Electronic Colloquium on Computational Complexity, Report No. 211 (2025)

Equivalence Between Coding and Complexity Lower Bounds

Jinqiao Hu*
Department of Computer Science
University of Warwick

Zhenjian Lu†
Department of Computer Science
University of Victoria

Igor C. Oliveira‡
Department of Computer Science
University of Warwick

November 24, 2025

Abstract

The classical coding theorem in Kolmogorov complexity [Lev74] states that if a string x is sampled with probability $\geq \delta$ by an algorithm with prefix-free domain, then $\mathsf{K}(x) \leq \log(1/\delta) + O(1)$. Motivated by applications in algorithms, average-case complexity, learning, and cryptography, computationally efficient variants of this result have been established for several recently introduced probabilistic measures of time-bounded Kolmogorov complexity, including rKt [LO21] and pK^t [LOZ22]. However, establishing a coding theorem for classical (non-probabilistic) notions of time-bounded Kolmogorov complexity, such as Kt complexity [Lev84], remains a longstanding open problem despite its significance. In particular, the current status of coding results reveals a fundamental gap in our understanding of the role of randomness in data compression.

In this work, we make progress by establishing the first equivalence between coding for Kt complexity and complexity lower bounds. Building on this equivalence, we show that similar characterizations hold for *non-deterministic* and *zero-error* variants of Kt complexity, demonstrating that coding is equivalent to a corresponding complexity separation in each case. We complement these results by establishing additional equivalences involving the computational hardness of approximating time-bounded Kolmogorov complexity, along with an *unconditional* lower bound on the complexity of approximating zero-error time-bounded Kolmogorov complexity.

These results reveal novel connections between coding (the existence of succinct encodings), complexity separations (e.g., NEXP versus BPP), and meta-complexity (the complexity of deciding if a succinct encoding exists). In particular, our work provides a new perspective on frontier questions in complexity theory and explains why coding theorems exist for rKt and pK^t but remain unknown for other measures of time-bounded Kolmogorov complexity. Finally, our results determine the minimal hardness assumptions sufficient for coding in different settings.

*E-mail: jinqiao.hu@warwick.ac.uk
†E-mail: zhenjianlu@uvic.ca
‡E-mail: igor.oliveira@warwick.ac.uk

ISSN 1433-8092

Contents

1 Introduction

1.1 Context and Motivation

The investigation of data compression problems and their computational complexity has seen significant progress and impact in recent years. In particular, a sequence of works have established that different notions of compression and their associated computational problems can be used to capture major open problems from theoretical computer science. Some notable examples include the existence of one-way functions [LP20] and secure key-agreement protocols [BLMP23] in cryptography, and the efficient learnability of Boolean circuits [CIKK16] and the complexity of inductive inference [HN23] in computational learning theory. Strikingly, for several statements that do not refer to compression, the only known proof of the result seems to crucially rely on ideas and techniques from compression. Among them, we have the existence of learning speedups [OS17], a connection between worst-case and average-case complexity [Hir21], and lower bounds on program size overhead in indistinguishability obfuscation [LMOP24].

A central tool in the study of compression is the coding theorem from Kolmogorov complexity [Lev74]. It states that if a string $x \in \{0,1\}^n$ is sampled with probability $\geq \delta$ by an algorithm with prefix-free domain then $\mathsf{K}(x) \leq \log(1/\delta) + O(1)$. This general result connects randomized computations to compression and is widely considered to be one of the pillars of the theory of Kolmogorov complexity [Lee06].

Due to the time-unbounded nature of Kolmogorov complexity, the coding theorem as stated above is typically not sufficient in algorithmic applications where the running time of algorithms is relevant. A few years ago, [LO21, LOZ22] established a similar result for certain time-bounded variants of Kolmogorov complexity, namely, rKt and pK^t complexities. Since then, these results have found several applications in cryptography [IRS22, LP23, HIL$^+$23, HLO24, HLN24, LP25], algorithm design and hardness results [HKLO24, LORS24, GK24], average-case complexity [LOZ22, LS24], learning theory [GKLO22, HN23, GK23], and complexity lower bounds [Hir22, San23, LP24].

While unconditional, a drawback of these coding results is that rKt and pK^t are probabilistic notions of time-bounded Kolmogorov complexity [LO22], meaning that randomness (and consequently uncertainty) is essential to the representation of the string x. Establishing a coding theorem for classical (non-probabilistic) notions of time-bounded Kolmogorov complexity, such as Levin's Kt complexity [Lev84], remains a long-standing open problem. It provides a natural computational setting where randomness offers a significant advantage over deterministic computations given our current knowledge of algorithms and complexity theory.

