexity, the proof techniques of average-case complexity have been often exploited. Using random self-reducibility and downward self-reducibility of some PSPACE-complete problem [31], Allender, Buhrman, Koucký, van Melkebeek, Ronneburger [33] showed that $\mathsf{PSPACE} \subseteq \mathsf{ZPP}^{\mathrm{MCSP}^{\mathsf{PSPACE}}}$, and it is immediate that the same proof technique shows that $\mathsf{PSPACE} \subseteq \mathsf{ZPP}^{\mathrm{GapMINKT}^{\mathsf{PSPACE}}}$. Combining their results with the BPP-hardness (Theorem IV.4), we immediately obtain "PSPACE-completeness" of $\mathrm{GapMINKT}^{\mathsf{PSPACE}}$ under deterministic reductions.

Corollary V.1. $\mathrm{GapMINKT}^{\mathsf{PSPACE}} \in \mathsf{P}$ *if and only if* $\mathsf{PSPACE} = \mathsf{P}$.

Proof Sketch: If $\mathrm{GapMINKT}^{\mathsf{PSPACE}} \in \mathsf{P}$, [33] implies $\mathsf{PSPACE} = \mathsf{ZPP}$. By Theorem IV.4, we also have $\mathsf{ZPP} \subseteq \mathsf{BPP} = \mathsf{P}$. ■

Impagliazzo, Kabanets, and Volkovich [43] generalized the result of [33] to $\mathfrak{C} \subseteq \mathsf{BPP}^{\mathrm{GapMINKT}^{\mathfrak{C}}}$ for any $\mathfrak{C} \in \{\oplus\mathsf{P}, \mathsf{P}^{\#\mathsf{P}}, \mathsf{PP}\}$. As in Corollary V.1, Theorem IV.4 enables improving their hardness results under randomized reductions to "deterministic reductions."

Corollary V.2. *Let* $\mathfrak{C} \in \{\mathsf{BPP}^{\oplus\mathsf{P}}, \mathsf{P}^{\#\mathsf{P}}, \mathsf{PP}\}$. *If* $\mathrm{GapMINKT}^{\mathfrak{C}} \in \mathsf{P}$, *then* $\mathfrak{C} = \mathsf{P}$.

Proof Sketch: Since $\mathfrak{C}$ includes NP (where $\mathsf{NP} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$ is due to [44]),[14] we can apply Theorem IV.4 and obtain $\mathsf{P} = \mathsf{BPP}$; thus, [43] implies $\mathfrak{C} \subseteq \mathsf{BPP} = \mathsf{P}$. ■

B. From Kolmogorov Complexity to Average-Case Complexity

Previously, Kolmogorov complexity *(instead of its meta-complexity)* was considered as a fundamental tool for analyzing average-case complexity. For example, Li and Vitányi [45] used Kolmogorov complexity to define a (not computable) distribution under which the average-case and worst-case complexity are equivalent.

[14]Since it is not known whether $\mathsf{NP} \subseteq \oplus\mathsf{P}$, we do not know whether Corollary V.2 holds for $\mathfrak{C} = \oplus\mathsf{P}$.

Antunes and Fortnow [46] characterized the running time of average-case algorithms by using the notion of *computational depth*. The computational depth (with time bound t) of a string x is defined as $\mathrm{cd}^t(x) := \mathrm{K}^t(x) - \mathrm{K}(x)$, whose notion was introduced by Antunes, Fortnow, van Melkebeek, and Vinodchandran [47]. Under the assumption that exponential time is not infinitely often in sub-exponential space, it was shown in [46] that, for all polynomial p, the running time of A is bounded by $2^{\mathrm{cd}^{p(|x|)}(x)+O(\log|x|)}$ for any input x if and only if A runs in average-case polynomial time with respect to any distribution $\mu \in \mathsf{PSamp}$.

We emphasize the fundamental differences between these previous results and this work. Our work characterizes average-case complexity via the *meta-complexity* of time-bounded Kolmogorov complexity, whereas the previous results characterize average-case complexity via Kolmogorov complexity itself. It should be also noted that the result of [46] is a conditional result, whereas our equivalence is unconditional, for which we make significant technical contributions.

C. Cryptographic Hitting Set Generator

Santhanam [48] posed the Universality Conjecture, under which a "succinct" hitting set generator can be extended arbitrary. The conjecture cannot be refuted unless one-way functions fail to exist, and, at the same time, its solvability remains unclear. It is left as an interesting research direction to make progress towards the Universality Conjecture using the proof techniques behind Theorem IV.1, which shows that a PH-computable hitting set generator can be slightly extended.

