Symmetry of Information from Meta-Complexity

Shuichi Hirahara
National Institute of Informatics, Tokyo, Japan

Abstract

Symmetry of information for time-bounded Kolmogorov complexity is a hypothetical inequality that relates time-bounded Kolmogorov complexity and its conditional analogue. In 1992, Longpré and Watanabe showed that symmetry of information holds if NP is easy in the worst case, which has been the state of the art over the last three decades. In this paper, we significantly improve this result by showing that symmetry of information holds under the weaker assumption that NP is easy on average. In fact, our proof techniques are applicable to any resource-bounded Kolmogorov complexity and enable proving symmetry of information from an efficient algorithm that computes resource-bounded Kolmogorov complexity.

We demonstrate the significance of our proof techniques by presenting two applications. First, using that symmetry of information does not hold for Levin's Kt-complexity, we prove that randomized Kt-complexity cannot be computed in time $2^{o(n)}$ on inputs of length n, which improves the previous quasi-polynomial lower bound of Oliveira (ICALP 2019). Our proof implements Kolmogorov's insightful approach to the P versus NP problem in the case of randomized Kt-complexity. Second, we consider the question of excluding Heuristica, i.e., a world in which NP is easy on average but NP $\neq$ P, from Impagliazzo's five worlds: Using symmetry of information, we prove that Heuristica is excluded if the problem of approximating time-bounded conditional Kolmogorov complexity $\mathrm{K}^t(x \mid y)$ up to some additive error is NP-hard for $t \gg |y|$. We complement this result by proving NP-hardness of approximating *sublinear*-time-bounded conditional Kolmogorov complexity up to a multiplicative factor of $|x|^{1/(\log\log|x|)^{O(1)}}$ for $t \ll |y|$. Our NP-hardness proof presents a new connection between sublinear-time-bounded conditional Kolmogorov complexity and a secret sharing scheme.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases resource-bounded Kolmogorov complexity, average-case complexity, pseudorandomness, hardness of approximation, unconditional lower bound

Digital Object Identifier 10.4230/LIPIcs.CCC.2022.26

Funding JST, PRESTO Grant Number JPMJPR2024, Japan.

Acknowledgements I thank Mikito Nanashima for helpful discussion, Rahul Santhanam for raising the question of finding an equivalent notion of the weak universal heuristic scheme, which inspired Lemma 7.5, and anonymous reviewers for strongly inspiring me to include Appendix A.1.

1 Introduction

One of the basic facts in information theory is symmetry of mutual information: For two random variables X and Y, the mutual information $I(X : Y)$ is symmetric: $I(X : Y) = I(Y : X)$. In terms of Shannon entropy H(-), this equality is equivalent to

$$\mathrm{H}(Y) - \mathrm{H}(Y \mid X) = \mathrm{H}(X) - \mathrm{H}(X \mid Y).$$

An alternative way of measuring the amount of information is to use *Kolmogorov complexity*. The Kolmogorov complexity $\mathrm{K}(x)$ of a finite string $x \in \{0,1\}^*$ is defined to be the length of a shortest program that prints x. While Shannon entropy can measure the average amount of information in a *collection* of strings, the notion of Kolmogorov complexity enables quantifying the amount of information in an *individual* string, thereby giving a finer notion than Shannon entropy. Kolmogorov complexity has had fundamental impacts on theoretical computer science [55].

Kolmogorov and Levin [74] established a fundamental property of Kolmogorov complexity: *Symmetry of information for Kolmogorov complexity* states that

$$\mathrm{K}(y) - \mathrm{K}(y \mid x) = \mathrm{K}(x) - \mathrm{K}(x \mid y) \pm O(\log(|x| + |y|)) \tag{1}$$

for any strings x and $y \in \{0,1\}^*$. Here, $\mathrm{K}(x \mid y)$ denotes the conditional Kolmogorov complexity of x given y, i.e., the length of a shortest program that prints x given y as input.

One disadvantage of Kolmogorov complexity, which was already noted in the seminal paper of Kolmogorov [51], is that $\mathrm{K}(x)$ does not take into account the complexity of generating the string x. Kolmogorov suggested considering t-time-bounded Kolmogorov complexity, denoted by $\mathrm{K}^t(x)$, which is the shortest size of a program that prints x in time t. According to Levin [52], as early as 1967 Kolmogorov suggested time-bounded versions of symmetry of information as an interesting avenue of research. We consider the following time-bounded version of symmetry of information.[1]

▶ **Definition 1.1.** Symmetry of information for time-bounded Kolmogorov complexity (SoI) *refers to the following hypothesis: There exists a polynomial p such that for any strings $x \in \{0,1\}^*$ and $y \in \{0,1\}^*$, for every $t \geq |x| + |y|$,*

$$\mathrm{K}^{p(t)}(x \mid y) + \mathrm{K}^{p(t)}(y) - \log p(t) \leq \mathrm{K}^t(x, y). \tag{SoI}$$

Throughout this paper, we refer to this hypothesis as SoI.

Note that $\mathrm{K}^t(x,y) \leq \mathrm{K}^{t/4}(y \mid x) + \mathrm{K}^{t/4}(x) + O(1)$ unconditionally holds for all large t because strings x and y can be computed by running a program of length $\mathrm{K}^{t/4}(x)$ that outputs x in time $t/4$ and a program of length $\mathrm{K}^{t/4}(y \mid x)$ that takes x as input and outputs y in time $t/4$.[2] Consequently, SoI implies that

$$\mathrm{K}^{p(t)}(x \mid y) + \mathrm{K}^{p(t)}(y) \leq \mathrm{K}^{t/4}(y \mid x) + \mathrm{K}^{t/4}(x) + O(\log t),$$

which is a natural time-bounded analogue of Equation (1).

The original Kolmogorov–Levin proof of Equation (1) is based on an exhaustive search of all strings of a given length. By applying the original proof to a space-bounded setting, symmetry of information for *space-bounded* Kolmogorov complexity was proved by Longpré and Mocas [57]. Interestingly, as is repeatedly stated by Levin (e.g., in [53, 19, 54]), even before the notion of NP-completeness was formalized, Kolmogorov envisioned that investigating symmetry of information for time-bounded Kolmogorov complexity would be a good approach toward $\mathsf{P} \neq \mathsf{NP}$. To quote [53],

> "this information symmetry theorem may be a good test case to prove that for some tasks exhaustive search cannot be avoided (in today's terms, $\mathsf{P} \neq \mathsf{NP}$)."

[1] Previous works [57, 58] studied a slightly different version called *polynomial*-time-bounded symmetry of information, in which the time bound t of SoI is fixed to an arbitrary polynomial. SoI is stronger than polynomial-time-bounded symmetry of information, which makes Theorem 1.2 stronger. Moreover, it is not hard to observe that the results of [57, 58] hold for SoI. We also mention that our version of SoI is useful to show worst-case-to-average-case connections; see Section 7.

[2] For our definition of time-bounded Kolmogorov complexity (Definition 3.1), there is a larger overhead of the simulation of the programs. For simplicity, we ignore this minor issue in this introduction.

In 1992, Longpré and Watanabe [58][3] showed that $\mathsf{P} = \mathsf{NP}$ implies SoI, and that SoI implies the non-existence of one-way functions. Their results can be seen as formalizing Kolmogorov's insightful approach to $\mathsf{P} \neq \mathsf{NP}$: Finding a *single* sequence of pairs (x, y) of strings that violates SoI implies $\mathsf{P} \neq \mathsf{NP}$! Despite the rich literature on Kolmogorov complexity [55], providing an exact characterization of SoI remains elusive. In fact, it has been a long-standing open question to improve the results of [58] over the last three decades.

In this work, we improve the result of [58] by showing that SoI holds under the weaker assumption that NP is easy on average.

▶ **Theorem 1.2.** *If* $\mathsf{DistNP} \subseteq \mathsf{AvgP}$*, then SoI holds.*

Here, the statement $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ means that for every language L in NP and for every polynomial-time samplable distribution $\mathcal{D}$,[4] there exists an errorless heuristic scheme that solves L on inputs sampled from $\mathcal{D}$; we refer the reader to the survey of Bogdanov and Trevisan [16] for background on average-case complexity theory.

We find Theorem 1.2 surprising: The assumption of Theorem 1.2 is that NP is easy on *most* instances, whereas SoI is an inequality for *every* pair of strings x and y; thus, it is natural to expect that SoI is related to the worst-case complexity of some problem. Indeed, we show that SoI is closely related to the worst-case complexity of GapMINKT, which is the problem of approximating time-bounded Kolmogorov complexity. We also show that, under the plausible assumption that E requires exponential-size circuits, SoI is sandwiched between the existence of an *errorless* heuristic scheme (AvgP) for computing $\mathrm{K}^t(\text{-})$ and the existence of an *error-prone* heuristic scheme (HeurP) for computing $\mathrm{K}^t(\text{-})$, thereby narrowing SoI.

▶ **Theorem 1.3** (informal; see Theorem 8.2 for the formal version)**.** *Assume that* $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{\epsilon n})$ *for some constant* $\epsilon > 0$*. In the following list, we have 1* $\iff$ *2* $\implies$ *3* $\implies$ *4 and 3* $\implies$ *5.*

1. $\mathrm{GapMINKT} \in \mathsf{P}$.
2. *For every polynomial-time samplable distribution* $\mathcal{D}$ *and for all sufficiently large polynomials* t*, there exists an* errorless *heuristic scheme that computes* $\mathrm{K}^{t(n)}(x)$ *for a random input* $x \sim \mathcal{D}_n$*.*
3. *SoI holds.*
4. *For every polynomial-time samplable distribution* $\mathcal{D}$ *and for all sufficiently large polynomials* t*, there exists an* error-prone *heuristic scheme that computes* $\mathrm{K}^{t(n)}(x)$ *for a random input* $x \sim \mathcal{D}_n$*.*
5. $\mathrm{Gap}_\tau\mathrm{MINKT} \in \mathsf{DTIME}(2^{O(n/\log n)})$ *for some function* $\tau(n, t) = 2^{O(n/\log n)}$*.*

The difference between errorless and error-prone heuristics is as follows: Error-prone heuristics compute a problem on most instances but are allowed to output a wrong answer. Errorless heuristics also compute a problem on most instances; in addition, they must output either a correct answer or a special failure symbol "$\perp$". Hirahara and Santhanam [38] showed the equivalence between errorless and error-prone average-case complexities of any problem that admits an instance checker in the sense of Blum and Kannan [15]. In particular, Theorem 1.3 implies that SoI is exactly characterized by the worst-case complexity of GapMINKT if time-bounded Kolmogorov complexity admits an instance checker (and E requires exponential-size circuits). Whether an instance checker can be constructed or not is an interesting open question.

[3] The conference version of [58] was presented in ISAAC 1992.
[4] In fact, even if we restrict the distribution $\mathcal{D}$ to $\{\mathcal{U}, \mathcal{T}\}$, the statement remains the same. Here, $\mathcal{U}$ denotes the uniform distribution and $\mathcal{T}$ denotes the tally distribution supported only on $\{1\}^*$. This is a consequence of the theorems of Impagliazzo and Levin [45], Buhrman et al. [18]; see [35].

Previously, under the assumption that $\mathsf{DistNP} \subseteq \mathsf{AvgP}$, [35] showed *weak symmetry of information*, which states that $|x| + \mathrm{K}^{\mathsf{poly}(t)}(y) - O(\log t) \leq \mathrm{K}^t(x, y)$ holds with high probability over $x \sim \{0,1\}^n$ uniformly chosen at random. Our proof is largely inspired by the proof of weak symmetry of information. In fact, it is "not hard" to prove Theorem 1.2 itself using the proof techniques developed in [35][5]. Independently of this work, Goldberg and Kabanets [28] proved Theorem 1.2 using the same proof techniques and used it to provide an alternative proof of main results of [35].[6]

Our main contribution is to develop more general proof techniques that are applicable to any resource-bounded Kolmogorov complexity and to demonstrate their significance: In general, for any (randomized) resource-bounded Kolmogorov complexity $\mathrm{K}\mu$(-), we show that symmetry of information for $\mathrm{K}\mu$(-) follows from an efficient algorithm that computes $\mathrm{K}\mu$(-). Since symmetry of information is a fundamental property of Kolmogorov complexity, we expect that our new proof techniques will have a variety of applications in future. In the following two subsections, we present two specific applications that concern two central questions of computational complexity theory: (1) $\mathsf{P} \neq \mathsf{NP}$ and (2) $\mathsf{P} \neq \mathsf{NP} \implies \mathsf{DistNP} \not\subseteq \mathsf{AvgP}$.

1.1 Kolmogorov's Approach to the P versus NP Problem

Is Kolmogorov's approach to the P versus NP problem an illusion made in the premature era of theoretical computer science? We present strong evidence that this is not the case: Using our new proof techniques of proving symmetry of information, we demonstrate that Kolmogorov's insightful approach is indeed useful to prove lower bounds! Specifically, we consider a randomized version of Levin's Kt-complexity. The randomized Kt-complexity of a string x, denoted by $\mathrm{rKt}(x)$, is defined as the minimum of $|M| + \log t$ over all the randomized programs M that print x with probability at least $\frac{2}{3}$ in time t [62]. Let GapMrKtP be the promise problem of approximating $\mathrm{rKt}(x)$ on input x up to an additive error of $O(\log |x|)$. Using an exhaustive search, it is easy to observe that $\mathrm{GapMrKtP} \in \text{pr-}\mathsf{BPTIME}(2^{O(n)})$. We prove that this exhaustive search cannot be avoided.

▶ **Theorem 1.4.** $\mathrm{GapMrKtP} \notin \mathsf{i.o.BPTIME}(2^{\epsilon n})$ *for some constant* $\epsilon > 0$.

The proof of Theorem 1.4 is given by implementing Kolmogorov's approach in the case of rKt: Ronneburger [66] proved that symmetry of information for Levin's Kt-complexity does not hold. Using this result, we observe that symmetry of information for rKt is also false (conditionally). Then, using our proof techniques, we prove symmetry of information for rKt from an efficient hypothetical algorithm that computes GapMrKtP, which leads to a contradiction.

Previously, Oliveira [62] proved $\mathrm{GapMrKtP} \notin \text{pr-}\mathsf{BPTIME}(2^{\log^{O(1)} n})$ using a different proof technique. Theorem 1.4 improves this result in the following two respects:[7]

1. The lower bound is improved from a quasi-polynomial $2^{\log^{O(1)} n}$ to a strongly exponential $2^{\epsilon n}$, which is optimal up to a constant exponent.
2. Our lower bound works against algorithms that are correct infinitely often ($\mathsf{i.o.}$).

[5] if the feasibility of proving Theorem 1.2 is given as advice; see Appendix A.1 for more comments.

[6] Independently, we also find a similar alternative proof; see Section 7.

[7] A caveat is that our lower bound is applicable to BPP, i.e., randomized algorithms that satisfy the promise of BPP-type algorithms over all the inputs, whereas the lower bound of [62] is also applicable to pr-BPP. In this respect, these two lower bounds are incomparable. We leave an open problem of improving our lower bound to $\mathsf{i.o.pr\text{-}BPTIME}(2^{\epsilon n})$. However, we note that $\mathsf{BPP} = \mathsf{P} = \text{pr-}\mathsf{BPP}$ under the plausible assumption that E requires exponential-size circuits [46], in which case there is no difference between the two lower bounds.

The fact that Theorem 1.4 improves the previous state-of-the-art result of [62] indicates the significance of the new approach of proving lower bounds via symmetry of information. [34] developed yet another proof technique of showing lower bounds for resource-bounded Kolmogorov complexity and showed that $\mathrm{K}^{t(|x|)}(x)$ cannot be computed in polynomial time if $t(n) = n^{\omega(1)}$. Interestingly, this previous proof technique does not seem to be applicable to the setting of Theorem 1.4,[8] which suggests the "novelty" of Kolmogorov's approach.

Why was Kolmogorov's approach not implemented before? Previously, the only proof technique of proving symmetry of information was due to Kolmogorov and Levin [74]; for example, the aforementioned work of Longpré and Watanabe [58] showed symmetry of information from a hypothetical efficient algorithm that computes some problem in PH, by applying the original Kolmogorov–Levin proof to the resource-bounded case. It was not known if symmetry of information can be proved from an efficient algorithm that computes resource-bounded Kolmogorov complexity. Our general proof techniques of showing symmetry of information from efficient algorithms make it possible to implement Kolmogorov's insightful approach, for the first time. Our results demonstrate the significance of Kolmogorov's approach and hint that it may indeed lead to the resolution of the P versus NP problem in future.

1.2 Toward Excluding Heuristica

One of the central open questions in computational complexity theory is to show the equivalence between the worst-case and average-case complexities of NP, e.g., $\mathsf{P} = \mathsf{NP} \iff \mathsf{DistNP} \subseteq \mathsf{AvgP}$. Equivalently, this question is well known as whether Heuristica can be excluded from Impagliazzo's five possible worlds [43]. Heuristica is a hypothetical world in which $\mathsf{P} \neq \mathsf{NP}$ and $\mathsf{DistNP} \subseteq \mathsf{AvgP}$. There are at least three types of results explaining the difficulty of excluding Heuristica: Standard proof techniques for relating worst-case and average-case complexities, such as black-box nonadaptive reductions [23, 17], hardness amplification procedures [73, 72], and relativizing proof techniques [44, 36], are known to be incapable of excluding Heuristica. To make progress on this central problem, we need to develop new types of proof techniques that are not subject to any of these technical barriers.

Recently, several new proof techniques of analyzing average-case complexity have been developed based on *meta-complexity*. Meta-complexity refers to the complexity of a problem that asks for complexity. One representative example of meta-computational problems is MINKT [50], which asks to compute $\mathrm{K}^t(x)$ given $(x, 1^t)$ as input. [31] developed a non-black-box reduction technique that goes beyond the limits of black-box reductions presented by Feigenbaum and Fortnow [23], Bogdanov and Trevisan [17], and showed GapMINKT $\in \mathsf{P}$ if $\mathsf{DistNP} \subseteq \mathsf{AvgP}$. Informally, GapMINKT is the problem of approximating the time-bounded Kolmogorov complexity $\mathrm{K}^t(x)$ up to an additive error of $O(\log t)$ on input $(x, 1^t)$.[9] As a consequence, Heuristica can be excluded if GapMINKT is NP-hard. Subsequently, [35] developed proof techniques that simultaneously overcome the limits of black-box reductions [23, 17] and the impossibility of hardness amplification procedures [73, 72], and proved (among other results) that $\mathsf{DistNP} \not\subseteq \mathsf{AvgP}$ follows from the worst-case assumption that $\mathsf{UP} \not\subseteq \mathsf{DTIME}(2^{O(n/\log n)})$. The next goal along this research line is to develop a proof

[8] By combining the infinitely-often pseudo-deterministic subexponential-time algorithm of Oliveira and Santhanam [63] with [34], it is possible to prove GapMrKtP $\not\in$ pr-$\mathsf{BPTIME}(2^{\log^{O(1)} n})$. However, this ends up with essentially the same proof as [62].

[9] More precisely, GapMINKT is the problem of computing a value v such that $\mathrm{K}^{p(t)}(x) - \log p(t) \leq v \leq \mathrm{K}^t(x)$ on input $(x, 1^t)$ such that $|x| \geq t$, where p is some polynomial; see Fact 3.4.

technique that overcomes the limits of relativizing barriers, for which quantitatively tight results are known: Building on the work of Impagliazzo [44], Hirahara and Nanashima [36] showed that a relativizing proof technique is incapable of improving the time complexity $2^{O(n/\log n)}$ achieved in [35] to $2^{o(n/\log n)}$. Chen et al. [20] *bypassed* this relativization barrier by showing that $\mathsf{NTIME}[n]$ is hard on average for quasi-linear-time algorithms if $\mathsf{UP} \not\subseteq \mathsf{DTIME}\left(2^{\sqrt{n \log n}}\right)$. To make further progress, we would need to develop a proof technique that *overcomes*[10] the relativization barrier.

In search of non-relativizing proof techniques, we review another line of research on meta-complexity. A long-standing open question on meta-complexity, which dates back to as early as the 1960s [69, 8], is to prove NP-hardness of meta-computational problems, such as MINKT. Recently, there has been substantial progress on this question in a line of research [7, 37, 40, 42, 41, 56, 4]. A salient feature of the proof techniques developed in this line of research is that these are apparently non-relativizing. Specifically, Ko [50] constructed an oracle under which $\mathsf{NP} \neq \mathsf{P}$ but GapMINKT is easy, thereby showing that a relativizing proof technique is incapable of showing NP-hardness of GapMINKT. In contrast, building on the work of Ilango [40], Allender et al. [4], Liu and Pass [56] showed NP-hardness of the problem of computing sublinear-time-bounded conditional Kolmogorov complexity, which we denote by MINcKT ("c" stands for "conditional").

Toward excluding Heuristica, we aim at combining the two lines of research reviewed above. One interpretation of SoI is an approximate equality between time-bounded conditional Kolmogorov complexity and its unconditional version. Specifically, SoI implies that

$$\begin{aligned} \mathrm{K}^{p(t)}(x \mid y) \leq \mathrm{K}^t(x,y) - \mathrm{K}^{p(t)}(y) + O(\log t) &\leq \mathrm{K}^{t/4}(x \mid y) + \mathrm{K}^{t/4}(y) - \mathrm{K}^{p(t)}(y) + O(\log t). \\ &= \mathrm{K}^{t/4}(x \mid y) + \mathrm{cd}^{t/4,p(t)}(y) + O(\log t). \end{aligned} \tag{2}$$

Here, $\mathrm{cd}^{s,t}(y) := \mathrm{K}^s(y) - \mathrm{K}^t(y)$ is called the *(s,t)-time-bounded computational depth* of y [9, 35] and is known to be small for most strings y [9, 11] (see Lemma 8.4 for a formal statement). Equation (2) means that the conditional Kolmogorov complexity $\mathrm{K}^{t'}(x \mid y)$ can be approximated up to an additive error of $\mathrm{cd}^{t/4,p(t)}(y) + O(\log t)$ for some $t' \in [t/4, p(t)]$ if time-bounded Kolmogorov complexity can be efficiently computed.[11] In other words, SoI enables reducing the problem of approximating *conditional* Kolmogorov complexity to the problem of approximating *unconditional* Kolmogorov complexity; i.e., GapMINcKT is reducible to GapMINKT. As a consequence of Theorem 1.2, we generalize the approach of [31] for excluding Heuristica as follows:

▶ **Theorem 1.5.** *If* Gap_τMINcKT *is* NP*-hard under randomized reductions for every polynomial τ, then Heuristica does not exist; that is,* $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ *if and only if* $\mathsf{P} = \mathsf{NP}$*. Here, the problem* Gap_τMINcKT *asks to compute an integer $k \in \mathbb{N}$ such that*

$$\mathrm{K}^{\tau(|x|,|y|,t)}(x \mid y) - \log \tau(|x|,|y|,t) \leq k \leq \mathrm{K}^t(x \mid y) + \mathrm{cd}^{t,\tau(|x|,|y|,t)}(y)$$

on input $(x, y, 1^t)$.

[10] We say that a proof technique A *overcomes* a barrier B if A proves a statement S such that proof techniques subject to B are incapable of proving S. The results of [20] do not necessarily overcome the relativization barriers of [44, 36] because the conclusion that $\mathsf{NTIME}[n]$ is hard on average is weaker than $\mathsf{DistNP} \not\subseteq \mathsf{AvgP}$.

[11] It is unlikely that the additive error term $\mathrm{cd}^{t/4,p(t)}(y)$ can be improved by using relativizing proof techniques; see Remark 6.4.

More generally, our proof techniques provide a general approach of reducing conditional resource-bounded Kolmogorov complexity $\mathrm{K}\mu(\text{-} \mid \text{-})$ to its unconditional analogue $\mathrm{K}\mu(\text{-})$ for every resource-bounded Kolmogorov complexity $\mathrm{K}\mu$: Using a hypothetical efficient algorithm that computes $\mathrm{K}\mu(\text{-})$, we prove symmetry of information for $\mathrm{K}\mu$. Then, we use symmetry of information to reduce conditional complexity $\mathrm{K}\mu(\text{-} \mid \text{-})$ to $\mathrm{K}\mu(\text{-})$.

Next, we investigate whether $\mathrm{Gap}_\tau\mathrm{MINcKT}$ can be proved to be NP-hard. Building on the proof techniques developed in [40, 4, 56], we prove NP-hardness of $\mathrm{Gap}_\tau\mathrm{MINcKT}$ if the approximation parameter $\tau(|x|, |y|, t)$ is sublinear in $|y|$, e.g., $\tau(|x|, |y|, t) \le |x|^{O(1)} \cdot |y|^{0.99} \cdot t^{O(1)}$.

▶ **Theorem 1.6** (informal; see Theorem 6.6 for the formal version)**.** *Let $c > 1$ be an arbitrary constant. Let $\tau\colon \mathbb{N}^3 \to \mathbb{N}$ be a function such that $\tau(n, m, t) \le n^c \cdot m^{1-1/c} \cdot t^c$ for all large n, m, and $t \in \mathbb{N}$. Then, $\mathrm{Gap}_\tau\mathrm{MINcKT}$ is NP-hard under randomized polynomial-time reductions. Moreover, it is NP-hard to approximate $\mathrm{K}^t(x \mid y)$ to within a factor of $|x|^{1/(\log\log|x|)^{O(1)}}$.*

The contributions of Theorem 1.6 are two-fold.

1. We improve an inapproximability factor. Most previous hardness proofs in the literature [7, 37, 40, 42, 41, 56, 4] reduce the set cover problem to meta-computational problems. In particular, the inapproximability factor obtained in [4, 56] is at most $O(\log|x|)$. Our inapproximability factor $|x|^{1/(\log\log|x|)^{O(1)}}$ is a nearly exponential improvement and is close to the trivial upper bound of $|x|$. In addition, we prove the NP-hardness of approximating $\mathrm{K}^t(x \mid y)$ even when there is an additive error term $\mathrm{cd}^{t,\tau(|x|,|y|,t)}(y)$.
   Our strong inapproximability is not only a quantitative improvement but also a *qualitative* improvement. In fact, the two previous results [4, 56] prove NP-hardness of two different versions of conditional Kolmogorov complexity, McKTP and MINcKT. Here, McKTP is the problem of computing the KT-complexity of x given y, i.e., the minimum of $|M| + t$ over all programs M that compute x given y as input in time t. Because of the previous weak inapproximability, these two results are proved by different proofs. Our inapproximability is strong enough to prove NP-hardness of McKTP and MINcKT simultaneously.
2. We present a new connection between a secret sharing scheme and sublinear-time-bounded conditional Kolmogorov complexity. This new connection abstracts essential proof ideas implicitly developed in the line of research and enables us to provide a streamlined proof.

Perspective

How should we interpret our results? Optimistically, our results indicate that excluding Heuristica might be reachable by combining current proof techniques. We are not aware of any technical barrier that suggests Theorems 1.5 and 1.6 cannot inherently be matched. It is currently plausible that all the technical barriers to excluding Heuristica that we are aware of ([23, 17, 73, 72, 44, 36]) can be overcame in this way. Pessimistically, our results can be understood as indicating the difference between sublinear-time-bounded Kolmogorov complexity and time-bounded Kolmogorov complexity. However, we emphasize that the fact that Theorems 1.2 and 1.5 do not work for sublinear-time bounded Kolmogorov complexity is not an inherent limitation of our proof techniques. Our proof techniques make it possible to prove symmetry of information even for some version of sublinear-time bounded Kolmogorov complexity *with error* [25] by using a pseudorandom generator construction from average-case hardness. Whether there is a technical barrier that explains the inherent difficulty of the current approach remains to be explored.[12]

[12] A candidate is the limits of oracle-independent reductions of [39].

1.3 Related Work

Symmetry of Information

Lee and Romashchenko [52] unconditionally showed that for every polynomial p, for some polynomial q, for any strings x and y of length n,

$$\mathrm{KAMD}^{q(n)}(x \mid y) + \mathrm{KAMD}^{q(n)}(y) - O(\log^3 n) \le \mathrm{K}^{p(n)}(x, y).$$

Here, KAMD denotes an Arthur–Merlin variant of distinguishing Kolmogorov complexity. Consequently, symmetry of information holds up to an additive error of $O(\log^3 n)$ under the assumption that $\mathrm{K}^{q'(n)}(x \mid y) \le \mathrm{KAMD}^{q(n)}(x \mid y) + O(\log n)$ holds for any strings x and y. They also showed that this assumption is in fact equivalent to $\mathsf{P} = \mathsf{NP}$ and hence gave no improvement over the conditional result of [58].

Symmetry of Information and Hardness

Symmetry of information is used to show hardness results for Kolmogorov complexity [34]. Similarly, an inequality analogous to SoI is used to show NP-hardness of a multi-output variant of MCSP [42]. The *Minimum Circuit Size Problem* (MCSP [47]) is the meta-computational problem that asks to compute the size of a minimum circuit that computes a given function $f \colon \{0,1\}^n \to \{0,1\}$ represented as the truth table of length 2^n. Ilango [40] showed that a conditional variant of MCSP is NP-hard under randomized reductions. Subsequently, Ilango et al. [42] proved NP-hardness of a multi-output variant of MCSP. Their proof is based on an inequality reminiscent of SoI, which enables translating the hardness result for the conditional variant of MCSP [40] to an unconditional variant.

The Shortest Vector Problem

The complexity landscape about GapMINcKT is reminiscent of the shortest vector problem GapSVP. If "approximation factors" are small, then GapMINcKT and GapSVP are NP-hard [30]. If "approximation factors" are large, then these problems admit worst-case-to-average-case connections. However, there are fundamental differences: On one hand, GapSVP is known to be in $\mathsf{NP} \cap \mathsf{coNP}$ for a large approximation factor [29, 1] and thus is unlikely to be NP-hard. Moreover, this phenomenon is inherent: The limits of black-box reductions presented by Bogdanov and Trevisan [17] show that any problem reducible to DistNP by nonadaptive black-box reductions must be in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. The worst-case-to-average-case connections presented for GapSVP are given by black-box reductions [2, 60] and thus subject to this barrier. On the other hand, the worst-case-to-average-case connection for GapMINcKT (Theorem 1.5) is not subject to the barrier of [17] since it is proved by a *non-black-box* reduction, i.e., a reduction that exploits the efficiency of a hypothetical heuristic algorithm for DistNP. There is no evidence against NP-hardness of $\mathrm{Gap}_\tau\mathrm{MINcKT}$ for any large polynomial τ; on the contrary, the NP-hardness of $\mathrm{Gap}_\tau\mathrm{MINcKT}$ for sublinear-time-bounded functions τ can be seen as supporting evidence.

2 Proof Techniques

In this section, we outline our proof techniques.

2.1 Symmetry of Information in Heuristica

We first sketch a proof of Theorem 1.2, which shows SoI in Heuristica. In doing so, we present a general technique of proving symmetry of information from any efficient algorithm that computes resource-bounded Kolmogorov complexity and any pseudorandom generator construction. We need two lemmas from previous work, which we review below.

Our proof of Theorem 1.2 is based on the meta-complexity of time-bounded Kolmogorov complexity: We crucially use the fact that time-bounded Kolmogorov complexity can be approximated in Heuristica.

▶ **Lemma 2.1** ([31, 33]). *If* $\mathsf{DistNP} \subseteq \mathsf{AvgP}$*, then* $\mathrm{GapMINKT} \in \mathsf{P}$.

This result makes it possible to *approximate* $\mathrm{K}^t(x)$. For the purpose of exposition, we assume below that there exists an algorithm which, on input (x, t), computes $\mathrm{K}^t(x)$ *exactly* in time $\mathsf{poly}(|x|, t)$. It is worth emphasizing that our proof deviates from the original Kolmogorov–Levin proof in this respect: Our proof is "meta-computational" in that we use $\mathrm{GapMINKT} \in \mathsf{P}$, whereas the Kolmogorov–Levin proof cannot be "meta-computational" since the resource-unbounded Kolmogorov complexity K(-) cannot be computed in finite time steps.

The second lemma is the reconstruction property of a *k-wise direct product generator* [34], which turned out to be a fundamental tool for analyzing Kolmogorov complexity [33, 65, 32, 35]. A k-wise direct product generator $\mathrm{DP}_k\colon \{0,1\}^n \times \{0,1\}^{nk} \to \{0,1\}^{nk+k}$ is defined as follows:

$$\mathrm{DP}_k(x; z) := (z^1, \dots, z^k, \langle z^1, x\rangle, \dots, \langle z^k, x\rangle)$$

for every $x \in \{0,1\}^n$ and every $z = (z^1, \dots, z^k) \in (\{0,1\}^n)^k$, where $\langle x, y\rangle$ denotes the inner product of x and $y \in \{0,1\}^n$ over GF(2), i.e., $\langle x, y\rangle := (\sum_{i=1}^n x_i y_i) \mod 2$. The k-wise direct product generator $\mathrm{DP}_k(x; \text{-})$ is a pseudorandom generator secure against an algorithm D if $\mathrm{K}(x \mid D) > k + O(\log |x|)$. More formally, we have the following property:

▶ **Lemma 2.2** (Deterministic Reconstruction for DP_k; see [35]). *Assume that* $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{\epsilon n})$ *for some constant* $\epsilon > 0$*. Then, there exists a polynomial* p *such that, for every* $n \in \mathbb{N}$*,* $x \in \{0,1\}^n$*, parameters* $k, \epsilon^{-1}, s \in \mathbb{N}$*, and for every randomized circuit* D *of size* s *such that*

$$\left| \Pr_{z,r}\left[D(\mathrm{DP}_k(x; z); r) = 1\right] - \Pr_{w,r}\left[D(w; r) = 1\right] \right| \geq \epsilon,$$

where $z \sim \{0,1\}^{nk}$*,* $w \sim \{0,1\}^{nk+k}$*, and* $r \sim \{0,1\}^s$*, it holds that*

$$\mathrm{K}^{p(ns/\epsilon)}(x \mid D) \leq k + \log p(ns/\epsilon).$$

The assumption of Lemma 2.2 is satisfied in Heuristica, as Buhrman et al. [18] showed that $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{\epsilon n})$ for some constant $\epsilon > 0$ if $\mathsf{DistNP} \subseteq \mathsf{AvgP}$. We mention in passing that the proof of Lemma 2.1 also uses Lemma 2.2.

Now, we sketch the proof of SoI from an efficient algorithm that computes K^t(-) and the pseudorandom generator construction DP(-;-). Fix strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ and an integer $t \geq n + m$. We claim

$$\mathrm{K}^{q(t)}(x \mid y) + \mathrm{K}^{q(t)}(y) - \log q(t) \leq \mathrm{K}^t(x, y)$$

for some universal polynomial q. The proof is given by analyzing the following three values for $z \sim \{0,1\}^{nk}$, $w \sim \{0,1\}^{nk+k}$, $z' \sim \{0,1\}^{m\ell}$, and $w' \sim \{0,1\}^{m\ell+\ell}$:

$$\begin{array}{llll}
\mathrm{K}^{t'}(& \mathrm{DP}_k(x;z), & \mathrm{DP}_\ell(y;z') &), \\
\mathrm{K}^{t'}(& w, & \mathrm{DP}_\ell(y;z') &), \\
\mathrm{K}^{t'}(& w, & w' &),
\end{array}$$

where t', k, and ℓ are parameters chosen later.

First, by a standard counting argument, we have

$$\mathrm{K}^{t'}(w,w') \approx |w| + |w'| \tag{3}$$

with high probability over the random choice of w and w'.

Next, by the contrapositive of Lemma 2.2, $\mathrm{DP}_\ell(y;\text{-})$ is a pseudorandom generator secure against uniform algorithms if $\ell \ll \mathrm{K}^{p(t)}(y)$, where p is some polynomial; thus, we obtain

$$\mathrm{K}^{t'}(w, \mathrm{DP}_\ell(y;z')) \approx \mathrm{K}^{t'}(w,w'). \tag{4}$$

In more detail, consider a randomized circuit D that takes w' as input as well as random bits w and checks whether $\mathrm{K}^{t'}(w,w') \geq \theta$ for a threshold θ. By Lemma 2.2, if D can distinguishes $\mathrm{DP}_\ell(y;z')$ and w', we would get $\mathrm{K}^{p(t)}(y) \leq \ell + \log p(t)$. We choose $\ell := \mathrm{K}^{p(t)}(y) - \log p(t) - 1$ so that this inequality does not hold. Equation (4) follows from the fact that D cannot distinguish the pseudorandom distribution $\mathrm{DP}_\ell(y;z')$ from the uniform distribution w'.

By combining Equations (3) and (4), we conclude that

$$\mathrm{K}^{t'}(w, \mathrm{DP}_\ell(y;z')) \gtrsim |w| + |w'| = |z| + k + |z'| + \ell \approx |z| + k + |z'| + \mathrm{K}^{p(t)}(y) \tag{5}$$

with high probability over the random choice of w and z'.

On the other hand, observe that for some $t' := \mathsf{poly}(t)$,

$$\mathrm{K}^{t'}(\mathrm{DP}_k(x;z), \mathrm{DP}_\ell(y;z')) \leq \mathrm{K}^t(x,y) + |z| + |z'| + O(\log n) \tag{6}$$

holds because the strings $\mathrm{DP}_k(x;z)$ and $\mathrm{DP}_\ell(y;z')$ can be computed from k, ℓ, z, z', and a program of size $\mathrm{K}^t(x,y)$ that outputs (x,y) in time t.

Comparing Equations (5) and (6), for a sufficiently large $k := \mathrm{K}^t(x,y) - \mathrm{K}^{p(t)}(y) + O(\log t)$, there exists a threshold θ' such that $\mathrm{K}^{t'}(w, \mathrm{DP}_\ell(y;z')) \geq \theta'$ with high probability over the random choice of w and z', whereas $\mathrm{K}^{t'}(\mathrm{DP}_k(x;z), \mathrm{DP}_\ell(y;z')) < \theta'$ for every z and z'. Let D_y be a randomized circuit that takes an input w and random bits z' and checks whether $\mathrm{K}^{t'}(w, \mathrm{DP}_\ell(y;z')) < \theta'$. Then, we obtain

$$1 = \Pr_{z,z'}[D_y(\mathrm{DP}_k(x;z);z') = 1] \gg \Pr_{w,z'}[D_y(w;z') = 1] \approx 0$$

Using Lemma 2.2, we obtain

$$\mathrm{K}^{\mathsf{poly}(t)}(x \mid D_y) \leq k + O(\log t) = \mathrm{K}^t(x,y) - \mathrm{K}^{p(t)}(y) + O(\log t).$$

Since the circuit D_y can be efficiently constructed from y, we have $\mathrm{K}^t(D_y \mid y) \leq O(\log t)$. It follows that for some large polynomial q,

$$\mathrm{K}^{q(t)}(x \mid y) \leq \mathrm{K}^t(x \mid D_y) + \mathrm{K}^t(D_y \mid y) + O(1) \leq \mathrm{K}^t(x,y) - \mathrm{K}^{p(t)}(y) + O(\log t)$$

as desired. This completes the proof of Theorem 1.2, except that the circuits D and D_y must carefully be defined using an algorithm for GapMINKT. A complete proof can be found in Section 4. We also present a simple proof of Theorem 1.2 based on weak symmetry of information of [35] in Appendix A and clarify that the techniques of proving Theorem 1.2 were already developed in [35].

2.2 Lower Bounds for rKt via Kolmogorov's Approach

Observe the generality of the proof technique described in Section 2.1: The only two ingredients of the proof of symmetry of information are (1) a hypothetical algorithm that computes resource-bounded Kolmogorov complexity and (2) a pseudorandom generator construction. The same proof strategy can be applied to any resource-bounded Kolmogorov complexity; the tightness of the resulting SoI is determined by the advice complexity of a pseudorandom generator construction.

We review the notion of pseudorandom generator construction and its advice complexity. A *pseudorandom generator construction* $G\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is an efficiently computable function that has a (randomized) *reconstruction procedure* R with the following property: For every $x \in \{0,1\}^n$, if an oracle D distinguishes the output distribution $G(x;\text{-})$ from the uniform distribution, then there exists an advice string $\alpha \in \{0,1\}^a$ such that the randomized algorithm R outputs x given α as advice with high probability. The length a of the advice string is referred to as the *advice complexity* of G. Trevisan and Vadhan [71] showed that the advice complexity a is always at least $m-d-3$. In the case of the k-wise direct product generator $\mathrm{DP}_k(x;\text{-})$, the advice complexity is $k+O(\log n)$, which is close to the lower bound of $m-d-3=k-3$. The fact that the additive term of SoI is $O(\log t)$ comes from the nearly optimal advice complexity of $\mathrm{DP}_k(x;\text{-})$. The disadvantage of DP_k, however, is that the seed length $d=nk$ is too large, which prevents us from obtaining the tight lower bound of Theorem 1.4.

There are several different constructions of pseudorandom generators in the literature (e.g., [70, 64, 68, 34]). For example, the pseudorandom generator construction G of Raz et al. [64] satisfies the following properties: The advice complexity of G is at most $m+O(\log^3 n)$. In particular, if some t-time algorithm distinguishes the output distribution $G(x;\text{-})$ from the uniform distribution, then $\mathrm{rKt}(x) \le m + O(\log^3 n) + O(\log t)$. Moreover, the seed length d is $O(\log^3 n)$, which is small enough for our application.