D. Hardness Amplification

We mention some previous works on hardness amplification. Impagliazzo, Jaiswal, Kabanets, and Wigderson [15, 16] showed a uniform version of Yao's XOR lemma; in particular, it was shown that, if $\mathsf{P}_{\|}^{\mathsf{NP}} \times \{\mathcal{U}\} \subseteq \mathsf{Heur}_{1/2+1/\mathsf{poly}(n)}\mathsf{BPP}$, then $\mathsf{P}_{\|}^{\mathsf{NP}} \times \{\mathcal{U}\} \subseteq \mathsf{Heur}_{1/\mathsf{poly}(n)}\mathsf{BPP}$. Bogdanov and Safra [6] proved a non-uniform and errorless version of Yao's XOR lemma, and showed that if $\mathfrak{C} \times \{\mathcal{U}\} \subseteq \mathsf{Avg}_{1-1/\mathsf{poly}(n)}\mathsf{P}/\mathsf{poly}$ then $\mathfrak{C} \times \{\mathcal{U}\} \subseteq \mathsf{AvgP}/\mathsf{poly}$ for any class $\mathfrak{C}$ closed under taking XORs.

Note that, if there were a *uniform and errorless* version of Yao's XOR lemma, it would have provided an alternative proof of the PH analogue of Open Question II.2 (i.e., the hardness amplification theorem for PH against uniform algorithms). However, as noted in [49], the proof techniques of [15, 16] "do not seem to apply to the errorless setting."

Acknowledgment

This work is supported by ACT-I, JST. We thank anonymous reviewers for helpful comments.

References

[1] L. A. Levin, "Average Case Complete Problems," *SIAM J. Comput.*, vol. 15, no. 1, pp. 285–286, 1986.

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

[3] A. Bogdanov and L. Trevisan, "Average-Case Complexity," *Foundations and Trends in Theoretical Computer Science*, vol. 2, no. 1, 2006.

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

[5] O. Goldreich, N. Nisan, and A. Wigderson, "On Yao's XOR-Lemma," in *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, 2011, pp. 273–301.

[6] A. Bogdanov and M. Safra, "Hardness Amplification for Errorless Heuristics," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2007, pp. 418–426.

[7] H. Buhrman, L. Fortnow, and A. Pavan, "Some Results on Derandomization," *Theory Comput. Syst.*, vol. 38, no. 2, pp. 211–227, 2005.

[8] S. Ben-David, B. Chor, O. Goldreich, and M. Luby, "On the Theory of Average Case Complexity," *J. Comput. Syst. Sci.*, vol. 44, no. 2, pp. 193–219, 1992.

[9] L. A. Levin, "One-way functions and pseudorandom generators," *Combinatorica*, vol. 7, no. 4, pp. 357–363, 1987.

[10] R. Impagliazzo, "Hard-Core Distributions for Somewhat Hard Problems," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 1995, pp. 538–545.

[11] R. Impagliazzo and A. Wigderson, "$P = BPP$ if E Requires Exponential Circuits: Derandomizing the XOR Lemma," in *Proceedings of the Symposium on the Theory of Computing (STOC)*, 1997, pp. 220–229.

[12] R. O'Donnell, "Hardness amplification within NP," *J. Comput. Syst. Sci.*, vol. 69, no. 1, pp. 68–94, 2004.

[13] A. Healy, S. P. Vadhan, and E. Viola, "Using Nondeterminism to Amplify Hardness," *SIAM J. Comput.*, vol. 35, no. 4, pp. 903–931, 2006.

[14] L. Trevisan, "On uniform amplification of hardness in NP," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2005, pp. 31–38.

[15] R. Impagliazzo, R. Jaiswal, and V. Kabanets, "Approximate List-Decoding of Direct Product Codes and Uniform Hardness Amplification," *SIAM J. Comput.*, vol. 39, no. 2, pp. 564–605, 2009.

[16] 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.

[17] E. Viola, "The complexity of constructing pseudorandom generators from hard functions," *Computational Complexity*, vol. 13, no. 3-4, pp. 147–188, 2005.

[18] C. Lu, S. Tsai, and H. Wu, "On the Complexity of Hardness Amplification," *IEEE Trans. Information Theory*, vol. 54, no. 10, pp. 4575–4586, 2008.