Now, we sketch the proof of Theorem 1.4. For simplicity, we claim that the problem MrKtP of computing rKt *exactly* cannot be solved in time $2^{o(n)}$. Toward a contradiction, assume $\mathrm{MrKtP} \in \mathsf{BPTIME}(2^{o(n)})$. Applying the general proof technique of Section 2.1 to (1) the algorithm that computes MrKtP and (2) the pseudorandom generator construction G, we obtain symmetry of information for rKt:

$$\mathrm{rKt}(x \mid y) + \mathrm{rKt}(y) \le \mathrm{rKt}(x,y) + O(\log^3 n) + o(n)$$

for any strings x and y of length n (Lemma 5.6). Next, we construct a pair (x,y) that violates this symmetry of information. Ronneburger [66] constructed such a pair in the case of Kt-complexity. We apply the same proof idea to the case of rKt-complexity. The pair (x,y) is constructed as follows: Exhaustively search a string $y \in \{0,1\}^n$ such that $\mathrm{rKt}(y) \ge n$ in time $2^{n+o(n)}$. Next, exhaustively search a string $x \in \{0,1\}^n$ such that $\mathrm{rKt}(x,y) \ge 2n - o(n)$ in time $2^{n+o(n)}$. The existence of such a string x is guaranteed by symmetry of information because $\mathrm{rKt}(x \mid y) + \mathrm{rKt}(y) \ge 2n$ holds for a string x such that $\mathrm{rKt}(x \mid y) \ge n$. Overall, the pair (x,y) is constructed in time

$$2^{n+o(n)} + 2^{n+o(n)} = 2^{n+o(n)},$$

However, by the definition of rKt, we obtain $\mathrm{rKt}(x,y) \le O(\log n) + \log 2^{n+o(n)} \le n + o(n)$, which contradicts the lower bound of $\mathrm{rKt}(x,y)$. We present the details in Section 5.

2.3 Secret Sharing Schemes and Sublinear-Time-Bounded Conditional Kolmogorov Complexity

We now present a new connection between a secret sharing scheme and sublinear-time-bounded conditional Kolmogorov complexity.

To do so, however, we first need to clarify the definition of $\mathrm{K}^t(x \mid y)$ for sublinear-time bounds t (i.e., $t \ll |x|$ and $t \ll |y|$). For the standard computational model of Turing machines, it takes at least $|y|$ time steps just to read the entire input y. To define sublinear-time-bounded Kolmogorov complexity in a meaningful way, we assume that algorithms are given random access to each bit of y; in other words, the input y is given as the oracle that, on query $i \in \{1, \dots, |y|, |y|+1\}$, answers the i-th bit of y if $i \le |y|$ and a special stop symbol "$\perp$" if $i = |y|+1$. Similarly, it takes at least $|x|$ steps for a Turing machine to output all the bits of x; instead, we assume that an algorithm is given an index $i \in \{1, \dots, |x|, |x|+1\}$ as input and is asked to compute the i-th bit of x if $i \le |x|$ and "$\perp$" if $i = |x|+1$. This notion of sublinear-time-bounded Kolmogorov complexity is standard in the literature (e.g., [3, 32]). See Section 3 for the formal definition.

Next, we review the notion of secret sharing scheme. A *secret sharing scheme*, introduced by Blakley [14], Shamir [67], is an algorithm that shares a secret among n parties so that any authorized set of parties can reconstruct the secret but no unauthorized set of parties can obtain any information about the secret. Specifically, let $\mathcal{A} \subseteq 2^{[n]}$ be an *access structure*, i.e., a monotone collection of subsets of $[n]$. We say that a pair $(\mathrm{Share}, \mathrm{Rec})$ of algorithms is a secret sharing scheme for $\mathcal{A}$ if it satisfies the following two properties:

Correctness: For a secret $x \in \{0,1\}^*$ and an authorized set $T \in \mathcal{A}$ of parties,

$$\mathrm{Rec}(\mathrm{Share}(x)_T) = x,$$

where $\mathrm{Share}(x)_T$ denotes the set of shares that are shared to some party $i \in T$.

Perfect Privacy: For an unauthorized set $T \notin \mathcal{A}$ of parties, X and $\mathrm{Share}(X)_T$ are statistically independent for every random variable X.

We refer the reader to the survey of Beimel [12] for more background on secret sharing schemes.

We present a generic reduction that takes an arbitrary secret sharing scheme and reduces the problem of computing the minimum size of an authorized set to sublinear-time-bounded conditional Kolmogorov complexity.

▷ Claim 2.3 (informal; see Theorem 6.14 for the formal version). Let $\mathcal{A} = \{\mathcal{A}_\varphi\}_{\varphi \in \{0,1\}^*}$ be a family of access structures for which there exists an efficient secret sharing scheme. Let $w(\mathcal{A}_\varphi) := \min_{T \in \mathcal{A}_\varphi} |T|$. Then, there is a randomized polynomial-time reduction that takes φ as input and outputs (x, y, t, λ) such that

$$w(\mathcal{A}_\varphi) \cdot \lambda \approx \mathrm{K}^t(x \mid y),$$

where $t \ll |y|$.

Benaloh and Leichter [13] showed that for any monotone formula φ, there exists an efficient secret sharing scheme for the access structure $\mathcal{A}_\varphi$ represented by φ.[13] Applying their secret sharing scheme to Claim 2.3, we obtain a reduction from the Minimum Monotone Satisfying

[13] More generally, Karchmer and Wigderson [49] showed that the access structure represented by any monotone span program admits an efficient secret sharing scheme.

Assignment (MMSA) problem to GapMINcKT. MMSA is known to be hard to approximate within a factor of $|\varphi|^{1/(\log\log|\varphi|)^{O(1)}}$ even for depth-3 monotone formulas[14] [22, 21], which will complete the proof of Theorem 1.6.

We now sketch a proof for Claim 2.3. Here is an outline of the reduction, which generalizes the reduction presented by [4, 56]: We share a random string $x \sim \{0,1\}^\ell$ among n parties using a secret sharing scheme $(\mathrm{Share}_\varphi, \mathrm{Rec}_\varphi)$ for $\mathcal{A}_\varphi$. Let $(s_1, \dots, s_n) := \mathrm{Share}_\varphi(x)$, where s_i is the share of the i-th party. Randomly choose $k_1, \dots, k_n \sim \{0,1\}^\lambda$, which are called "keys". We define an oracle $y' \colon \{0,1\}^\lambda \to \{0,1\}^m$ such that $y'(k_i) := s_i$ and $y'(k) := 0^m$ if $k \notin \{k_i \mid i \in [n]\}$. Let $y \in \{0,1\}^{m \cdot 2^\lambda}$ denote the truth table of y', i.e., the concatenation of the values of $y(k)$ for all the inputs $k \in \{0,1\}^\lambda$ in lexicographical order. We claim that for some time bound t,

$$\mathrm{K}^t(x \mid y) \lesssim w(\mathcal{A}_\varphi) \cdot \lambda + |\varphi| \tag{7}$$

and that

$$\mathrm{K}^t(x \mid y) \gtrsim w(\mathcal{A}_\varphi) \cdot \lambda. \tag{8}$$

It is easy to see Equation (7): Let $T \in \mathcal{A}_\varphi$ be an authorized set of parties. Consider an oracle program that, given oracle access to y, takes the set $\{k_i \in \{0,1\}^\lambda \mid i \in T\}$ as input, queries k_i to the oracle y and obtains the i-th share $s_i = y'(k_i)$ for all $i \in T$, and then reconstructs the secret $x = \mathrm{Rec}_\varphi(\{s_i \mid i \in T\})$. The size of this program is at most approximately $|T| \cdot \lambda + |\varphi|$. To see (8), assume, toward a contradiction, that there exists an oracle program M of size $\ll w(\mathcal{A}_\varphi) \cdot \lambda$ such that M prints x in time t given oracle access to y. Since $t \ll |y|$ and $k_1, \dots, k_n$ are chosen randomly, it can be shown that the program M of size $|M|$ can read at most approximately $|M|/\lambda$ shares from y. The intuition behind this is that the i-th share s_i cannot be accessed without knowing the i-th key k_i, which must be hard-wired in the program M. Let T be the set of indices $i \in [n]$ such that the i-th share s_i is read by M. Then, we have $|T| \lesssim |M|/\lambda \ll w(\mathcal{A}_\varphi)$, which implies that $T \notin \mathcal{A}_\varphi$. However, by the perfect privacy of the secret sharing scheme, it is not possible to reconstruct x from $\{s_i \mid i \in T\}$, which is a contradiction.

There are two issues in the proof sketch above. First, it can be the case that $w(\mathcal{A}_\varphi)\cdot\lambda \ll |\varphi|$, in which case Equation (7) is too loose. Second, we need to show that an additive error term $\mathrm{cd}^{t,\tau(|x|,|y|,t)}(y)$ is small compared to $w(\mathcal{A}_\varphi) \cdot \lambda$. Fortunately, there is a simple trick that simultaneously fixes these two issues: Share D independent secrets $x^1, \dots, x^D \sim \{0,1\}^\ell$ among n parties, construct oracles $y^1, \dots, y^D$ for each secret, and reduce φ to $(x, y, t, D\lambda)$ for $x := (x^1, \dots, x^D)$ and $y := (y^1, \dots, y^D)$. Then we get

$$w(\mathcal{A}_\varphi) \cdot \lambda \lesssim \frac{1}{D} \cdot \mathrm{K}^t(x \mid y) \lesssim w(\mathcal{A}_\varphi) \cdot \lambda + \frac{|\varphi|}{D},$$

whose last term is negligible for a large enough D. This fixes the first issue. Moreover, it can be shown that the "amortized" computational depth $\frac{1}{D} \cdot \mathrm{cd}^t(y)$ goes to 0 as D increases, which fixes the second issue.

Finally, we explain the relationship between our reduction and the previous reductions in [40, 4, 56]. Previously, the set cover problem was reduced to MINcKT and a conditional variant of MCSP by implicitly using a secret sharing scheme that can share a random string and satisfies the imperfect privacy. Specifically, let $S_1, \cdots, S_n \subseteq [m]$ be an instance

[14] It is worth mentioning that MMSA for depth-2 monotone formulas corresponds to the set cover problem.

of the set cover problem. Let $\mathcal{A}$ be the collection of the sets $T \subseteq [n]$ that form a cover; i.e., $\bigcup_{i \in T} S_i = [m]$. Then, in [40, 4, 56], the following secret sharing scheme for $\mathcal{A}$ was implicitly used: For a secret $x = (x^1, \ldots, x^m) \sim (\{0,1\}^\ell)^m$, the i-th share s_i is defined to be $\{x^j \mid j \in S_i\}$ for each $i \in [n]$. This secret sharing scheme satisfies correctness, i.e., the property that x can be reconstructed from $s_T := \{s_i \mid i \in T\}$ for any authorized set $T \in \mathcal{A}$; it also satisfies imperfect privacy in the sense that, for any unauthorized set $T \notin \mathcal{A}$, there exists $j \in [m] \setminus \bigcup_{i \in T} S_i$ such that no information about x^j is leaked from s_T. Our generalized reduction abstracts the essential ideas developed in [40, 4, 56] and would be useful for making further progress on the final goal of proving NP-hardness of GapMINKT and excluding Heuristica.

Organization

In Section 4, we present a formal proof of SoI in Heuristica. In Section 5, we prove the lower bound for rKt. In Section 6, we examine the complexity of conditional Kolmogorov complexity. In Section 7, we present an application of SoI to average-case complexity. In Section 8, we present several statements that follow from SoI.

3 Preliminaries

Notation

$[n]$ denotes $\{1, \ldots, n\}$. For a function $p \colon \mathbb{N} \to \mathbb{N}$ and $i \in \mathbb{N}$, the function $p^{(i)} \colon \mathbb{N} \to \mathbb{N}$ is recursively defined as follows: $p^{(0)}(n) := n$ and $p^{(i+1)}(n) := p^{(i)}(p(n))$ for every $n \in \mathbb{N}$.

Kolmogorov Complexity

For a string $x \in \{0,1\}^*$ and an index $i \in \mathbb{N}$, let x_i denote the i-th bit of x if $i \leq |x|$ and $\perp$ if $i > |x|$. We fix an efficient universal Turing machine U. Time-bounded Kolmogorov complexity is formally defined as follows.

▶ **Definition 3.1** (Time-bounded Kolmogorov complexity). *For strings $x, y \in \{0,1\}^*$, a time bound $t \in \mathbb{N} \cup \{\infty\}$, and an oracle A, the A-oracle t-time-bounded Kolmogorov complexity of x given y is defined as*

$$\mathrm{K}^{t,A}(x \mid y) := \min\left\{|d| \;\middle|\; U^{A,y,d} \text{ outputs } x_i \text{ on input } i \text{ in time } t \text{ for every } i \in [1, \ldots, |x|+1]\right\}.$$

Here, $U^{A,y,d}$ indicates that the universal Turing machine is given oracle access to A and each bit of y and $d \in \{0,1\}^*$. We omit the superscript A if $A = \emptyset$, the superscript t if $t = \infty$, and "$\mid y$" if y is the empty string.

A simple counting argument implies the following basic fact of Kolmogorov-randomness.

▶ **Fact 3.2.** *For any integer $s \geq 1$ and any string $y \in \{0,1\}^*$, the number of strings $x \in \{0,1\}^*$ such that $\mathrm{K}(x \mid y) < s$ is less than 2^s.*

Proof. The number of programs of length less than s is at most $\sum_{i=0}^{s-1} 2^i < 2^s$. ◀

▶ **Definition 3.3** ([50, 33]). *For a polynomial $\tau \colon \mathbb{N} \to \mathbb{N}$, let*

$$\Pi_{\mathrm{YES}} := \left\{(x, 1^t, 1^s) \;\middle|\; \mathrm{K}^t(x) \leq s\right\},$$
$$\Pi_{\mathrm{NO}} := \left\{(x, 1^t, 1^s) \;\middle|\; \mathrm{K}^{\tau(|x|,t)}(x) > s + \log \tau(|x|, t)\right\}.$$

We define $\mathrm{Gap}_\tau\mathrm{MINKT}$ *to be the promise problem* $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$. *We say that* $\mathrm{GapMINKT} \in \mathsf{P}$ *if there exists some polynomial* τ *such that* $\mathrm{Gap}_\tau\mathrm{MINKT} \in \mathsf{P}$.

The problem GapMINKT can be equivalently formulated as follows.

▶ **Fact 3.4** ([35]). *The following are equivalent.*
1. $\mathrm{GapMINKT} \in \mathsf{P}$.
2. *There exist a polynomial-time algorithm* $\widetilde{\mathrm{K}}$ *and a polynomial* p *such that*

$$\mathrm{K}^{p(t)}(x) - \log p(t) \le \widetilde{\mathrm{K}}(x, 1^t) \le \mathrm{K}^t(x)$$

for every string $x \in \{0,1\}^*$ *and every integer* $t \ge |x|$.

We recall the notion of time-bounded computational depth.

▶ **Definition 3.5** (Time-Bounded Computational Depth [9, 35]). *For time bounds* $s \in \mathbb{N}$ *and* $t \in \mathbb{N} \cup \{\infty\}$ *and a string* $x \in \{0,1\}^*$, *the* (s,t)-time-bounded computational depth *of* x *is defined as*

$$\mathrm{cd}^{s,t}(x) := \mathrm{K}^s(x) - \mathrm{K}^t(x).$$

We omit the superscript t *if* $t = \infty$.

4 Symmetry of Information from Average-Case Easiness of NP

In this section, we present a formal proof of Theorem 1.2. In fact, under a plausible assumption that E requires exponential-size circuits to compute, we prove that SoI follows from an approximation algorithm for time-bounded Kolmogorov complexity.

▶ **Theorem 4.1.** *If* $\mathrm{GapMINKT} \in \mathsf{P}$ *and* $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{\epsilon n})$ *for some constant* $\epsilon > 0$, *then SoI holds.*

Theorem 4.1 immediately implies Theorem 1.2.

Proof of Theorem 1.2. It is known that the two hypotheses of Theorem 4.1 are implied by the assumption that NP is easy on average: Buhrman et al. [18] showed that $\mathsf{E} \not\subseteq \bigcap_{\epsilon>0} \mathsf{i.o.SIZE}(2^{\epsilon n})$ if $\mathsf{DistNP} \subseteq \mathsf{AvgP}$; We also have $\mathrm{GapMINKT} \in \mathsf{P}$ if $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ by Lemma 2.1. ◀

We now present the formal proof of Theorem 4.1.

Proof of Theorem 4.1. Let $\widetilde{\mathrm{K}}$ be the polynomial-time algorithm of Fact 3.4 which satisfies the property that there exists a polynomial p such that for every $x \in \{0,1\}^*$ and every $t \ge |x|$,

$$\mathrm{K}^{p(t)}(x) - \log p(t) \le \widetilde{\mathrm{K}}(x; 1^t) \le \mathrm{K}^t(x) \tag{9}$$

Fix strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ and an integer $t \ge n + m$. The proof of SoI is given by analyzing the following three values for $z \sim \{0,1\}^{nk}$, $w \sim \{0,1\}^{nk+k}$, $z' \sim \{0,1\}^{m\ell}$, and $w' \sim \{0,1\}^{m\ell+\ell}$:

$$\begin{array}{llll}
\widetilde{\mathrm{K}}(& \mathrm{DP}_k(x;z), & \mathrm{DP}_\ell(y;z') & ;1^{t'}), \\
\widetilde{\mathrm{K}}(& w, & \mathrm{DP}_\ell(y;z') & ;1^{t'}), \\
\widetilde{\mathrm{K}}(& w, & w' & ;1^{t'}),
\end{array}$$

where $t' = t^{O(1)}$, $k \approx \mathrm{K}^t(x,y) - \ell$, and $\ell \approx \mathrm{K}^{\mathsf{poly}(t)}(y)$ are parameters chosen later.

First, observe that Fact 3.2 implies that $\mathrm{K}(w,w') \geq |w| + |w'| - 2$ with probability at least $\frac{3}{4}$. Let $\theta := |w| + |w'| - 2 - \log p(t')$; using Equation (9), we obtain

$$\Pr_{w,w'}\left[\widetilde{\mathrm{K}}(w,w';1^{t'}) \geq \theta\right] \geq \frac{3}{4}. \tag{10}$$

Next, we set the parameter ℓ to $\mathrm{K}^{p'(t)}(y) - \log p'(t) - 1$, where p' is some large polynomial. Consider a randomized circuit D that takes w' as input as well as random bits w and outputs 1 if and only if $\widetilde{\mathrm{K}}(w,w';1^{t'}) \geq \theta$. By the contrapositive of Lemma 2.2, $\mathrm{DP}_\ell(y;\text{-})$ is a pseudorandom generator secure against D; i.e., D cannot distinguish $\mathrm{DP}_\ell(y;z')$ and w' in the sense that

$$\left|\Pr_{w,z'}\left[\widetilde{\mathrm{K}}(w,\mathrm{DP}_\ell(y;z');1^{t'}) \geq \theta\right] - \Pr_{w,w'}\left[\widetilde{\mathrm{K}}(w,w';1^{t'}) \geq \theta\right]\right| < \frac{1}{4},$$

which, together with Equation (10), implies that

$$\Pr_{w,z'}\left[\widetilde{\mathrm{K}}(w,\mathrm{DP}_\ell(y;z');1^{t'}) \geq \theta\right] \geq \frac{1}{2}.$$

Finally, we compare $\widetilde{\mathrm{K}}(w,\mathrm{DP}_\ell(y;z');1^{t'})$ with $\widetilde{\mathrm{K}}(\mathrm{DP}_k(x;z),\mathrm{DP}_\ell(y;z');1^{t'})$. On one hand, since $|w| = |z| + k$ and $|w'| = |z'| + \ell$, we have

$$\Pr_{w,z'}\left[\widetilde{\mathrm{K}}(w,\mathrm{DP}_\ell(y;z');1^{t'}) \geq |z| + |z'| + k + \ell - 2 - \log p(t')\right] \geq \frac{1}{2}. \tag{11}$$

On the other hand, observe that for some $t' := \mathsf{poly}(t)$,

$$\widetilde{\mathrm{K}}(\mathrm{DP}_k(x;z),\mathrm{DP}_\ell(y;z');1^{t'}) \leq \mathrm{K}^{t'}(\mathrm{DP}_k(x;z),\mathrm{DP}_\ell(y;z')) \leq \mathrm{K}^t(x,y) + |z| + |z'| + O(\log n)$$

holds because the strings $\mathrm{DP}_k(x;z)$ and $\mathrm{DP}_\ell(y;z')$ can be computed from k, ℓ, z, z', and a program of size $\mathrm{K}^t(x,y)$ that outputs (x,y) in time t. We now set $k := \mathrm{K}^t(x,y) - \ell + O(\log t)$ so that

$$\Pr_{z,z'}\left[\widetilde{\mathrm{K}}(\mathrm{DP}_k(x;z),\mathrm{DP}_\ell(y;z');1^{t'}) < |z| + |z'| + k + \ell - 2 - \log p(t')\right] = 1. \tag{12}$$

Let D_y be a randomized circuit that takes an input w and random bits z' and outputs 1 if and only if $\widetilde{\mathrm{K}}(w,\mathrm{DP}_\ell(y;z');1^{t'}) < |z|+|z'|+k+\ell-2-\log p(t')$. It follows from Equations (11) and (12) that

$$\Pr_{z,z'}\left[D_y(\mathrm{DP}_k(x;z);z') = 1\right] - \Pr_{w,z'}\left[D_y(w;z') = 1\right] \geq 1 - \frac{1}{2} = \frac{1}{2}.$$

Using Lemma 2.2, we obtain

$$\mathrm{K}^{\mathsf{poly}(t)}(x \mid D_y) \leq k + O(\log t) = \mathrm{K}^t(x,y) - \mathrm{K}^{p'(t)}(y) + O(\log t).$$

It follows that for some large polynomial q,

$$\begin{aligned}\mathrm{K}^{q(t)}(x \mid y) \leq \mathrm{K}^t(x \mid D_y) + \mathrm{K}^t(D_y \mid y) + O(1) &\leq \mathrm{K}^t(x,y) - \mathrm{K}^{p'(t)}(y) + O(\log t)\\ &\leq \mathrm{K}^t(x,y) - \mathrm{K}^{q(t)}(y) + \log q(t)\end{aligned}$$

as desired. ◀

5 A Lower Bound for Randomized Kt Complexity

Here, we implement Kolmogorov's insightful approach to the P versus NP problem for randomized Kt complexity. We recall the formal definition of rKt, which was introduced by Oliveira [62].

▶ **Definition 5.1** ([62]). *For every $x \in \{0,1\}^*$ and every $\lambda \in [0,1]$, the randomized time-bounded Kolmogorov complexity $\mathrm{rKt}_\lambda(x)$ of x is defined as*

$$\mathrm{rKt}_\lambda(x \mid y) := \min\left\{ |d| + \lceil \log t \rceil \;\middle|\; \Pr_{r \sim \{0,1\}^t}[\, U(d,y,r) \textit{ outputs } x \textit{ in time } t\,] \geq \lambda \right\}.$$

We omit the subscript "λ" if $\lambda = \frac{2}{3}$ and "$\mid y$" if y is the empty string. The language MrKtP *is defined as*

$$\mathrm{MrKtP} := \{(x, 1^s) \mid \mathrm{rKt}(x) \leq s\}.$$

The definition of rKt_λ is robust with respect to a choice of λ.

▶ **Lemma 5.2** (Lu and Oliveira [59]). *For all $x \in \{0,1\}^*$ and $\lambda \in (0,1]$,*

$$\mathrm{rKt}(x) \leq \mathrm{rKt}_\lambda(x) + O(\log(1/\lambda)).$$

For simplicity, we first prove the following weaker lower bound than Theorem 1.4.

▶ **Theorem 5.3.** *There exists a constant $\epsilon > 0$ such that* $\mathrm{MrKtP} \notin \mathsf{BPTIME}(2^{\epsilon n})$.

In what follows, let $\epsilon > 0$ be a sufficiently small positive constant. The O notation below does not depend on ϵ. We say that a *pseudo-deterministic algorithm M computes y on input x* [27] if

$$\Pr_M[M(x) = y] \geq \frac{2}{3},$$

where the probability is taken over an internal coin flip sequence of the randomized algorithm M. We use the following pseudorandom generator construction.

▶ **Lemma 5.4** (see the proof of [32, Theorem 4.7]). *For all sufficiently large $n, m \in \mathbb{N}$ such that $m \leq 2n$, there exists a triple $(G, A, R^{(\text{-})})$ such that*

$$\begin{aligned} G&\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m, \\ A&\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m, \\ R^{(\text{-})}&\colon \{0,1\}^m \times \{0,1\}^d \times \{0,1\}^r \to (\{0,1\}^n)^L, \end{aligned}$$

and for every $x \in \{0,1\}^n$ and any $D\colon \{0,1\}^m \to \{0,1\}$ such that

$$\Pr_{\substack{z \sim \{0,1\}^d \\ w' \sim \{0,1\}^{O(m)}}}[D(G(x;z);w') = 1] - \Pr_{\substack{w \sim \{0,1\}^m \\ w' \sim \{0,1\}^{O(m)}}}[D(w;w') = 1] \geq \frac{1}{m},$$

it holds that

$$\Pr_{\substack{w \sim \{0,1\}^r \\ z \sim \{0,1\}^d}}\left[x \in R^D(A(x,z), z, w)\right] \geq \frac{1}{2m^2}.$$

Here, we have $d = O(\log^3 n)$, $r = O(m)$, and $L = \mathsf{poly}(m)$. Moreover, G and A can be computed in time $\mathsf{poly}(n)$ and R^D can be computed in time $\mathsf{poly}(n)$ with oracle access to D.

Proof Sketch. The construction is given by the improvement of Trevisan's extractor [70] given by Raz et al. [64]. Specifically, we encode x using a list-decodable error-correcting code and regard the encoding of x as the truth table of a hard function f. Then, the pseudorandom generator construction G is defined as the Nisan–Wigderson generator [61] instantiated with the function f and the weak design of [64]. ([32, Theorem 4.7] is stated as a black-box *hitting set generator* construction; however, it is easy to observe that the construction actually provides a *pseudorandom generator* construction, which is a stronger object.) ◀

▶ **Corollary 5.5.** *Under the same hypothesis of Lemma 5.4, it holds that*

$$\mathrm{rKt}^D(x) \le m + O(\log^3 n).$$

Proof. By an averaging argument, there exists a string $z \in \{0,1\}^d$ such that

$$\Pr_w \left[x \in R^D(A(x,z),z,w)\right] \ge \frac{1}{2m^2}.$$

Let $\alpha := A(x,z) \in \{0,1\}^m$. The string x can be described by the following pseudo-deterministic algorithm M: M takes the advice string $\alpha \in \{0,1\}^m$ and $z \in \{0,1\}^d$ as input, picks $w \sim \{0,1\}^r$ randomly, and outputs a random element of $R^D(\alpha,z,w)$. Since the list size is at most $L = \mathsf{poly}(m)$, the probability that the output of M is equal to x is at least $\lambda := \frac{1}{2m^2} \cdot \frac{1}{L}$. Therefore, we obtain $\mathrm{rKt}_\lambda(x) \le m + d + O(\log n)$. The result follows from Lemma 5.2. ◀

▶ **Lemma 5.6.** *If* $\mathrm{MrKtP} \in \mathsf{BPTIME}(2^{\epsilon n})$*, then for all sufficiently large* $n \in \mathbb{N}$ *and any strings* $x \in \{0,1\}^n$ *and* $y \in \{0,1\}^n$*, it holds that*

$$\mathrm{rKt}(x \mid y) + \mathrm{rKt}(y) \le \mathrm{rKt}(x,y) + O(\epsilon n).$$

Proof. We may assume without loss of generality that $\mathrm{rKt}(y) \gg \log^3 n$. Let m and m' be parameters chosen later. Let $G\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and $G'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$ be the pseudorandom generator construction of Lemma 5.4 with parameters m and m', respectively.

Fix $z \in \{0,1\}^d$ and $z' \in \{0,1\}^{d'}$. Since the strings $(G(x;z), G'(y;z'))$ can be described by z, z', and a program that computes (x,y), we obtain

$$\mathrm{rKt}(G(x;z), G'(y;z')) \le \mathrm{rKt}(x,y) + |z| + |z'| + O(\log n) = \mathrm{rKt}(x,y) + O(\log^3 n). \tag{13}$$

Pick $w \sim \{0,1\}^m$ and $z' \sim \{0,1\}^{d'}$ randomly. We claim that $\mathrm{rKt}(w, G'(y;z')) \ge |w| + |w'| - O(\log n)$ with high probability. Toward a contradiction, assume that

$$\Pr_{w,z'}\left[\mathrm{rKt}(w, G'(y;z')) < |w| + |w'| - O(\log n)\right] \ge \frac{1}{2}.$$

By Fact 3.2, we have

$$\Pr_{w,w'}\left[\mathrm{rKt}(w,w') \ge |w| + |w'| - O(\log n)\right] \ge 1 - o(1).$$

We define an oracle D so that $D(w';w) := 1$ if and only if $\mathrm{rKt}(w,w') < |w| + |w'| - O(\log n)$. Then, the two inequalities above show that D distinguishes the output distribution of $G'(y;\text{-})$ from the uniform distribution. By Corollary 5.5, we obtain

$$\mathrm{rKt}^D(y) \le m' + O(\log^3 n).$$

Using the assumption that MrKtP $\in$ BPTIME$(2^{\epsilon n})$, the oracle D can be computed by a randomized algorithm running in time $2^{O(\epsilon m')}$. It follows that

$$\mathrm{rKt}(y) \le m' + O(\epsilon m') = (1 + O(\epsilon)) \cdot m'.$$

We choose m' so that this inequality does not hold; that is,

$$m' := (1 - O(\epsilon)) \cdot \mathrm{rKt}(y).$$

Then, we obtain a contradiction and conclude that

$$\Pr_{w,z'}[\mathrm{rKt}(w, G'(y; z')) \ge m + m' - O(\log n)] \ge \frac{1}{2}. \tag{14}$$

Define D_y to be an oracle such that $D_y(w; z') := 1$ if and only if $\mathrm{rKt}(w, G'(y; z')) \le \mathrm{rKt}(x, y) + O(\log^3 n)$. By Equations (14) and (13), D_y distinguishes the output distribution of $G(x; \text{-})$ from the uniform distribution if

$$m + m' - O(\log n) \ge \mathrm{rKt}(x, y) + O(\log^3 n).$$

We define $m := \mathrm{rKt}(x, y) - m' + O(\log^3 n)$ so that this inequality holds. By Corollary 5.5, we obtain

$$\mathrm{rKt}^{D_y}(x) \le m + O(\log^3 n).$$

Since the oracle D_y can be computed in time $2^{O(\epsilon m)}$ given y as hard-wired input, we conclude that

$$\begin{aligned}\mathrm{rKt}(x \mid y) &\le m + O(\epsilon m). \\ &\le \mathrm{rKt}(x, y) - m' + O(\epsilon n). \\ &\le \mathrm{rKt}(x, y) - (1 - O(\epsilon)) \cdot \mathrm{rKt}(y) + O(\epsilon n). \\ &\le \mathrm{rKt}(x, y) - \mathrm{rKt}(y) + O(\epsilon n).\end{aligned}$$ ◀

Proof of Theorem 5.3. The outline of the proof is as follows: First, we find a string $y \in \{0,1\}^n$ such that $\mathrm{rKt}(y) \ge n$ by an exhaustive search over all strings $y \in \{0,1\}^n$. Next, we find a string $x \in \{0,1\}^n$ such that $\mathrm{rKt}(x, y) \ge 2n - O(\epsilon n)$ by an exhaustive search over all strings $x \in \{0,1\}^n$. The existence of such a string x is guaranteed by Lemma 5.6. Finally, we observe that the pair (x, y) can be pseudo-deterministically computed in time $2^{n+O(\epsilon n)}$, which contradicts the lower bound on $\mathrm{rKt}(x, y)$. Details follow.

Using the assumption that MrKtP $\in$ BPTIME$(2^{\epsilon n})$, let M be the randomized algorithm that computes $\mathrm{rKt}(y)$ on input y of length n in time $2^{O(\epsilon n)}$ with probability at least $1 - 2^{-2n}$. Fix $n \in \mathbb{N}$. Let y_n be the lexicographically first string $y \in \{0,1\}^n$ such that $\mathrm{rKt}(y) \ge n$. Note that the existence of such a string y_n is guaranteed by Fact 3.2.

We claim that there exists a pseudo-deterministic algorithm M_1 that computes y_n on input $n \in \mathbb{N}$ in time $2^{n+O(\epsilon n)}$. For all strings $y \in \{0,1\}^n$ in lexicographical order, the algorithm M_1 tests whether $\mathrm{rKt}(y) \ge n$ using M, and outputs the first string y that passes this test. Since the error probability of M is at most 2^{-2n}, by a union bound, the algorithm M_1 computes y_n pseudo-deterministically with probability $1 - 2^{-n}$. The running time of M_1 is at most $2^{n+O(\epsilon n)}$.

Next, let x_n be the lexicographically first string $x \in \{0,1\}^n$ such that

$$\mathrm{rKt}(x, y_n) \ge 2n - c \cdot \epsilon n, \tag{15}$$

where c is a large constant to be chosen independently of ϵ. We prove that x_n is well defined, i.e., there exists a string $x \in \{0,1\}^n$ such that $\mathrm{rKt}(x, y_n) \geq 2n - c \cdot \epsilon n$. By Fact 3.2, there exists a string $x \in \{0,1\}^n$ such that $\mathrm{rKt}(x \mid y) \geq n$. By Lemma 5.6,

$$\mathrm{rKt}(x,y) + O(\epsilon n) \geq \mathrm{rKt}(x \mid y) + \mathrm{rKt}(y) \geq 2n.$$

Choosing a large constant c, we obtain $\mathrm{rKt}(x,y) \geq 2n - c \cdot \epsilon n$, as desired.

We claim that there exists a pseudo-deterministic algorithm M_2 that computes x_n given y_n as input in time $2^{n+O(\epsilon n)}$. For all strings $x \in \{0,1\}^n$ in lexicographical order, the algorithm M_2 tests whether $\mathrm{rKt}(x, y_n) \geq 2n - c \cdot \epsilon n$ using M, and outputs the first string x that passes this test. Using a union bound, the error probability of M_2 is bounded by 2^{-n}. The running time of M_2 is at most $2^{n+O(\epsilon n)}$.

Finally, by combining the algorithms M_1 and M_2, we obtain an algorithm that computes (x_n, y_n) pseudo-deterministically in time $2^{n+O(\epsilon n)}$. By the definition of rKt, this implies that

$$\mathrm{rKt}(x_n, y_n) \leq n + O(\epsilon n).$$

We also have $\mathrm{rKt}(x_n, y_n) \geq 2n - O(\epsilon n)$ by Equation (15). Therefore, we obtain $2n - O(\epsilon n) \leq n + O(\epsilon n)$, which is a contradiction for a sufficiently small constant $\epsilon > 0$. ◀

We observe that MrKtP is in the exponential-time variant of PP.

▶ **Proposition 5.7.** MrKtP $\in$ PEXP.

Proof. Since PP is closed under truth-table reductions [26], it suffices to show that MrKtP is reducible to PP via an exponential-time truth-table reduction. Let $(x, 1^s)$ be an input such that $x \in \{0,1\}^n$ and $s \in \mathbb{N}$. For each string $d \in \{0,1\}^*$ of length at most s, we ask the PP oracle whether $U(d,r)$ outputs x in time t for at least a $\frac{2}{3}$-fraction of $r \in \{0,1\}^t$, where $t := 2^{s-|d|}$. We accept the input $(x, 1^s)$ if and only if a positive answer is returned from the oracle. ◀

We expect that Theorem 5.3 can be extended to a lower bound against PP-type algorithms. However, the original motivation of Oliveira [62] is to study a gap version of MrKtP, for which the upper bound of pr-BPE can be proved. Specifically, let GapMrKtP denote the promise problem $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ such that $\Pi_{\mathrm{YES}} = \{(x,s) \mid \mathrm{rKt}(x) \leq s\}$ and $\Pi_{\mathrm{NO}} = \{(x,s) \mid \mathrm{rKt}(x) > s + c \log |x|\}$, where c is a sufficiently large constant. One can observe GapMrKtP $\in$ pr-BPE [62]. Theorem 5.3 can be extended to this promise problem:

▶ **Theorem 5.8** (Theorem 1.4, restated)**.** GapMrKtP $\notin$ i.o.BPTIME$(2^{\epsilon n})$ *for some constant* $\epsilon > 0$.

Here, abusing a notation, for a class $\mathfrak{C}$ of languages, we say that a promise problem $\Pi \in \mathfrak{C}$ if there exists a language $L \in \mathfrak{C}$ such that $\Pi_{\mathrm{YES}} \subseteq L \subseteq \{0,1\}^* \setminus \Pi_{\mathrm{NO}}$ for $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}}) = \Pi$. The lower bound of Theorem 5.8 does work against a bounded probabilistic algorithm that satisfies the promise of BPP-type algorithms over *all* the inputs. In contrast, the upper bound pr-BPE = pr-BPTIME$(2^{O(n)})$ holds only for an algorithm that may not satisfy the promise of BPP-type algorithms for *some* inputs. This leaves a qualitative gap between the lower bound and the upper bound on GapMrKtP; closing this gap is an interesting open question. We note, however, that there is no gap under the plausible assumption that E $\not\subseteq$ i.o.SIZE$(2^{\epsilon n})$ for some constant $\epsilon > 0$, in which case pr-BPP = P = BPP [46].

Proof of Theorem 5.8. The proof is essentially the same with Theorem 5.3. We only explain how to modify the proof of Theorem 5.3.

Toward a contradiction, we assume the existence of a randomized algorithm M that computes some language L *infinitely often*, where L is a language consistent with GapMrKtP. For all large $n \in \mathbb{N}$ and every $x \in \{0,1\}^n$ and $s \in [n]$, we assume that the binary encoding of (x, s) is exactly equal to $m(n)$, where $m(n) = \Theta(n)$ is some function. This ensures that for infinitely many $n \in \mathbb{N}$, the algorithm M computes L correctly on input (x, s) for every $x \in \{0,1\}^n$ and every $s \in [n]$.

First, we explain how to extend the lower bound of Theorem 5.3 to the infinitely-often version. Let n be the length of $x_n \in \{0,1\}^n$ and $y_n \in \{0,1\}^n$ constructed in the proof of Theorem 5.3. By inspection, in the proof of Theorem 5.3, we use the algorithm M on inputs of length at most n', where $n' = O(n)$. Any input x of length less than n' can be padded to an input of length n' by mapping x to $x' := x1^{n'-|x|}$. This ensures that the length of any input to M that we use in the proof is exactly equal to n'. Moreover, the correctness of the proof remains unchanged because $\mathrm{rKt}(x) = \mathrm{rKt}(x') \pm O(\log n)$.

Second, we explain how to deal with the promise problem. In the proof of Theorem 5.3, we exhaustively search the lexicographically first string $y_n \in \{0,1\}^n$ such that $\mathrm{rKt}(y_n) \geq n$. Instead, we search the lexicographically first string $y_n \in \{0,1\}^n$ such that $(y_n, n-2c\cdot\log n) \notin L$. Since $L \subseteq \{0,1\}^* \setminus \Pi_{\mathrm{No}}$, this ensures that $\mathrm{rKt}(y_n) \geq n - c\log n$. Similarly, we search the lexicographically first string $x_n \in \{0,1\}^n$ such that $((x_n, y_n), 2n - O(\epsilon n)) \notin L$. This ensures that $\mathrm{rKt}(x_n, y_n) \geq 2n - O(\epsilon n)$. Since the language L can be decided by the randomized algorithm M, we can compute the pair (x_n, y_n) pseudo-deterministically in time $2^{n+O(\epsilon n)}$. This is a contradiction. ◀

6 The Complexity of Conditional Kolmogorov Complexity

In this section, we examine the complexity of the problem GapMINcKT of approximating time-bounded conditional Kolmogorov complexity.

▶ **Definition 6.1.** *For a polynomial $\tau\colon \mathbb{N}^3 \to \mathbb{N}$, we define the promise problem* $\mathrm{Gap}_\tau\mathrm{MINcKT}$ *to be* $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ *such that*

$$\Pi_{\mathrm{YES}} := \left\{ (x, y, 1^t, 1^s) \;\middle|\; \mathrm{K}^t(x \mid y) \leq s - \mathrm{cd}^{t,\tau(|x|,|y|,t)}(y) \right\},$$
$$\Pi_{\mathrm{NO}} := \left\{ (x, y, 1^t, 1^s) \;\middle|\; \mathrm{K}^{\tau(|x|,|y|,t)}(x \mid y) > s + \log \tau(|x|,|y|,t) \right\}.$$

6.1 Approximating Conditional Kolmogorov Complexity in Heuristica

We observe that SoI enables reducing GapMINcKT to GapMINKT.

▶ **Proposition 6.2.** *Assume that SoI holds. If* $\mathrm{GapMINKT} \in \mathsf{P}$*, then there exists a polynomial* $\tau\colon \mathbb{N}^3 \to \mathbb{N}$ *such that* $\mathrm{Gap}_\tau\mathrm{MINcKT} \in \mathsf{P}$.