[19] R. Shaltiel and E. Viola, "Hardness Amplification Proofs Require Majority," *SIAM J. Comput.*, vol. 39, no. 7, pp. 3122–3154, 2010.

[20] A. Grinberg, R. Shaltiel, and E. Viola, "Indistinguishability by Adaptive Procedures with Advice, and Lower Bounds on Hardness Amplification Proofs," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 956–966.

[21] J. Feigenbaum and L. Fortnow, "Random-Self-Reducibility of Complete Sets," *SIAM J. Comput.*, vol. 22, no. 5, pp. 994–1005, 1993.

[22] A. Bogdanov and L. Trevisan, "On Worst-Case to Average-Case Reductions for NP Problems," *SIAM J. Comput.*, vol. 36, no. 4, pp. 1119–1159, 2006.

[23] S. Hirahara, "Non-black-box Worst-case to Average-case Reductions within NP," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 247–258.

[24] B. A. Trakhtenbrot, "A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms," *IEEE Annals of the History of Computing*, vol. 6, no. 4, pp. 384–400, 1984.

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

[26] E. Allender, "The Complexity of Complexity," in *Computability and Complexity - Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*, 2017, pp. 79–94.

[27] S. Hirahara and R. Santhanam, "On the Average-Case Complexity of MCSP and Its Variants," in *Proceedings of the Computational Complexity Conference (CCC)*, 2017, pp. 7:1–7:20.

[28] 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.

[29] 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.

[30] M. Sudan, L. Trevisan, and S. P. Vadhan, "Pseudorandom Generators without the XOR Lemma," *J. Comput. Syst. Sci.*, vol. 62, no. 2, pp. 236–266, 2001.

[31] L. Trevisan and S. P. Vadhan, "Pseudorandomness and Average-Case Complexity Via Uniform Reductions," *Computational Complexity*, vol. 16, no. 4, pp. 331–364, 2007.

[32] R. Schuler and T. Yamakami, "Structural Average Case Complexity," *J. Comput. Syst. Sci.*, vol. 52, no. 2, pp. 308–348, 1996.

[33] 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.

[34] E. Allender, D. Holden, and V. Kabanets, "The Minimum Oracle Circuit Size Problem," *Computational Complexity*, vol. 26, no. 2, pp. 469–496, 2017.

[35] S. Hirahara and O. Watanabe, "On Nonadaptive Security Reductions of Hitting Set Generators," in *Proceedings of the Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM)*, 2020, pp. 15:1–15:14.

[36] 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.

[37] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[38] S. Hirahara, "Unexpected Power of Random Strings," in *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, 2020, pp. 41:1–41:13.

[39] ——, "Unexpected hardness results for Kolmogorov complexity under uniform reductions," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 2020, pp. 1038–1051.

[40] R. Impagliazzo, "Relativized Separations of Worst-

[46] L. F. C. Antunes and L. Fortnow, "Worst-Case Running Times for Average-Case Algorithms," in *Proceedings of the Conference on Computational Complexity* Case and Average-Case Complexities for NP," in *Proceedings of the Conference on Computational Complexity (CCC)*, 2011, pp. 104–114.

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

[42] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A Pseudorandom Generator from any One-way Function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[43] R. Impagliazzo, V. Kabanets, and I. Volkovich, "The Power of Natural Properties as Oracles," in *Proceedings of the Computational Complexity Conference (CCC)*, 2018, pp. 7:1–7:20.

[44] L. G. Valiant and V. V. Vazirani, "NP is as Easy as Detecting Unique Solutions," *Theor. Comput. Sci.*, vol. 47, no. 3, pp. 85–93, 1986.

[45] M. Li and P. M. B. Vitányi, "Average Case Complexity Under the Universal Distribution Equals Worst-Case Complexity," *Inf. Process. Lett.*, vol. 42, no. 3, pp. 145–149, 1992.
(CCC), 2009, pp. 298–303.

[47] L. Antunes, L. Fortnow, D. van Melkebeek, and N. V. Vinodchandran, "Computational depth: Concept and applications," *Theor. Comput. Sci.*, vol. 354, no. 3, pp. 391–404, 2006.

[48] 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.

[49] T. Watson, "Query Complexity in Errorless Hardness Amplification," *Comput. Complex.*, vol. 24, no. 4, pp. 823–850, 2015.