Proof. Let $\widetilde{\mathrm{K}}$ be the polynomial-time algorithm of Fact 3.4 such that for some polynomial p,

$$\mathrm{K}^{p(t)}(x) - \log p(t) \leq \widetilde{\mathrm{K}}(x, 1^t) \leq \mathrm{K}^t(x)$$

for every string $x \in \{0,1\}^*$ and every integer $t \geq |x|$. Applying this inequality to xy and y, we obtain

$$\mathrm{K}^{p(t)}(xy) - \log p(t) \leq \widetilde{\mathrm{K}}(xy, 1^t) \leq \mathrm{K}^t(xy),$$
$$\mathrm{K}^{p(t)}(y) - \log p(t) \leq \widetilde{\mathrm{K}}(y, 1^t) \leq \mathrm{K}^t(y)$$

for every $t \geq |x| + |y|$. Using these inequalities, we next analyze $\widetilde{\mathrm{K}}(xy, 1^{p(t)}) - \widetilde{\mathrm{K}}(y, 1^{p^{(3)}(t)})$. On one hand, we have

$$\begin{aligned}\widetilde{\mathrm{K}}(xy, 1^{p(t)}) - \widetilde{\mathrm{K}}(y, 1^{p^{(3)}(t)}) &\leq \mathrm{K}^{p(t)}(xy) - \mathrm{K}^{p^{(4)}(t)}(y) + \log p^{(4)}(t) \\ &\leq \mathrm{K}^t(x \mid y) + \mathrm{K}^t(y) + O(1) - \mathrm{K}^{p^{(4)}(t)}(y) + \log p^{(4)}(t) \\ &= \mathrm{K}^t(x \mid y) + \mathrm{cd}^{t,p^{(4)}(t)}(y) + \log p^{(4)}(t) + O(1).\end{aligned}$$

On the other hand, we have

$$\begin{aligned}\widetilde{\mathrm{K}}(xy, 1^{p(t)}) - \widetilde{\mathrm{K}}(y, 1^{p^{(3)}(t)}) &\geq \mathrm{K}^{p^{(2)}(t)}(xy) - \log p^{(2)}(t) - \mathrm{K}^{p^{(3)}(t)}(y) \\ &\geq \mathrm{K}^{p^{(3)}(t)}(x \mid y) - 2\log p^{(3)}(t),\end{aligned}$$

where the last inequality follows from SoI.

We now set $\tau(n, m, t) := p^{(4)}(t+n+m)$ so that $p^{(4)}(t) \leq \tau(|x|, |y|, t)$ for any $x, y \in \{0,1\}^*$ and $t \in \mathbb{N}$. Define an algorithm B so that $B(x, y, 1^t) := \widetilde{\mathrm{K}}(xy, 1^{p(t')}) - \widetilde{\mathrm{K}}(y, 1^{p^{(3)}(t')}) - \frac{1}{2} \cdot \log \tau(n, m, t)$, where $t' := \max\{|x| + |y|, t\}$. Then, we have

$$\mathrm{K}^{\tau(n,m,t)}(x \mid y) - \log \tau(n, m, t) \leq B(x, y, 1^t) \leq \mathrm{K}^t(x \mid y) + \mathrm{cd}^{t,\tau(n,m,t)}(y)$$

for $n := |x|$ and $m := |y|$.

In order to show that $\mathrm{Gap}_\tau\mathrm{MINcKT} \in \mathsf{P}$, consider an algorithm M such that M accepts an input $(x, y, 1^t, 1^s)$ if and only if $B(x, y, 1^t) \leq s$. It is easy to see that M accepts every Yes instance and rejects every No instance of $\mathrm{Gap}_\tau\mathrm{MINcKT}$. ◀

▶ **Corollary 6.3.** *If* DistNP ⊆ AvgP*, then* $\mathrm{Gap}_\tau\mathrm{MINcKT} \in \mathsf{P}$ *for some polynomial* τ*.*

Proof. This immediately follows from Proposition 6.2, Theorem 1.2, , and Lemma 2.1 ◀

As a consequence, Heuristica can be excluded if GapMINcKT is NP-hard under randomized reductions.

Proof of Theorem 1.5. Assume that DistNP ⊆ AvgP. By Corollary 6.3, there exists a polynomial τ such that $\mathrm{Gap}_\tau\mathrm{MINcKT} \in \mathsf{P}$. By the NP-hardness assumption, we obtain NP ⊆ BPP = P, where the last equality follows from the theorem of Buhrman et al. [18]. ◀

▶ Remark 6.4 (On the optimality of the additive error). Generalizing Corollary 6.3, it holds that $\mathsf{Dist}\Sigma_2^{\mathsf{p}} \subseteq \mathsf{AvgP}$ implies that $\mathrm{GapMINcKT}^{\mathsf{NP}} \in \mathsf{P}$, where $\mathrm{GapMINcKT}^{\mathsf{NP}}$ is an NP-oracle version of GapMINcKT which asks to approximate $\mathrm{K}^{t,\mathrm{SAT}}(x \mid y)$ within an additive error $\mathrm{cd}^{t,p(t)}(y) + O(\log t)$. Below, we informally argue that a relativizing proof technique is unlikely to improve the error term $\mathrm{cd}^{t,p(t)}(y)$ of $\mathrm{GapMINcKT}^{\mathsf{NP}}$: In [34], NP-hardness of $\mathrm{MINcKT}^{\mathsf{NP}}$ was proved. The NP-hardness reduction in fact shows that for every $L \in \mathsf{NP}$, an instance y for L can be reduced to the problem of approximating $\mathrm{K}^{t,\mathsf{NP}}(x \mid y)$ up to an additive error Δ in time $2^{O(\Delta + \log|y|)}$. In particular, applying this reduction to $\mathrm{GapMINcKT}^{\mathsf{NP}}$, L can be solved in time $2^{O(\mathrm{cd}^{t,p(t)}(y) + \log|y| + \log t)}$ on input $(y, 1^t)$ if $\mathrm{GapMINcKT}^{\mathsf{NP}} \in \mathsf{P}$. This induces a universal heuristic scheme for L, which yields an algorithm that solves L in time $2^{O(n/\log n)}$ (see Section 7). Now, if $\mathrm{K}^{t,\mathrm{SAT}}(x \mid y)$ could be approximated with an additive error $o(\mathrm{cd}^{t,p(t)}(y))$ under the assumption that $\mathsf{Dist}\Sigma_2^{\mathsf{p}} \subseteq \mathsf{AvgP}$, then we would obtain an improved algorithm that solves every language $L \in \mathsf{NP}$ in time $2^{o(n/\log n)}$, which contradicts the relativization barrier of [36].

6.2 NP-Hardness of Sublinear-Time-Bounded Conditional Kolmogorov Complexity

We now show that $\mathrm{Gap}_\tau\mathrm{MINcKT}$ is NP-hard under randomized reductions if $\tau(|x|,|y|,t)$ is sublinear in the length of y. In fact, we prove that it is NP-hard to approximate $\mathrm{K}^t(x \mid y)$ to within a factor of $|x|^{1/(\log\log|x|)^{O(1)}}$. We define a version of GapMINcKT that incorporates a multiplicative factor as follows.

▶ **Definition 6.5.** *For a polynomial $\tau\colon \mathbb{N}^3 \to \mathbb{N}$ and a function $\sigma\colon \mathbb{N} \to \mathbb{N}$, we define the promise problem $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ to be $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ such that*

$$\Pi_{\mathrm{YES}} := \left\{(x,y,1^t,1^s) \;\middle|\; \mathrm{K}^t(x \mid y) \le s - \mathrm{cd}^{t,\tau(|x|,|y|,t)}(y)\right\},$$
$$\Pi_{\mathrm{NO}} := \left\{(x,y,1^t,1^s) \;\middle|\; \mathrm{K}^{\tau(|x|,|y|,t)}(x \mid y) > \sigma(|x|)\cdot s + \log\tau(|x|,|y|,t)\right\}.$$

Note that $\mathrm{Gap}_{\tau,1}\mathrm{MINcKT} = \mathrm{Gap}_\tau\mathrm{MINcKT}$ is trivially reducible to $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ for every $\sigma \ge 1$. We show that $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ is NP-hard for $\sigma(|x|) = |x|^{1/(\log\log|x|)^{O(1)}}$.

▶ **Theorem 6.6.** *Let $c > 1$ be an arbitrary constant and $\tau\colon \mathbb{N}^3 \to \mathbb{N}$ be a function such that $\tau(n,m,t) \le n^c \cdot m^{1-1/c} \cdot t^c$ for all large n, m, and $t \in \mathbb{N}$. Then, $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ is NP-hard under one-query randomized polynomial-time reductions for some function $\sigma\colon \mathbb{N} \to \mathbb{N}$ such that $\sigma(n) = n^{1/(\log\log n)^{O(1)}}$.*

6.2.1 Secret Sharing Scheme

We review the notion of secret sharing scheme below.

▶ **Definition 6.7** (Access Structure). *An* access structure *$\mathcal{A} \subseteq 2^{[n]}$ is a "monotone" collection of subsets of $[n]$; that is, for every $T \supseteq S \in \mathcal{A}$, we have $T \in \mathcal{A}$. The minimum weight of $\mathcal{A}$ is defined to be $w(\mathcal{A}) := \min\{|T| \mid T \in \mathcal{A}\}$.*

We will prove that there is a generic reduction from the problem of estimating the weight of access structures $\mathcal{A}$ to GapMINcKT if there exists an efficient secret sharing scheme for $\mathcal{A}$.

▶ **Definition 6.8** (Secret Sharing [12]). *A* secret sharing scheme *for $\mathcal{A}$ is a pair* (Share, Rec) *of a randomized algorithm* Share *and a deterministic algorithm* Rec *with the following properties for every $\ell \in \mathbb{N}$:*

1. Correctness: *For every $T \in \mathcal{A}$ and for every string $x \in \{0,1\}^\ell$, any output of* $\mathrm{Share}(x)$ *is a sequence $(y_1,\dots,y_n)$ of n strings that satisfies*

$$\mathrm{Rec}(y_T) = x,$$

where $y_T := \{(i, y_i) \mid i \in T\}$.

2. Privacy: *For every $T \notin \mathcal{A}$ and for every random variable X on $\{0,1\}^\ell$, the random variables X and $\mathrm{Share}(X)_T$ are statistically independent.*

We observe that the privacy condition can be stated in terms of Kolmogorov complexity.[15]

[15] We mention in passing that Kolmogorov complexity-theoretic versions of privacy conditions are studied in, e.g., [10, 48].

▶ **Lemma 6.9.** *Let* (Share, Rec) *be a secret sharing scheme for an access structure* $\mathcal{A}$ *over* $[n]$*. Then, for every* ℓ *and* $k \in \mathbb{N}$*, it holds that*

$$\Pr\left[\min_{T \notin \mathcal{A}} \mathrm{K}(X \mid \mathrm{Share}(X)_T) \geq \ell - n - k\right] \geq 1 - 2^{-k},$$

where X *is the uniform distribution over* $\{0,1\}^\ell$ *and the probability is taken over* X *as well as the internal randomness of* Share*.*

Proof. Fix an arbitrary subset $T \notin \mathcal{A}$ of $[n]$. Let $y_T \in \mathrm{supp}(\mathrm{Share}(X)_T)$ be an outcome of $\mathrm{Share}(X)_T$. By Fact 3.2, we obtain

$$\Pr\left[\mathrm{K}(X \mid y_T) < \ell - n - k\right] \leq 2^{-n-k}.$$

By the privacy of the secret sharing scheme, the random variables X and $\mathrm{Share}(X)_T$ are statistically independent. By averaging the inequality above over all $y_T \in \mathrm{supp}(\mathrm{Share}(X)_T)$, we obtain

$$\Pr\left[\mathrm{K}(X \mid \mathrm{Share}(X)_T) < \ell - n - k\right] \leq 2^{-n-k}.$$

Finally, we take the union bound over all subsets $T \notin \mathcal{A}$ and conclude that

$$\Pr\left[\exists T \notin \mathcal{A},\ \mathrm{K}(X \mid \mathrm{Share}(X)_T) < \ell - n - k\right] \leq 2^{-k}. \qquad ◀$$

The “efficiency” of a secret sharing scheme is defined as follows.

▶ **Definition 6.10.** *A family* $\mathcal{A} = \{\mathcal{A}_\varphi\}_{\varphi \in \{0,1\}^*}$ *of access structures is said to* admit efficient secret sharing schemes *if there exists a pair* (Share, Rec) *of a randomized polynomial-time algorithm* Share *and a deterministic polynomial-time algorithm* Rec *such that for every* $\varphi \in \{0,1\}^*$*, the pair* $(\mathrm{Share}(\varphi, \text{-}), \mathrm{Rec}(\varphi, \text{-}))$ *is a secret sharing scheme for the access structure* $\mathcal{A}_\varphi$*.*

Benaloh and Leichter [13] showed that access structures represented by monotone formulas admit efficient secret sharing schemes.

▶ **Lemma 6.11** ([13])**.** *Let* $\mathcal{A} := \{\mathcal{A}_\varphi\}_{\varphi \in \{0,1\}^*}$ *be the family of access structures* $\mathcal{A}_\varphi := \{T \subseteq [n] \mid \varphi(\chi_T) = 1\}$*, where* φ *is a monotone formula on* n *variables and* $\chi_T \in \{0,1\}^n$ *denotes the characteristic vector of* $T \subseteq [n]$*. Then,* $\mathcal{A}$ *admits efficient secret sharing schemes.*

6.2.2 Minimum Monotone Satisfying Assignment

In order to prove Theorem 6.6, we reduce the Minimum Monotone Satisfying Assignment (MMSA) problem to GapMINcKT.

▶ **Definition 6.12** (Minimum Monotone Satisfying Assignment; MMSA)**.** *For a monotone formula* φ *on* n *variables, the* weight *of an assignment* $\alpha \in \{0,1\}^n$ *is defined to be* $\sum_{i=1}^n \alpha_i$*. Let* $\mathrm{MMSA}(\varphi)$ *denote the minimum weight of* $\alpha \in \{0,1\}^n$ *such that* $\varphi(\alpha) = 1$*.*

Observe that $\mathrm{MMSA}(\varphi) = w(\mathcal{A}_\varphi)$ for the family $\mathcal{A}$ of access structures of Lemma 6.11. It is known that MMSA is NP-hard to approximate:

▶ **Lemma 6.13** ([22, 21])**.** *For some function* $g(n) = n^{1/(\log\log n)^{O(1)}}$*, it is* NP*-hard to solve the promise problem* $\mathrm{Gap}_g\mathrm{MMSA} = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ *defined as follows:*

$$\Pi_{\mathrm{YES}} := \{(\varphi, s) \mid \mathrm{MMSA}(\varphi) \leq s\},$$
$$\Pi_{\mathrm{NO}} := \{(\varphi, s) \mid \mathrm{MMSA}(\varphi) > s \cdot g(|\varphi|)\},$$

where $|\varphi|$ *denotes the length of the binary string that represents* φ*.*

Proof Sketch. Dinur and Safra [22] presented a polynomial-time reduction from any constraint satisfaction problem (CSP) to $\mathrm{Gap}_g\mathrm{MMSA}$ with the following property: Let ψ be an instance of the CSP. Let (φ, s) be the output of the reduction. Let D be the arity of ψ. If ψ is satisfiable, then $\mathrm{MMSA}(\varphi) \le s$. If there is no assignment satisfying a $1/\bigl(2(2Dg)^D\bigr)$ fraction of the constraints of ψ, then $\mathrm{MMSA}(\varphi) > s \cdot g$, where g is an arbitrary parameter. Dinur et al. [21] showed that it is NP-hard to decide whether a given CSP with n variables and of arity $D = (\log\log n)^{O(1)}$ is satisfiable or every assignment satisfies at most a $1/\mathsf{poly}(n)$ fraction of the constraints. Combining these two hardness results, we conclude that $\mathrm{Gap}_g\mathrm{MMSA}$ is NP-hard, where $g \ge \mathsf{poly}(n)^{1/D} \cdot D^{-D} \ge n^{1/(\log\log n)^{O(1)}}$. ◀

6.2.3 A Generic Connection from Secret Sharing Schemes

We now formally state a generic reduction that reduces the problem of estimating the weight of access structures to GapMINcKT.

▶ **Theorem 6.14.** *Let $\mathcal{A} = \{\mathcal{A}_\varphi\}_{\varphi \in \{0,1\}^*}$ be a family of access structures that admits efficient secret sharing schemes. Let $\tau \colon \mathbb{N}^3 \to \mathbb{N}$ be a function such that $t \le \tau(n, m, t) \le n^c \cdot m^{1-1/c} \cdot t^c$ for some constant $c > 1$. Then, there exists a randomized polynomial-time algorithm R that takes $\varphi \in \{0,1\}^*$ as input and outputs $x, y \in \{0,1\}^*$ and $t, \rho \in \mathbb{N}$ such that*

$$\frac{\rho}{8c} \cdot w(\mathcal{A}_\varphi) + \log t' \le \mathrm{K}^{t'}(x \mid y) \le \mathrm{K}^t(x \mid y) \le \rho \cdot w(\mathcal{A}_\varphi) - \mathrm{cd}^t(y)$$

holds with probability at least $1 - o(1)$ over the internal randomness of R, where $t' := \tau(|x|, |y|, t)$.

It is easy to observe that this reduction implies NP-hardness of GapMINcKT.

Proof of Theorem 6.6. Let $\mathcal{A} = \{\mathcal{A}_\varphi\}_\varphi$ be the family of access structures for monotone formulas given in Lemma 6.11. Let R be the reduction of Theorem 6.14. Consider a reduction R' that reduces an instance (φ, s) of $\mathrm{Gap}_g\mathrm{MMSA}$ to an instance $(x, y, 1^t, 1^{s'})$ of $\mathrm{Gap}_\tau\mathrm{MINcKT}$ such that $(x, y, t, \rho) := R(\varphi)$ and $s' := s \cdot \rho$.

Let $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}}) := \mathrm{Gap}_g\mathrm{MMSA}$. We claim below that the reduction R' reduces $\mathrm{Gap}_g\mathrm{MMSA}$ to $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ for some σ, which implies NP-hardness of $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ by Lemma 6.13. If $(\varphi, s) \in \Pi_{\mathrm{YES}}$, then $w(\mathcal{A}_\varphi) = \mathrm{MMSA}(\varphi) \le s$. By the properties of R, with high probability, it holds that $\mathrm{K}^t(x \mid y) \le \rho \cdot w(\mathcal{A}_\varphi) - \mathrm{cd}^t(y) \le s' - \mathrm{cd}^{t,t'}(y)$, which implies that $(x, y, 1^t, 1^{s'})$ is a YES instance of $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$. If $(\varphi, s) \in \Pi_{\mathrm{NO}}$, then $w(\mathcal{A}_\varphi) = \mathrm{MMSA}(\varphi) > g(|\varphi|) \cdot s$. Also by the properties of R, with high probability, it holds that $\mathrm{K}^{t'}(x \mid y) \ge \frac{\rho}{8c} \cdot w(\mathcal{A}_\varphi) + \log t' > \frac{g(|\varphi|)}{8c} \cdot s' + \log t'$, which implies that $(x, y, 1^t, 1^{s'})$ is a NO instance of $\mathrm{Gap}_{\tau,\sigma}\mathrm{MINcKT}$ for some $\sigma(|x|) := \frac{g(|\varphi|)}{8c} \ge |\varphi|^{1/(\log\log|\varphi|)^{O(1)}} \ge |x|^{1/(\log\log|x|)^{O(1)}}$, where the last inequality follows from the fact that $|\varphi| \ge |x|^{\Omega(1)}$. ◀

It remains to prove Theorem 6.14. To show that the additive error $\mathrm{cd}^t(y)$ is relatively small, we use the fact that for D independent random samples $y^1, \ldots, y^D$ from a distribution $\mathcal{D}$ samplable by polynomial-size circuits, the amortized time-bounded Kolmogorov complexity of $y^1, \ldots, y^D$ approaches the entropy of $\mathcal{D}$ asymptotically.

▶ **Lemma 6.15** ([6, 5])**.** *Let $\mathcal{D} = \{\mathcal{D}_x\}_{x \in \{0,1\}^*}$ be a family of distributions sampled by circuits of size $\mathsf{poly}(|x|)$. Then, there exist a polynomial p and a constant $\delta > 0$ such that for every $x \in \{0,1\}^*$ and for every $D \ge p(|x|)$,*

$$\frac{1}{D}\cdot \mathrm{K}^t(y^1,\ldots,y^D) \le \mathrm{H}(\mathcal{D}_x) + \frac{1}{2}\cdot D^{-\delta},$$
$$\frac{1}{D}\cdot \mathrm{K}(y^1,\ldots,y^D) \ge \mathrm{H}(\mathcal{D}_x) - \frac{1}{2}\cdot D^{-\delta}$$

holds with probability at least $1-2^{-|x|}$ *over a random choice of* D *independent samples* $y^1,\ldots,y^D$ *from* $\mathcal{D}_x$*, where* $t := p(|x|)$*.*

This lemma immediately implies that the computational depth of independent samples from $\mathcal{D}_x$ is small.

▶ **Corollary 6.16.** *Under the same assumptions as Lemma 6.15, with probability at least* $1-2^{-|x|}$ *over a random choice of* D *independent samples* $y^1,\ldots,y^D$ *from* $\mathcal{D}_x$*, it holds that*

$$\mathrm{cd}^t(y^1,\ldots,y^D) \le D^{1-\delta}.$$

Proof. It follows from Lemma 6.15 that

$$\mathrm{cd}^t(y^1,\ldots,y^D) = \mathrm{K}^t(y^1,\ldots,y^D) - \mathrm{K}(y^1,\ldots,y^D) \le D^{1-\delta}. \qquad ◀$$

We are now ready to prove Theorem 6.14.

Proof of Theorem 6.14. Fix an input $\varphi \in \{0,1\}^*$. Let n be the number of parties in the access structure $\mathcal{A}_\varphi$. Let $\lambda = O(\log|\varphi|)$, t, and ℓ be parameters chosen later.

We first define a family $\mathcal{D} = \{\mathcal{D}_\varphi\}_{\varphi\in\{0,1\}^*}$ of distributions by using the following sampling procedure: Choose $x \sim \{0,1\}^\ell$ uniformly at random. Let $(s_1,\ldots,s_n) := \mathrm{Share}(\varphi,x)$ and let m be the length of each share; i.e., $m := |s_i|$. Pick $k_1,\ldots,k_n \sim \{0,1\}^\lambda$ randomly. We define a string $y \in \{0,1\}^{2^\lambda\cdot m}$, which we identify with a function $y\colon \{0,1\}^\lambda \to \{0,1\}^m$: For every $q \in \{0,1\}^\lambda$, let $y(q) := s_i$ if $q = k_i$ for some $i \in [n]$ and let $y(q) := 0^m$ otherwise. Note that y is not well defined if $(k_i)_{i\in[n]}$ is not pairwise distinct; however, we will show that y is well defined with high probability. The output of the sampling procedure is defined as (x,y,k,s), where $k := (k_1,\ldots,k_n)$ and $s := (s_1,\ldots,s_n)$.

The algorithm R operates as follows: For a given input $\varphi \in \{0,1\}^*$, pick D independent samples $(x^1,y^1,k^1,s^1),\ldots,(x^D,y^D,k^D,s^D)$ from $\mathcal{D}_\varphi$. Let $x := (x^1,\ldots,x^D)$, $y := (y^1,\ldots,y^D)$, and $\rho := 2\lambda D$. The output of R is defined to be (x,y,t,ρ).

Below, we prove the correctness of R using a sequence of claims. Let $k := (k_i^d \mid i\in[n], d\in[D])$ and let $s := (s_i^d \mid i\in[n], d\in[D])$. We first observe that k is Kolmogorov-random with high probability:

▷ Claim 6.17. With probability at least $1-o(1)$, it holds that

$$\mathrm{K}(k \mid s) \ge nD\lambda - \log n. \tag{16}$$

Proof. Since $k = (k_i^d)_{i\in[n],d\in[D]}$ is uniformly distributed over $(\{0,1\}^\lambda)^{nD}$ and independent of s, the claim follows from Fact 3.2. ◁

▷ Claim 6.18. If Equation (16) holds, then $k = (k_i^d)_{i\in[n],d\in[D]}$ is pairwise distinct and hence y is well defined.

Proof. If there exists $(i,d) \ne (i',d')$ such that $k_i^d = k_{i'}^{d'}$, then k can be described by a program of size $nD\lambda - \lambda + O(\log nD)$. Choosing a large enough $\lambda = O(\log|\varphi|)$, this implies that $\mathrm{K}(k) \le nD\lambda - \lambda + O(\log nD) < nD\lambda - \log n$, which contradicts Equation (16). ◁

Below, we assume that Equation (16) holds. We prove an upper bound of $\mathrm{K}^t(x \mid y)$.

▷ Claim 6.19. $\mathrm{K}^t(x \mid y) \le \rho \cdot w(\mathcal{A}_\varphi) - \mathrm{cd}^t(y)$ holds with probability at least $1 - o(1)$.

Proof. Let $T \in \mathcal{A}_\varphi$ be a minimum authorized set of parties such that $|T| = w(\mathcal{A}_\varphi)$. We present an oracle program M^y that takes an index d and outputs $x^d \in \{0,1\}^\ell$ as follows: M^y takes $d \in [D]$ as input and T, $\left\{k_i^d \;\middle|\; i \in T, d \in [D]\right\}$ and φ as hard-wired input, computes $\left\{s_i^d \;\middle|\; i \in T\right\} = \left\{y^d(k_i) \;\middle|\; i \in T\right\}$ by making queries to the oracle y, and outputs $\mathrm{Rec}(\varphi, s_T^d)$, which is equal to x^d by the correctness of a secret sharing scheme and the assumption that $T \in \mathcal{A}_\varphi$. The size of the oracle machine M is at most

$$O(|T| \cdot \log n) + |T| \cdot D \cdot \lambda + |\varphi| + O(\log D) \le 2\lambda D \cdot w(\mathcal{A}_\varphi) - D^{1-\delta},$$

where $\delta > 0$ is the constant from Corollary 6.16 and this inequality follows by choosing sufficiently large $D \ge \mathsf{poly}(|\varphi|)$. Applying Corollary 6.16 to the distribution of y, we obtain $\mathrm{cd}^t(y) \le D^{1-\delta}$ with probability at least $1-o(1)$; thus, $|M| \le \rho \cdot w(\mathcal{A}_\varphi) - \mathrm{cd}^t(y)$. The running time of M is at most $\mathsf{poly}(|\varphi|)$, which is independent of λ. We choose t so that this running time is at most t. ◁

The remainder of the proof is devoted to proving the lower bound.

▷ Claim 6.20. Let $\theta := \frac{\rho}{8c} \cdot w(\mathcal{A}_\varphi) + \log t'$. Then, $\mathrm{K}^{t'}(x \mid y) \ge \theta$ with probability at least $1 - o(1)$.

We first clarify the condition of random variables under which the claim holds. Let $s_{[n]}^{[D]\setminus\{d\}} := (s_i^{d'} \mid i \in [n], d' \in [D] \setminus \{d\})$.

▷ Claim 6.21. With probability at least $1 - o(1)$, it holds that for every unauthorized set $T \notin \mathcal{A}_\varphi$ of parties and for every $d \in [D]$,

$$\mathrm{K}\left(x^d \;\middle|\; s_T^d, s_{[n]}^{[D]\setminus\{d\}}, k\right) \ge \ell - n - 2\log D. \tag{17}$$

Proof. Fix $d \in [D]$. Since the random variable x^d is independent of $s_{[n]}^{[D]\setminus\{d\}}$ and k, by Lemma 6.9,

$$\min_{T \notin \mathcal{A}_\varphi} \mathrm{K}\left(x^d \;\middle|\; s_T^d, s_{[n]}^{[D]\setminus\{d\}}, k\right) \ge \ell - n - 2\log D$$

holds with probability at least $1 - \frac{1}{D^2}$. The claim follows by taking a union bound over all $d \in [D]$. ◁

In what follows, we assume Equations (16) and (17) and prove $\mathrm{K}^{t'}(x \mid y) \ge \theta$, which will complete the proof of Claim 6.20. Assume, by way of contradiction, that $\mathrm{K}^{t'}(x \mid y) < \theta$. Let $M \in \{0,1\}^*$ be the description of an oracle program such that $|M| \le \theta + O(\log \ell)$ and $M^y(d)$ outputs $x^d \in \{0,1\}^\ell$ in time $t'' := t'\ell$ for every $d \in [D]$; the machine M can be constructed from a program that witnesses $\mathrm{K}^{t'}(x \mid y) < \theta$. For each $d \in [D]$, let $T(M,d)$ be the set of indices $i \in [n]$ such that some bit of $y^d(k_i^d) \in \{0,1\}^m$ is queried during the computation of $M^y(d)$.

▷ Claim 6.22. Let M be an oracle program that runs in time t'' and let $\alpha := \sum_{d=1}^{D} |T(M,d)|$. Then,

$$\mathrm{K}(k \mid s) \le |M| + (nD - \alpha) \cdot \lambda + \alpha \cdot (\log t'' + \log nD) + O(\log nD).$$

Proof. By the definition of $T(M,d)$, for every $i \in T(M,d)$, there exists a time step $t_{i,d} \in [t'']$ such that M^y makes a query k_i^d to the oracle y on input $d \in [D]$. To describe k given s, consider the following program M': M' takes

$$\{(i,d,t_{i,d}) \in [n] \times [D] \times [t''] \mid i \in T(M,d)\},$$
$$\left\{k_i^d \in \{0,1\}^\lambda \;\middle|\; (i,d) \in [n] \times [D], i \notin T(M,d)\right\},$$

and M as input and simulates M^y on input d for every $d \in [D]$. At the time step $t_{i,d}$ of the simulation of $M^y(d)$ for some $i \in T(M,d)$, M' reads the query k_i^d that $M^y(d)$ makes to the oracle y^d and answers the query with s_i^d. M' continues the simulation until the time step t'', at which point M' knows k_i^d for every $i \in T(M,d)$. Finally, M' outputs k. The input of M' can be encoded as a string of length $\alpha \cdot (\log t'' + \log nD) + (nD - \alpha) \cdot \lambda + |M| + O(\log nD)$. ◁

It follows from Claim 6.22 and Equation (16) that

$$nD\lambda - \log n \le |M| + (nD - \alpha) \cdot \lambda + \alpha \cdot (\log t'' + \log nD) + O(\log nD),$$

which can be simplified to

$$(\lambda - \log t'' - \log nD) \cdot \alpha \le |M| + O(\log nD). \tag{18}$$

Since $t''/\ell = t' = \tau(|x|,|y|,t) \le |x|^c \cdot |y|^{1-1/c} \cdot t^c \le 2^{(1-1/c)\cdot\lambda} \cdot (D \cdot |\varphi| \cdot \ell \cdot m)^{O(c)}$, we may choose $\lambda = O(\log|\varphi|)$ large enough so that $t'' \le 2^{(1-1/2c)\cdot\lambda - \log nD}$. Then, by Equation (18), we obtain

$$\alpha \le \frac{2c}{\lambda} \cdot (|M| + O(\log nD)) < \frac{4c}{\lambda} \cdot (\theta - \log t') = \frac{\rho}{2\lambda} \cdot w(\mathcal{A}_\varphi) = D \cdot w(\mathcal{A}_\varphi),$$

where the second inequality follows from $|M| + O(\log nD) \le \theta + O(\log nD\ell t') < 2\theta$.[16] It follows that there exists $d \in [D]$ such that $|T(M,d)| \le \alpha/D < w(\mathcal{A}_\varphi)$, which implies that

$$T(M,d) \notin \mathcal{A}_\varphi.$$

By Equation (17), we obtain

$$\mathrm{K}\left(x^d \;\middle|\; s^d_{T(M,d)}, s^{[D]\setminus\{d\}}_{[n]}, k\right) \ge \ell - n - 2\log D.$$

However, this contradicts the following claim.

▷ Claim 6.23. Let M be an oracle program such that $M^y(d)$ outputs x^d for every $d \in [D]$. Then, for every $d \in [D]$, it holds that

$$\mathrm{K}\left(x^d \;\middle|\; s^d_{T(M,d)}, s^{[D]\setminus\{d\}}_{[n]}, k\right) \le |M| + O(\log D).$$

Proof. Since $M^y(d)$ outputs x^d without making any query in $\left\{k_i^d \;\middle|\; i \in [n] \setminus T(M,d)\right\}$, we can define another oracle $z = (z^1, \dots, z^D)$ such that $z^{d'} := y^{d'}$ for every $d' \in [D] \setminus \{d\}$ and $z^d(k_i^d) := s_i^d$ for every $i \in T(M,d)$ and $z^d(k) := 0^m$ otherwise, so that there is no difference between y and z on inputs queried by $M^y(d)$; therefore, we obtain $x^d = M^y(d) = M^z(d)$. The claim follows by observing that the oracle z can be constructed from M, $d \in [D]$, $s^d_{T(M,d)}$, $s^{[D]\setminus\{d\}}_{[n]}$, and k. ◁

[16] We may assume without loss of generality that $w(\mathcal{A}_\varphi) \ge 1$; then, we have $\theta \ge \Omega(\rho) \ge \Omega(D)$, which can be assumed to be larger than logarithmic terms.

We conclude that $\ell - n - 2\log D \le |M| + O(\log D) \le O(\theta) \le O(n\lambda D)$, which is a contradiction by letting $\ell \gg O(n\lambda D)$. This completes the proof of Claim 6.20. ◀

▶ Remark 6.24. It is not hard to observe that our NP-hardness hardness reduction also proves the NP-hardness of McKTP [4] because the upper bound of Claim 6.19 also holds for $\mathrm{KT}(x \mid y)$.

7 A Simple Proof for Worst-Case to Average-Case Connections

In this section, using SoI, we present an alternative proof of the connection from the worst-case complexity of the polynomial hierarchy PH to the average-case complexity of PH.

▶ **Theorem 7.1** ([35]). DistPH ⊆ AvgP *implies that* PH ⊆ $\mathsf{DTIME}(2^{O(n/\log n)})$.

One important component of the proof of Theorem 7.1 is the notion of universal heuristic scheme.

▶ **Definition 7.2** (Universal Heuristic Scheme [35]). *A* (strong) universal heuristic scheme *for a language L is a pair (S, C) of polynomial-time algorithms such that, for some polynomial p, for any $n \in \mathbb{N}$, any $t \ge p(n)$, and any $x \in \{0,1\}^n$,*
1. *if $\mathrm{cd}^{t,p(t)}(x) \le k$, then $C(x, 1^t, 1^k) = 1$, and*
2. *if $C(x, 1^t, 1^k) = 1$, then $S(x, 1^t, 1^{2^k}) = L(x)$.*

S and C are referred to as a solver *and a* checker*, respectively.*

A strong universal heuristic scheme enables the construction of an efficient algorithm:

▶ **Lemma 7.3** ([35]). *For every language L, if there exists a strong universal heuristic scheme for L, then $L \in$* $\mathsf{DTIME}(2^{O(n/\log n)})$.

Our goal is to construct a strong universal heuristic scheme for every language in PH. To this end, we introduce a weaker version of a universal heuristic scheme:

▶ **Definition 7.4.** *A* weak universal heuristic scheme *for a language L is a polynomial-time algorithm S such that, for some polynomial p, for any $n \in \mathbb{N}$, any $t \ge p(n)$, and any $x \in \{0,1\}^n$, if $\mathrm{cd}^{t,p(t)}(x) \le k$, then $S(x, 1^t, 1^{2^k}) = L(x)$.*

We observe that weak and strong universal heuristic schemes are in fact equivalent in Heuristica.

▶ **Lemma 7.5.** *If* GapMINKT ∈ P*, the following are equivalent for every language L.*
1. *There exists a strong universal heuristic scheme for L.*
2. *There exists a weak universal heuristic scheme for L.*

Under the assumption that DistNP ⊆ AvgP, [35] showed the equivalence between the existence of a strong universal heuristic scheme for L and $\{L\} \times \mathrm{PSAMP} \subseteq \mathsf{Avg_P P}$, where $\mathsf{Avg_P P}$ denotes the class of distributional problems solvable by algorithms whose running time is bounded above by some polynomial-time-computable average-case polynomial-time bound. Lemma 7.5 adds a new equivalent statement to this.

Proof of Lemma 7.5.

strong → weak. For a strong universal heuristic scheme (S, C), the solver S satisfies the definition of a weak universal heuristic scheme.

weak → strong. The idea of constructing a checker is to estimate the time-bounded computational depth of an input by using an algorithm for GapMINKT.

Let S be a weak universal heuristic scheme for L and let p be the polynomial in Definition 7.4. Let $\widetilde{\mathrm{K}}$ be the polynomial-time algorithm of Fact 3.4 such that for every $x \in \{0,1\}^*$ and every $t \geq |x|$,[17]

$$\mathrm{K}^{p(t)}(x) - \log p(t) \leq \widetilde{\mathrm{K}}(x, 1^t) \leq \mathrm{K}^t(x).$$

Observe that

$$\mathrm{cd}^{p(t),p^{(2)}(t)}(x) - \log p(t) \leq \widetilde{\mathrm{K}}(x, 1^t) - \widetilde{\mathrm{K}}(x, 1^{p^{(2)}(t)}) \leq \mathrm{cd}^{t,p^{(3)}(t)}(x) + \log p^{(3)}(t). \tag{19}$$

We define a checker C as follows: $C(x, 1^t, 1^k) = 1$ if and only if $\widetilde{\mathrm{K}}(x, 1^t) - \widetilde{\mathrm{K}}(x, 1^{p^{(2)}(t)}) \leq k + \log p^{(3)}(t)$. We define a solver S' so that $S'(x, 1^t, 1^{2^k}) := S'(x, 1^{p(t)}, 1^{2^{k'}})$, where $k' := k + \log p(t) + \log p^{(3)}(t)$.

Below, we claim that (S', C) is a strong universal heuristic scheme by showing that it satisfies the two properties of Definition 7.2.

1. If $\mathrm{cd}^{t,p^{(3)}(t)}(x) \leq k$, then by the upper bound of Equation (19), we have $C(x, 1^t, 1^k) = 1$.
2. If $C(x, 1^t, 1^k) = 1$, then by the definition of C and by the lower bound of Equation (19), we obtain $\mathrm{cd}^{p(t),p^{(2)}(t)}(x) \leq k + \log p(t) + \log p^{(3)}(t) = k'$. It follows from the property of the weak heuristic scheme S that $S'(x, 1^t, 1^{2^k}) = S(x, 1^{p(t)}, 1^{2^{k'}}) = L(x)$. ◀

The following lemma shows that any string that can be efficiently compressed with some PH oracle can also be compressed without the oracle if DistPH $\subseteq$ AvgP.

▶ **Lemma 7.6** ([33]). *Let A be an oracle. Assume that* $\mathsf{DistNP}^A \subseteq \mathsf{AvgP}$. *Then, there exists a polynomial p such that, for every $x \in \{0,1\}^*$ and every $t \geq |x|$,*

$$\mathrm{K}^{p(t)}(x) \leq \mathrm{K}^{t,A}(x) + \log p(t).$$

We now use SoI to construct a weak universal heuristic scheme for every language in PH.

▶ **Lemma 7.7.** *Let $k \in \mathbb{N}$. If $\mathsf{Dist}\Sigma^{\mathsf{p}}_{k+1} \subseteq \mathsf{AvgP}$, then for every language $L \in \Sigma^{\mathsf{p}}_k$, there exists a weak universal heuristic scheme for L.*

Proof. We prove this by induction on $k \in \mathbb{N}$. The base case ($k = 0$) is trivial because every language $L \in \Sigma^{\mathsf{p}}_0 = \mathsf{P}$ admits a weak universal heuristic scheme. Let $k \geq 1$. Let V be a language in Π^{p}_{k-1} such that $x \in L$ if and only if $V(x, y) = 1$ for some $y \in \{0,1\}^{\mathsf{poly}(|x|)}$. For every $x \in L$, let y_x be the lexicographically first string y such that $V(x, y) = 1$. The following claim is the key to the construction of a weak universal heuristic scheme.

▷ Claim 7.8. There exists a polynomial q such that for every $x \in L$ and every $t \geq |x|$,

$$\mathrm{K}^{q(t)}(y_x \mid x) \leq \mathrm{cd}^{t,q(t)}(x) + \log q(t).$$

Proof. By a standard search-to-decision reduction, y_x can be computed from x in polynomial time with oracle access to some oracle A in Σ^{p}_k; thus, it follows that

$$\mathrm{K}^{p^{(2)}(t)}(y_x, x) \leq \mathrm{K}^{p(t),A}(y_x, x) + \log p^{(2)}(t) \leq \mathrm{K}^t(x) + \log p^{(2)}(t) + O(1),$$

where the first inequality follows from Lemma 7.6.

[17] We may assume without loss of generality that the polynomial p in Fact 3.4 is the same polynomial with Definition 7.4.

We claim that $\mathrm{K}^{q(t)}(y_x \mid x) \le \mathrm{cd}^{t,q(t)}(x) + \log q(t)$ for some polynomial q. Note that SoI holds because of Theorem 1.2. Using SoI, we obtain

$$\begin{aligned}\mathrm{K}^{p^{(3)}(t)}(y_x \mid x) &\le \mathrm{K}^{p^{(2)}(t)}(y_x, x) - \mathrm{K}^{p^{(3)}(t)}(x) + \log p^{(3)}(t)\\ &\le \mathrm{K}^t(x) - \mathrm{K}^{p^{(3)}(t)}(x) + O(\log p^{(3)}(t))\\ &= \mathrm{cd}^{t,p^{(3)}(t)}(x) + O(\log p^{(3)}(t)),\end{aligned}$$

where the first inequality follows from SoI. The claim follows by letting $q(t) := p^{(3)}(t)^{O(1)}$. ◁

By the induction hypothesis, there exists a weak universal heuristic scheme S for V. Let p be the polynomial in Definition 7.4.

We now present a weak universal heuristic scheme S' for L: The algorithm S' takes $(x, 1^t, 1^{2^k})$ as input and computes the set Y of strings $y \in \{0,1\}^*$ such that there exists a program of length at most $k + \log q(t)$ that takes x as input and outputs y in time $q(t)$. Equivalently, we define

$$Y := \left\{ y \in \{0,1\}^* \;\middle|\; \mathrm{K}^{q(t)}(y \mid x) \le k + \log q(t). \right\}$$

Note that $|Y| \le 2^{k+\log q(t)+1}$ and Y can be computed in time $\mathsf{poly}(|x|, t, 2^k)$. The algorithm S' outputs 1 if and only if there exists a string $y \in Y$ such that $S((x,y), 1^{t'}, 1^{2^{k'}}) = 1$, where $t' = t^{O(1)}$ and $k' = O(k + \log t)$ are parameters chosen later. Clearly, S' is a polynomial-time algorithm.

We claim the correctness of S'. Let q' be a polynomial chosen later. Assume that $\mathrm{cd}^{t,q'(t)}(x) \le k$. We claim that for some parameter $t' = q(t)^{O(1)}$ and for every $y \in Y$, the $(t', p(t'))$-time-bounded computational depth of (x,y) is at most k', which will imply that the output of the weak universal heuristic scheme S is correct on input (x,y). For every $y \in Y$, we have

$$\begin{aligned}\mathrm{cd}^{t',p(t')}(x,y) &\le \mathrm{K}^{q(t)}(x) + \mathrm{K}^{q(t)}(y \mid x) - \mathrm{K}^{p(t')}(x,y) + O(1)\\ &\le \mathrm{cd}^{q(t),2p(t')}(x) + k + \log q(t) + O(1)\\ &\le 2k + \log q(t) + O(1) =: k',\end{aligned}$$

where the first inequality follows from the definition of time-bounded computational depth, the second inequality follows from the fact that $\mathrm{K}^{2p(t')}(x) \le \mathrm{K}^{p(t')}(x,y) + O(1)$ and $y \in Y$, and the third inequality follows from the assumption that $\mathrm{cd}^{q(t),2p(t')}(x) \le \mathrm{cd}^{t,q'(t)}(x) \le k$, where we define $q'(t) := 2p(t')$. By the correctness of the weak universal heuristic scheme S, we obtain $S((x,y), 1^{t'}, 1^{2^{k'}}) = V(x,y)$. If $x \in L$, Claim 7.8 implies that $y_x \in Y$; thus, we have $S((x,y_x), 1^{t'}, 1^{2^{k'}}) = V(x,y_x) = 1$, which implies that S' outputs 1. If $x \notin L$, then $V(x,y) = 0$ for every string y; thus, we obtain $S((x,y), 1^{t'}, 1^{2^{k'}}) = V(x,y) = 0$, which implies that S' outputs 0. ◀

Proof of Theorem 7.1. By Lemma 7.7, every language $L \in \mathsf{PH}$ admits a weak universal heuristic scheme. By Lemmas 2.1 and 7.5, the weak universal heuristic scheme can be converted into a strong universal heuristic scheme. Finally, using Lemma 7.3, we obtain $L \in \mathsf{DTIME}(2^{O(n/\log n)})$. ◀

8 What Is Implied by Symmetry of Information?

Toward giving an exact characterization of SoI, we investigate what SoI implies. We show that SoI is sandwiched between the existence of *errorless* heuristic schemes (denoted by AvgP) for MINKT and the existence of *error-prone* heuristic schemes (denoted by HeurP) for MINKT under the plausible assumption that E requires exponential-sized circuits. We start with the definition of error-prone and errorless heuristic schemes.

▶ **Definition 8.1.** *For a function* $t\colon \mathbb{N} \to \mathbb{N}$ *and a family* $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ *of distributions, an algorithm* A *is said to be an* error-prone heuristic scheme *for* $\mathrm{K}^t(\text{-})$ *with respect to* $\mathcal{D}$ *if for every* $n \in \mathbb{N}$ *and every* $\delta^{-1} \in \mathbb{N}$*,*

1. $A(x; n, \delta)$ *halts in time* $\mathsf{poly}(n/\delta)$ *for every* $x \in \operatorname{supp}(\mathcal{D}_n)$*.*
2. $\Pr_{x\sim\mathcal{D}_n}\left[A(x;n,\delta) \neq \mathrm{K}^{t(n)}(x)\right] \leq \delta$*.*

If A *satisfies the additional condition that*
resume $A(x;n,\delta) \in \left\{\mathrm{K}^{t(n)}(x), \perp\right\}$ *for every* $x \in \operatorname{supp}(\mathcal{D}_n)$*,*
then A *is said to be an* errorless heuristic scheme *for* $\mathrm{K}^t(\text{-})$ *with respect to* $\mathcal{D}$*. We write* $(\mathrm{K}^t(\text{-}), \mathcal{D}) \in \mathsf{AvgP}$ *and* $(\mathrm{K}^t(\text{-}), \mathcal{D}) \in \mathsf{HeurP}$ *if there exists an errorless heuristic scheme and an error-prone heuristic scheme for* $\mathrm{K}^t(\text{-})$ *with respect to* $\mathcal{D}$*, respectively.*

Definition 8.1 is different from the standard definition given in [16] in the following two respects.

1. The classes AvgP and HeurP are usually defined as the class of *decision* problems; here, we require that heuristic algorithms output an integer $\mathrm{K}^{t(n)}(x) \in \mathbb{N}$ on input $x \in \operatorname{supp}(\mathcal{D}_n)$.
2. The output $\mathrm{K}^{t(n)}(x)$ of the distributional problem $(\mathrm{K}^t(\text{-}), \mathcal{D})$ depends on the size parameter n.[18]

We mention that it is possible to state Theorem 8.2 below using only the standard definitions, though the statement becomes somewhat awkward; see Footnote 19. We now state the main result of this section.

▶ **Theorem 8.2.** *Assume that* $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{\epsilon n})$ *for some constant* $\epsilon > 0$*. In the following list, we have 1* $\iff$ *2* $\implies$ *3* $\implies$ *4 and 3* $\implies$ *5.*

1. $\mathrm{GapMINKT} \in \mathsf{P}$*.*
2. *For every* $\mathcal{D} \in \mathrm{PSAMP}$*, there exists a polynomial* t_0 *such that* $(\mathrm{K}^t(\text{-}), \mathcal{D}) \in \mathsf{AvgP}$ *for every polynomial* $t \geq t_0$*.*[19]
3. *SoI holds.*
4. *For every* $\mathcal{D} \in \mathrm{PSAMP}$*, there exists a polynomial* t_0 *such that* $(\mathrm{K}^t(\text{-}), \mathcal{D}) \in \mathsf{HeurP}$ *for every polynomial* $t \geq t_0$*.*
5. $\mathrm{Gap}_\tau\mathrm{MINKT} \in \mathsf{DTIME}(2^{O(n/\log n)})$ *for some function* $\tau(n,t) = 2^{O(n/\log n)}$*.*

Here, PSAMP denotes the class of families $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ of distributions such that there exists a randomized polynomial-time algorithm S such that the distribution induced by $S(1^n)$ is identical to $\mathcal{D}_n$.

Longpré and Watanabe [58] showed that SoI implies that $\mathrm{K}^t(\text{-})$ admits an error-prone heuristic scheme with respect to the uniform distribution. Theorem 8.2 extends their result to an arbitrary polynomial-time samplable distribution. To prove this, we need the following result from [58].

[18] The output does not depend on the size parameter n in the special case that $n = |x|$ for every $x \in \operatorname{supp}(\mathcal{D}_n)$.

[19] This statement can be equivalently stated as the statement that $(\mathrm{MINKT}, \mathcal{D}^t) \in \mathsf{AvgP}$, where MINKT is the language defined as $\left\{(x, 1^t, 1^s) \;\middle|\; \mathrm{K}^t(x) \leq s\right\}$ and $\mathcal{D}^t$ denotes the family $\{\mathcal{D}^t_n\}_{n\in\mathbb{N}}$ of distributions such that $\mathcal{D}^t_n$ is the distribution that picks $(x, 1^s) \sim \mathcal{D}_n$ and outputs a sample $(x, 1^{t(n)}, 1^s)$.

▶ **Lemma 8.3** ([58]). *If SoI holds, then there exist a polynomial p and a polynomial-time algorithm M such that for every $x \in \{0,1\}^*$ and every $t \geq p(|x|)$ such that $\mathrm{cd}^{t,p(t)}(x) \leq k$, on input $(x, 1^t, 1^{2^k})$, M outputs the lexicographically first program of length $\mathrm{K}^t(x)$ that outputs x in time t.*

Interestingly, this was proved even before the notion of computational depth was introduced by Antunes et al. [9]. Lemma 8.3 is reminiscent of a weak universal heuristic scheme for the search version of MINKT; however, it does not satisfy the definition of a weak universal heuristic scheme in that the parameter t cannot be chosen independently of the instance of MINKT. For completeness, we present the proof of Lemma 8.3.

Proof. Let $x \in \{0,1\}^*$, $t \in \mathbb{N}$, and $k \in \mathbb{N}$ be given inputs. Let $d_{x,t} \in \{0,1\}^*$ be the lexicographically first program of length $\mathrm{K}^t(x)$ that outputs x in time t. By SoI, for some polynomial p, we have

$$\begin{aligned}\mathrm{K}^{p(2t)}(d_{x,t} \mid x) &\leq \mathrm{K}^{2t}(d_{x,t}, x) - \mathrm{K}^{p(2t)}(x) + \log p(2t) \\ &\leq |d_{x,t}| + O(1) - \mathrm{K}^{p(2t)}(x) + \log p(2t) = \mathrm{cd}^{t,p(2t)}(x) + \log p(2t) + O(1),\end{aligned}$$

where the last inequality holds because $(d_{x,t}, x)$ can be computed from the description $d_{x,t}$ of the program. Consequently, there exists a polynomial q such that

$$\mathrm{K}^{q(t)}(d_{x,t} \mid x) \leq \mathrm{cd}^{t,q(t)}(x) + \log q(t). \tag{20}$$

We now describe the algorithm M: Given an input $(x, 1^t, 1^{2^k})$ such that $\mathrm{cd}^{t,q(t)}(x) \leq k$, the algorithm M computes the set Y of strings $d \in \{0,1\}^*$ such that $\mathrm{K}^{q(t)}(d \mid x) \leq k + \log q(t)$ and the program described by d outputs x in time t. The set Y can be computed in time $\mathsf{poly}(|x|, t, 2^k)$ by enumerating all the programs of length at most $k + \log q(t)$ that take x as input. Let $s := \min\{|d| \mid d \in Y\}$. The algorithm M outputs the lexicographically first string $d \in Y \cap \{0,1\}^s$.

To see the correctness of M, fix an input $(x, 1^t, 1^{2^k})$ such that $\mathrm{cd}^{t,q(t)}(x) \leq k$. We claim that M outputs $d_{x,t}$. Observe that $s \geq \mathrm{K}^t(x)$ by the definition of $\mathrm{K}^t(x)$. Moreover, it follows from Equation (20) that $d_{x,t} \in Y$; hence, we obtain $s \leq |d_{x,t}| = \mathrm{K}^t(x)$. Since $s = \mathrm{K}^t(x)$, the lexicographically first string $d \in Y \cap \{0,1\}^s$ is equal to $d_{x,t}$. ◀

Under a plausible derandomization hypothesis, Antunes and Fortnow [11] showed that if a string x is drawn from a polynomial-time samplable distribution, then the computational depth of x is small with high probability. The same conclusion holds under SoI.

▶ **Lemma 8.4** (see [35, Theorem 9.6 and Corollary 9.8]). *If SoI holds, then for every $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}} \in \mathrm{PSamp}$, there exists a polynomial t such that for every $n \in \mathbb{N}$,*

$$\Pr_{x \sim \mathcal{D}_n}\left[\mathrm{cd}^{t(n)}(x) > k\right] \leq 2^{-k + \log t(n)}.$$

▶ **Theorem 8.5.** *If SoI holds, then for every $\mathcal{D} \in \mathrm{PSamp}$, there exist a polynomial t and an error-prone heuristic scheme for $\mathrm{K}^t(\text{-})$ with respect to $\mathcal{D}$.*

Proof. Let M be the polynomial-time algorithm and p be the polynomial of Lemma 8.3. Let t_0 be the polynomial of Lemma 8.4.

We describe an error-prone heuristic scheme A: The algorithm A takes $(x; n, \delta)$ as input, where x is sampled from $\mathcal{D}_n$ and $\delta^{-1} \in \mathbb{N}$ is an error parameter. Let $k := \log(1/\delta) + \log t_0(n)$. The algorithm A outputs the length of the program computed by $M(x, 1^{t(n)}, 1^{2^k})$. Clearly, the algorithm A runs in time $\mathsf{poly}(n/\delta)$.

We claim the correctness of A, i.e., for every $n \in \mathbb{N}$ and every $\delta^{-1} \in \mathbb{N}$,

$$\Pr_{x \sim \mathcal{D}_n}\left[A(x; n, \delta) \neq \mathrm{K}^{t(n)}(x)\right] \leq \delta.$$

By the property of M, the algorithm M fails to output a program of size $\mathrm{K}^{t(n)}(x)$ on input $(x, 1^{t(n)}, 1^{2^k})$ only if $\mathrm{cd}^{t(n),p(t(n))}(x) > k$, which implies that $\mathrm{cd}^{t_0(n)}(x) \geq \mathrm{cd}^{t(n)}(x) > k$. By Lemma 8.4, this happens with probability at most $2^{-k+\log t_0(n)} \leq \delta$. We conclude that the probability that $A(x; n, \delta) \neq \mathrm{K}^{t(n)}(x)$ is at most δ, as desired. ◀

Next, we present an errorless heuristic version of Theorem 8.5.

▶ **Theorem 8.6.** *If SoI holds and* $\mathrm{GapMINKT} \in \mathsf{P}$*, then for every* $\mathcal{D} \in \mathrm{PSamp}$*, there exist a polynomial* t *and an errorless heuristic scheme for* $\mathrm{K}^t(\text{-})$ *with respect to* $\mathcal{D}$*.*

Proof. The proof is similar to that of Theorem 8.5, except that we use the algorithm for GapMINKT to translate the error-prone algorithm of Theorem 8.5 to an errorless algorithm in a way similar to Lemma 7.5.

Let p be the larger of the polynomial of Lemma 8.3 and the polynomial of Fact 3.4. Let C be the checker defined in the proof of Lemma 7.5, which satisfies the following properties: For every $x \in \{0,1\}^*$ and every $t \geq |x|$ and every $t \in \mathbb{N}$,

1. if $\mathrm{cd}^{t,p^{(3)}(t)}(x) \leq k$, then $C(x, 1^t, 1^k) = 1$, and
2. if $C(x, 1^t, 1^k) = 1$, then $\mathrm{cd}^{p(t),p^{(2)}(t)}(x) \leq k + \log p(t) + \log p^{(3)}(t)$.

We now describe an errorless heuristic scheme A: The algorithm A takes $(x; n, \delta)$ as input. Let $t := t(n)$ and $k := \log(1/\delta) + \log t$. The algorithm A outputs the special failure symbol $\perp$ if $C(x, 1^t, 1^k) = 0$. Otherwise, A outputs the length of the program computed by $M(x, 1^{p(t)}, 1^{2^{k'}})$, where we define $k' := k + \log p(t) + \log p^{(3)}(t)$. Clearly, the algorithm A runs in time $\mathsf{poly}(n/\delta)$.

To prove the correctness of A, we first show that A is errorless. For every $n \in \mathbb{N}$ and every $x \in \mathrm{supp}(\mathcal{D}_n)$, we claim

$$A(x; n, \delta) \in \{\mathrm{K}^t(x), \perp\}$$

by considering the following two cases: (1) If the checker C outputs 0, then by definition, A outputs $\perp$. (2) Otherwise, we have $C(x, 1^t, 1^k) = 1$, which implies that $\mathrm{cd}^{p(t),p^{(2)}(t)}(x) \leq k'$. In the latter case, by the properties of M, the output of A is equal to $\mathrm{K}^t(x)$.

Next, we show that the failure probability of A is at most δ. Observe that A fails only if $C(x, t, 1^k) = 0$, which implies that $\mathrm{cd}^t(x) \geq \mathrm{cd}^{t,p^{(3)}(t)}(x) > k$. This happens with probability at most $2^{-k+\log t} \leq \delta$ by Lemma 8.4. We conclude that

$$\Pr_{x \sim \mathcal{D}_n}[A(x; n, \delta) = \perp] \leq \delta.$$ ◀

Finally, we construct a slightly sub-exponential-time algorithm for GapMINKT from SoI.

▶ **Theorem 8.7.** *If SoI holds, then for every constant* $\delta > 0$*, there exists an algorithm* M *that, on input* (x, t) *such that* $|x| \leq t \leq 2^{|x|^{1-\delta}}$*, outputs a program of length* $\mathrm{K}^t(x)$ *that outputs* x *in time* t'*, where* $t' = 2^{O(|x|/\log|x|)}$*. The algorithm* M *runs in time* $2^{O(|x|/\log|x|)}$ *on input* (x, t)*.*

In particular, the length of the program d computed by M on input (x, t) satisfies that

$$\mathrm{K}^{t'}(x) \leq |d| \leq \mathrm{K}^t(x). \tag{21}$$

Proof of Theorem 8.7. Let M be the algorithm and p be the polynomial from Lemma 8.3. Fix a string $x \in \{0,1\}^*$ of length n and an integer $t \in \mathbb{N}$. Following [35], we consider the following telescoping sum:

$$\sum_{i=1}^{I} \mathrm{cd}^{p^{(i-1)}(t),p^{(i)}(t)}(x) = \mathrm{cd}^{t,p^{(I)}(t)}(x) \le n + O(1) \le 2n,$$

where I is a parameter chosen later. By taking the minimum term on the left-hand side, there exists an index $i \in [I]$ such that

$$\mathrm{cd}^{p^{(i-1)}(t),p^{(i)}(t)}(x) \le 2n/I. \tag{22}$$

We define an algorithm M' as follows. Given (x,t) as input, let $t_0 := \max\{t, p(n)\}$ and $t_i := p^{(i)}(t_0)$ for every $i \in [I]$. Let d_i be the program computed by $M(x, 1^{t_i}, 1^{2^{2n/I}})$ for every $i \in [I]$. The algorithm M' outputs the shortest program d_i that outputs x in time t_i.

The running time of M' is at most

$$\mathsf{poly}(t_I, 2^{2n/I}) \le t^{c^I} \cdot 2^{O(n/I)} \le 2^{O(c^I \cdot n^{1-\delta} + n/I)},$$

where c is some universal constant. By letting $I := \epsilon \log n$ for a sufficiently small constant $\epsilon > 0$, the running time can be bounded by $2^{O(n/\log n)}$.

We prove the correctness of M'. Let i^* be the index that satisfies Equation (22). Then, by the correctness of M, the program d_{i^*} prints x in time t_{i^*} and $|d_{i^*}| = \mathrm{K}^{t_{i^*}}(x)$. This means that the output of M' is well defined. Let d_i be the program computed by M'. By the definition of M', we have $|d_i| \le |d_{i^*}| = \mathrm{K}^{t_{i^*}}(x) \le \mathrm{K}^t(x)$. Moreover, the program d_i prints x in time $t_i \le t_I$. ◀

Proof of Theorem 8.2. The implication from Item 2 to 1 is proved in [33]. Theorems 4.1 and 8.6 prove the implication from Item 1 to 2. Theorem 4.1 proves the implication from Item 1 to 3. Theorem 8.5 proves the implication from Item 3 to 4.

The implication from Item 3 to 5 easily follows from Theorem 8.7: We describe a $2^{O(n/\log n)}$-time algorithm that solves $\mathrm{Gap}_\tau\mathrm{MINKT}$. Given an instance $(x, 1^t, 1^s)$ of $\mathrm{Gap}_\tau\mathrm{MINKT}$, if $t \le |x|^2$, then we use the search algorithm M of Theorem 8.7 and output 1 if and only if the length of the program computed by $M(x,t)$ is at most s. The search algorithm M runs in time $2^{O(|x|/\log|x|)}$. If $t > |x|^2$, then we use a trivial exhaustive search to find a shortest program that prints x in time t; this exhaustive search runs in time $2^{O(|x|)} \cdot t^{O(1)}$. In both cases, the algorithm runs in time $2^{O(n/\log n)}$, where n denotes the length $\Theta(|x| + t)$ of the instance $(x, 1^t, 1^s)$. The correctness of the algorithm follows from Equation (21). ◀

References

1 Dorit Aharonov and Oded Regev. Lattice problems in NP ∩ coNP. *J. ACM*, 52(5):749–765, 2005. `doi:10.1145/1089023.1089025`.

2 Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proceedings of the Symposium on the Theory of Computing (STOC)*, pages 99–108, 1996. `doi:10.1145/237814.237838`.

3 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from Random Strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006. `doi:10.1137/050628994`.

4 Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. One-Way Functions and a Conditional Variant of MKTP. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 7:1–7:19, 2021. `doi:10.4230/LIPIcs.FSTTCS.2021.7`.

5 Eric Allender, John Gouwar, Shuichi Hirahara, and Caleb Robelle. Cryptographic Hardness Under Projections for Time-Bounded Kolmogorov Complexity. In *Proceedings of the International Symposium on Algorithms and Computation (ISAAC)*, pages 54:1–54:17, 2021. doi:10.4230/LIPIcs.ISAAC.2021.54.

6 Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum Circuit Size, Graph Isomorphism, and Related Problems. *SIAM J. Comput.*, 47(4):1339–1372, 2018. doi:10.1137/17M1157970.

7 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing Disjunctive Normal Form Formulas and AC^0 Circuits Given a Truth Table. *SIAM J. Comput.*, 38(1):63–84, 2008. doi:10.1137/060664537.

8 Eric Allender, Michal Koucký, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. *J. Comput. Syst. Sci.*, 77(1):14–40, 2011. doi:10.1016/j.jcss.2010.06.004.

9 Luis Antunes, Lance Fortnow, Dieter van Melkebeek, and N. V. Vinodchandran. Computational depth: Concept and applications. *Theor. Comput. Sci.*, 354(3):391–404, 2006. doi:10.1016/j.tcs.2005.11.033.

10 Luis Antunes, Sophie Laplante, Alexandre Pinto, and Liliana C. M. Salvador. Cryptographic Security of Individual Instances. In *Proceedings of Information Theoretic Security (ICITS)*, pages 195–210, 2007. doi:10.1007/978-3-642-10230-1_17.

11 Luis Filipe Coelho Antunes and Lance Fortnow. Worst-Case Running Times for Average-Case Algorithms. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 298–303, 2009. doi:10.1109/CCC.2009.12.

12 Amos Beimel. Secret-Sharing Schemes: A Survey. In *The Third International Workshop on Coding and Cryptology (IWCC)*, pages 11–46, 2011. doi:10.1007/978-3-642-20901-7_2.

13 Josh Cohen Benaloh and Jerry Leichter. Generalized Secret Sharing and Monotone Functions. In *Proceedings of the International Cryptology Conference (CRYPTO)*, pages 27–35, 1988. doi:10.1007/0-387-34799-2_3.

14 George Robert Blakley. Safeguarding cryptographic keys. In *International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318, 1979. doi:10.1109/MARK.1979.8817296.

15 Manuel Blum and Sampath Kannan. Designing Programs that Check Their Work. *J. ACM*, 42(1):269–291, 1995. doi:10.1145/200836.200880.

16 Andrej Bogdanov and Luca Trevisan. Average-Case Complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1), 2006. doi:10.1561/0400000004.

17 Andrej Bogdanov and Luca Trevisan. On Worst-Case to Average-Case Reductions for NP Problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.

18 Harry Buhrman, Lance Fortnow, and Aduri Pavan. Some Results on Derandomization. *Theory Comput. Syst.*, 38(2):211–227, 2005. doi:10.1007/s00224-004-1194-y.

19 Harry Buhrman and Elvira Mayordomo. An Excursion to the Kolmogorov Random Strings. *J. Comput. Syst. Sci.*, 54(3):393–399, 1997. doi:10.1006/jcss.1997.1484.

20 Lijie Chen, Shuichi Hirahara, and Neekon 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)*, pages 67:1–67:17, 2022.

21 Irit Dinur, Prahladh Harsha, and Guy Kindler. Polynomially Low Error PCPs with polyloglog n Queries via Modular Composition. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 267–276, 2015. doi:10.1145/2746539.2746630.

22 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5):247–254, 2004. doi:10.1016/j.ipl.2003.11.007.

23 Joan Feigenbaum and Lance Fortnow. Random-Self-Reducibility of Complete Sets. *SIAM J. Comput.*, 22(5):994–1005, 1993. doi:10.1137/0222061.

24 Lance Fortnow and Martin Kummer. On Resource-Bounded Instance Complexity. *Theor. Comput. Sci.*, 161(1&2):123–140, 1996. doi:10.1016/0304-3975(95)00097-6.

25 Lance Fortnow, Troy Lee, and Nikolai K. Vereshchagin. Kolmogorov Complexity with Error. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, pages 137–148, 2006. doi:10.1007/11672142_10.

26 Lance Fortnow and Nick Reingold. PP is Closed Under Truth-Table Reductions. *Inf. Comput.*, 124(1):1–6, 1996. doi:10.1006/inco.1996.0001.

27 Eran Gat and Shafi Goldwasser. Probabilistic Search Algorithms with Unique Answers and Their Cryptographic Applications. *Electron. Colloquium Comput. Complex.*, page 136, 2011. URL: https://eccc.weizmann.ac.il/report/2011/136.

28 Halley Goldberg and Valentine 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)*, 007, 2022.

29 Oded Goldreich and Shafi Goldwasser. On the Limits of Nonapproximability of Lattice Problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000. doi:10.1006/jcss.1999.1686.

30 Ishay Haviv and Oded Regev. Tensor-based Hardness of the Shortest Vector Problem to within Almost Polynomial Factors. *Theory of Computing*, 8(1):513–531, 2012. doi:10.4086/toc.2012.v008a023.

31 Shuichi Hirahara. Non-black-box Worst-case to Average-case Reductions within NP. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 247–258, 2018.

32 Shuichi Hirahara. Characterizing Average-Case Complexity of PH by Worst-Case Meta-Complexity. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 50–60, 2020.

33 Shuichi Hirahara. Non-Disjoint Promise Problems from Meta-Computational View of Pseudorandom Generator Constructions. In *Proceedings of the Computational Complexity Conference (CCC)*, pages 20:1–20:47, 2020. doi:10.4230/LIPIcs.CCC.2020.20.

34 Shuichi Hirahara. Unexpected hardness results for Kolmogorov complexity under uniform reductions. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 1038–1051, 2020. doi:10.1145/3357713.3384251.

35 Shuichi Hirahara. Average-case hardness of NP from exponential worst-case hardness assumptions. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 292–302, 2021. doi:10.1145/3406325.3451065.

36 Shuichi Hirahara and Mikito Nanashima. On Worst-Case Learning in Relativized Heuristica. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2021.

37 Shuichi Hirahara, Igor Carboni Oliveira, and Rahul Santhanam. NP-hardness of Minimum Circuit Size Problem for OR-AND-MOD Circuits. In *Proceedings of the Computational Complexity Conference (CCC)*, pages 5:1–5:31, 2018. doi:10.4230/LIPIcs.CCC.2018.5.

38 Shuichi Hirahara and Rahul Santhanam. Errorless versus Error-prone Average-case Complexity. In *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, pages 38:1–38:23, 2022.

39 Shuichi Hirahara and Osamu Watanabe. Limits of Minimum Circuit Size Problem as Oracle. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 18:1–18:20, 2016. doi:10.4230/LIPIcs.CCC.2016.18.

40 Rahul 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)*, pages 34:1–34:26, 2020. doi:10.4230/LIPIcs.ITCS.2020.34.

41 Rahul Ilango. Constant Depth Formula and Partial Function Versions of MCSP are Hard. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 424–433, 2020.

42 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-Hardness of Circuit Minimization for Multi-Output Functions. In *Proceedings of the Computational Complexity Conference (CCC)*, pages 22:1–22:36, 2020. doi:10.4230/LIPIcs.CCC.2020.22.

43 Russell Impagliazzo. A Personal View of Average-Case Complexity. In *Proceedings of the Structure in Complexity Theory Conference*, pages 134–147, 1995. doi:10.1109/SCT.1995.514853.

44 Russell Impagliazzo. Relativized Separations of Worst-Case and Average-Case Complexities for NP. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 104–114, 2011. doi:10.1109/CCC.2011.34.

45 Russell Impagliazzo and Leonid 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)*, pages 812–821, 1990. doi:10.1109/FSCS.1990.89604.

46 Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E Requires Exponential Circuits: Derandomizing the XOR Lemma. In *Proceedings of the Symposium on the Theory of Computing (STOC)*, pages 220–229, 1997. doi:10.1145/258533.258590.

47 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 73–79, 2000. doi:10.1145/335305.335314.

48 Tarik Kaced. Almost-perfect secret sharing. In *Proceedings of the International Symposium on Information Theory (ISIT)*, pages 1603–1607, 2011. doi:10.1109/ISIT.2011.6033816.

49 Mauricio Karchmer and Avi Wigderson. On Span Programs. In *Proceedings of the Structure in Complexity Theory Conference*, pages 102–111, 1993. doi:10.1109/SCT.1993.336536.

50 Ker-I Ko. On the Complexity of Learning Minimum Time-Bounded Turing Machines. *SIAM J. Comput.*, 20(5):962–986, 1991. doi:10.1137/0220059.

51 Andrei N Kolmogorov. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7, 1965.

52 Troy Lee and Andrei E. Romashchenko. Resource bounded symmetry of information revisited. *Theor. Comput. Sci.*, 345(2-3):386–405, 2005. doi:10.1016/j.tcs.2005.07.017.

53 Leonid A. Levin. The Tale of One-Way Functions. *Probl. Inf. Transm.*, 39(1):92–103, 2003. doi:10.1023/A:1023634616182.

54 Leonid A. Levin. How do we succeed in tasks like proving Fermat's Theorem or predicting the Higgs boson?, 2021. An invited talk given at STOC 2021. The transcript is available at https://www.cs.bu.edu/fac/lnd/expo/stoc21/txt.pdf.

55 Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, Third Edition.* Texts in Computer Science. Springer, 2008. doi:10.1007/978-0-387-49820-1.

56 Yanyi Liu and Rafael Pass. On One-way Functions from NP-Complete Problems. *Electron. Colloquium Comput. Complex.*, 28:59, 2021. URL: https://eccc.weizmann.ac.il/report/2021/059.

57 Luc Longpré and Sarah Mocas. Symmetry of Information and One-Way Functions. *Inf. Process. Lett.*, 46(2):95–100, 1993. doi:10.1016/0020-0190(93)90204-M.

58 Luc Longpré and Osamu Watanabe. On Symmetry of Information and Polynomial Time Invertibility. *Inf. Comput.*, 121(1):14–22, 1995. doi:10.1006/inco.1995.1120.

59 Zhenjian Lu and Igor Carboni Oliveira. An Efficient Coding Theorem via Probabilistic Representations and Its Applications. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 94:1–94:20, 2021. doi:10.4230/LIPIcs.ICALP.2021.94.

60 Daniele Micciancio and Oded Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM J. Comput.*, 37(1):267–302, 2007. doi:10.1137/S0097539705447360.

61 Noam Nisan and Avi Wigderson. Hardness vs Randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

62 Igor Carboni Oliveira. Randomness and Intractability in Kolmogorov Complexity. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 32:1–32:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.32.

63 Igor Carboni Oliveira and Rahul Santhanam. Pseudodeterministic constructions in subexponential time. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 665–677, 2017. `doi:10.1145/3055399.3055500`.

64 Ran Raz, Omer Reingold, and Salil P. Vadhan. Extracting all the Randomness and Reducing the Error in Trevisan's Extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002. `doi:10.1006/jcss.2002.1824`.

65 Hanlin Ren and Rahul Santhanam. Hardness of KT Characterizes Parallel Cryptography. In *Proceedings of the Computational Complexity Conference (CCC)*, pages 35:1–35:58, 2021. `doi:10.4230/LIPIcs.CCC.2021.35`.

66 Detlef Ronneburger. *Kolmogorov Complexity and Derandomization*. PhD thesis, Rutgers UniversityDept. of Computer Science Laboratory for Computer Sci. Research Hill Center, NJ, USA, 2004.

67 Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979. `doi:10.1145/359168.359176`.

68 Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Lossless Condensers, Unbalanced Expanders, And Extractors. *Combinatorica*, 27(2):213–240, 2007. `doi:10.1007/s00493-007-0053-2`.

69 Boris A. Trakhtenbrot. A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984. `doi:10.1109/MAHC.1984.10036`.

70 Luca Trevisan. Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, 2001. `doi:10.1145/502090.502099`.

71 Luca Trevisan and Salil P. Vadhan. Pseudorandomness and Average-Case Complexity Via Uniform Reductions. *Computational Complexity*, 16(4):331–364, 2007. `doi:10.1007/s00037-007-0233-x`.

72 Emanuele Viola. On Constructing Parallel Pseudorandom Generators from One-Way Functions. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 183–197, 2005. `doi:10.1109/CCC.2005.16`.

73 Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005. `doi:10.1007/s00037-004-0187-1`.

74 AK Zvonkin and LA Levin. The complexity of finite objects and the algorithmic concepts of randomness and information. *UMN (Russian Math. Surveys)*, 25(6):83–124, 1970.

A Symmetry of Information from Weak Symmetry of Information

In this appendix, we present a simple proof of SoI based on weak symmetry of information. Weak symmetry of information is formally stated as follows:

▶ **Lemma A.1** (Weak Symmetry of Information [35]). *If* $\mathrm{GapMINKT} \in \mathsf{P}$ *and* $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{\epsilon n})$ *for some constant* $\epsilon > 0$*, then there exists a polynomial* p *such that, for any* $n, m \in \mathbb{N}$*, any* $t \geq n + m$*, any* $\epsilon > 0$*, and any* $x \in \{0,1\}^n$*,*

$$\Pr_{w \sim \{0,1\}^m}\left[\mathrm{K}^t(x, w) \geq \mathrm{K}^{p(t/\epsilon)}(x) + m - \log p(t/\epsilon)\right] \geq 1 - \epsilon.$$

Alternative Proof of Theorem 4.1. Let M be an algorithm for $\mathrm{Gap}_\tau\mathrm{MINKT}$, where τ is some polynomial. Fix x, $y \in \{0,1\}^*$, and $t \in \mathbb{N}$ such that $t \geq |x| + |y|$. Let $n := |x|$. We prove SoI by analyzing the behavior of

$$M(\mathrm{DP}_k(x; z) \cdot y, 1^{t'}, 1^s)$$

over a random choice of $z \sim \{0,1\}^{nk}$, where k, s and t' are parameters chosen later. To this end, we first compare the time-bounded Kolmogorov complexity of $\mathrm{DP}_k(x;z) \cdot y$ with that of $w \cdot y$ for random choices of $z \sim \{0,1\}^{nk}$ and $w \sim \{0,1\}^{nk+k}$: On one hand, since $\mathrm{DP}_k(x;z) \cdot y$ can be described by z, $k \le O(t)$, and a program that outputs (x, y), we have

$$\mathrm{K}^{t'}(\mathrm{DP}_k(x;z) \cdot y) \le \mathrm{K}^t(x,y) + |z| + \log t', \tag{23}$$

where $t' = \mathsf{poly}(t)$ is some polynomial. On the other hand, by Lemma A.1, there exists a polynomial p_0 such that

$$\mathrm{K}^{t'}(w \cdot y) \ge \mathrm{K}^{p_0(t')}(y) + |w| - \log p_0(t') \tag{24}$$

with probability at least $\frac{1}{2}$ over a random choice of $w \sim \{0,1\}^{nk+k}$.

Comparing Equations (23) and (24), when k is sufficiently large, the output distribution of $\mathrm{DP}_k(x;\text{-})$ can be distinguished from the uniform distribution by using the algorithm M. In more detail, let $s := \mathrm{K}^t(x,y) + |z| + \log t'$, which is the right-hand side of Equation (23). Let $k := \mathrm{K}^t(x,y) - \mathrm{K}^{p_0(t')}(y) + \log p_0(t') + \log t' + \log \tau(n', t') + 1$ so that the right-hand side of Equation (24) is greater than $s + \log \tau(n', t')$, where $n' := |w| + |y|$. Then, Equation (23) implies that $\mathrm{DP}_k(x;z) \cdot y$ is a Yes instance of $\mathrm{Gap}_\tau\mathrm{MINKT}$; thus, we obtain

$$\Pr_z\left[M(\mathrm{DP}_k(x;z) \cdot y, 1^{t'}, 1^s) = 1\right] = 1.$$

Equation (24) implies that $w \cdot y$ is a No instance of $\mathrm{Gap}_\tau\mathrm{MINKT}$ with probability at least $\frac{1}{2}$; thus, we obtain

$$\Pr_w\left[M(w \cdot y, 1^{t'}, 1^s) = 1\right] \le \frac{1}{2}.$$

Define a circuit D_y so that $D_y(w) := M(w \cdot y, 1^{t'}, 1^s)$ for every input $w \in \{0,1\}^{nk+k}$; then, it follows that

$$\Pr_z[D_y(\mathrm{DP}_k(x;z)) = 1] - \Pr_w[D_y(w) = 1] \ge \frac{1}{2},$$

which, by Lemma 2.2, implies that

$$\mathrm{K}^{p_1(t)}(x \mid D_y) \le k + \log p_1(t)$$

for some polynomial p_1. Finally, observe that the circuit D_y can be described by using $y \in \{0,1\}^*$, n, k, t', and $s \in \mathbb{N}$ as well as an $O(1)$-size program for M; therefore, we obtain

$$\mathrm{K}^{p_1(t)}(x \mid y) \le \mathrm{K}^t(x \mid D_y) + O(\log t) \le k + O(\log t) \le \mathrm{K}^t(x,y) - \mathrm{K}^{p_0(t')}(y) + O(\log t).$$

By choosing a large enough polynomial p, it follows that

$$\mathrm{K}^{p(t)}(x \mid y) + \mathrm{K}^{p(t)}(y) \le \mathrm{K}^t(x,y) + \log p(t).$$ ◀

A.1 Why Was Symmetry of Information Not Proved Before?

The alternative proof of Theorem 1.2 is reminiscent of the lemma of [35] that constructs a universal heuristic scheme from an algorithm for $\mathrm{Gap}(\mathrm{K}^{\mathsf{PH}} \text{ vs } \mathrm{K})$. In retrospect, the proof techniques for Theorem 1.2 were already developed in [35]. It is natural to ask why Theorem 1.2 was not proved in [35]. The reason is that previous results suggested the infeasibility of proving Theorem 1.2, as we explain below.

Weak symmetry of information is implicitly proved in [33] under the strong assumption that $\mathsf{DistPH} \subseteq \mathsf{AvgP}$. Specifically, [33] showed that if $\mathsf{DistPH} \subseteq \mathsf{AvgP}$, then

$$\mathrm{K}^{\mathsf{poly}(t)}(x) \le \mathrm{K}^{t,\mathsf{PH}}(x) + O(\log t)$$

for every $x \in \{0,1\}^*$ and every $t \ge |x|$.[20] That is, the PH-oracle time-bounded Kolmogorov complexity of x is approximately equal to $\mathrm{K}^t(x)$. By using the Kolmogorov–Levin proof of symmetry of information (as in [58]), it can be shown that

$$\mathrm{K}^{\mathsf{poly}(t),\mathsf{PH}}(y \mid x) + \mathrm{K}^{\mathsf{poly}(t),\mathsf{PH}}(x) \le \mathrm{K}^t(x,y) + O(\log t).$$

Now, for a random $y \sim \{0,1\}^m$, we have $\mathrm{K}^{\mathsf{poly}(t),\mathsf{PH}}(y \mid x) \approx |y|$ by Fact 3.2. Therefore, we obtain weak symmetry of information:

$$|y| + \mathrm{K}^{\mathsf{poly}(t)}(x) \le \mathrm{K}^t(x,y) + O(\log t)$$

holds with high probability over a random choice of $y \sim \{0,1\}^m$.

One may be tempted to try to prove symmetry of information by extending the result of [33] to

$$\mathrm{K}^{\mathsf{poly}(t)}(x \mid y) \le \mathrm{K}^{t,\mathsf{PH}}(x \mid y) + O(\log t) \tag{25}$$

under the assumption that $\mathsf{DistPH} \subseteq \mathsf{AvgP}$. However, this statement is in fact equivalent to the equivalence between the average-case easiness of PH and the worst-case easiness of PH (i.e., $\mathsf{P} = \mathsf{PH} \iff \mathsf{DistPH} \subseteq \mathsf{AvgP}$) [52, 24], which is a long-standing open question and cannot be proved by relativizing proof techniques [44, 36]. These results suggest the difficulty of proving symmetry of information because it relates conditional Kolmogorov complexity and unconditional Kolmogorov complexity, which previously seemed to imply Equation (25).

[35] proved $\mathsf{DistPH} \subseteq \mathsf{AvgP} \implies \mathsf{PH} \subseteq \mathsf{DTIME}(2^{O(n/\log n)})$, which is the first non-trivial worst-case-to-average-case connection of PH. [35] also proved weak symmetry of information under the weaker assumption that $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ than [33]. What was overlooked in [35] is that symmetry of information does not necessarily imply Equation (25). In fact, under the assumption that $\mathsf{DistPH} \subseteq \mathsf{AvgP}$, SoI implies a weaker statement that

$$\mathrm{K}^{\mathsf{poly}(t)}(x \mid y) \le \mathrm{K}^{t,\mathsf{PH}}(x \mid y) + \mathrm{cd}^{t,\mathsf{poly}(t)}(y) + O(\log t). \tag{26}$$

This statement looks quite similar to Equation (25), especially because the computational depth of y is small for most strings y (Lemma 8.4). Although Equation (26) is not sufficient to obtain $\mathsf{P} = \mathsf{PH}$, it does suffice to prove $\mathsf{PH} \subseteq \mathsf{DTIME}(2^{O(n/\log n)})$, which provides the alternative proof of the main results of [35] presented in Section 7. The fact that SoI provides the worst-case-to-average-case connections indicates the importance of SoI in average-case complexity theory.

[20] $\mathrm{K}^{t,\mathsf{PH}}(x)$ is an informal notation that represents the A-oracle t-time-bounded Kolmogorov complexity of x for an oracle $A \in \mathsf{PH}$. The statement is true for every oracle $A \in \mathsf{PH}$.