Let κ be a measure of (time-bounded) Kolmogorov complexity, such as Kt, rKt, etc. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family. In order to put our results in perspective, we can informally classify coding theorems according to the amount of compression they achieve:

- Optimal Coding: $\kappa(x) \leq \log(1/\mathcal{D}_n(x)) + O(\log n)$. This is known for $\kappa \in \{\mathsf{K}, \mathsf{pK}^t\}$ [Lev84, LOZ22].
- Near-Optimal Coding: $\kappa(x) \leq O(\log(1/\mathcal{D}_n(x)) + \log n)$. This is known for $\kappa = \mathsf{rKt}$ [LO21, LOZ22].
- Weak Coding: $\kappa(x) \leq \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}$, for a fixed but arbitrarily small $\varepsilon > 0$.
- Non-Trivial Coding: $\kappa(x) \leq n - \omega(\log n)$ assuming, say, x is generated with probability $\mathcal{D}_n(x) \geq 0.99$.

For non-probabilistic measures of time-bounded Kolmogorov complexity, *conditional* results are known:

– Antunes and Fortnow [AF09] established that optimal coding holds for K^t under the assumption that exponential time is not infinitely often in subexponential space.

– Under the existence of pseudorandom generators of exponential stretch secure against non-uniform circuits, near-optimal coding holds for Kt (e.g., by combining [LO21] and [GKLO22, Section A.2]).

– In the other direction, Lee [Lee06, Chapter 5] proved that if optimal coding holds for $\mathsf{K}^{\mathsf{poly}}$ then $\mathsf{EXP} \neq \mathsf{BPP}$.

There is a sharp contrast between the *unconditional* results established for probabilistic measures such as rKt and pK^t, and the *conditional* results known for non-probabilistic measures such as Kt, which require strong computational assumptions. Motivated by this discrepancy and with the goal of advancing our understanding of randomness in computation, we systematically investigate the prospects of achieving better coding results in time-bounded Kolmogorov complexity. From a technical perspective, we are interested in the following basic questions:

(1) *Is it possible to show non-trivial coding for* Kt *without hardness assumptions?*

(2) *If non-trivial coding for* Kt *is difficult to achieve, can we at least improve the existing coding result for* rKt *in order to achieve zero-error encodings?*

(3) *Is there a connection between coding* (*i.e., the existence of succinct encodings*) *and the hardness of the corresponding meta-computational problem* (*i.e., the task of deciding if a succinct encoding exists*)*?*

More broadly, we seek to deepen our knowledge of the role of randomness in data compression and identify when it can be eliminated without incurring significant overhead, under minimal hardness assumptions.

1.2 Results

Summary. Our main contribution is to show that coding and complexity lower bounds are in fact *equivalent*. As a consequence of our results and techniques, we also establish a surprising equivalence between *weak coding* and *non-trivial* coding for different measures of time-bounded Kolmogorov complexity. Finally, we extend these equivalences by considering the computational complexity of estimating time-bounded Kolmogorov complexity. This extends our results and uncovers a novel connection between the existence of succinct encodings (coding) and the feasibility of deciding when a succinct encoding exists (meta-complexity).

Altogether, our results completely answer Questions 1-3 stated above. They also show that the validity of a key property (coding) of Kolmogorov complexity in the time-bounded setting captures several frontier questions in complexity theory. This exhibits another significant example of the relevance of compression to central questions in theoretical computer science.

Organization. In Section 1.2.1, we establish the equivalence between coding and complexity lower bounds for Kt complexity. Section 1.2.2 and Section 1.2.3 extend these results to the non-deterministic and zero-error variants, denoted by nKt and zKt, respectively. The computational complexity of estimating time-bounded Kolmogorov complexity is explored in Section 1.2.4.

1.2.1 Coding for Deterministic Time-Bounded Kolmogorov Complexity

Fix an efficient universal machine U. Recall that for a string $x \in \{0,1\}^*$, we let

$$\mathsf{Kt}(x) \triangleq \min_{p \in \{0,1\}^*,\, t \in \mathbb{N}} \Big\{ |p| + \lceil \log t \rceil : U^t(p) = x \Big\}.$$

The notation $U^t(p)$ denotes the output of U on input string p when it computes for at most t steps. It is also possible to consider a relativized version of Kt, namely $\mathsf{Kt}^{\mathcal{O}}$, where we give the universal Turing machine U oracle access to the set $\mathcal{O} \subseteq \{0,1\}^*$.

Recall that an ensemble $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ of distributions $\mathcal{D}_n$ supported over $\{0,1\}^*$ is polynomial-time samplable if there is a polynomial-time randomized algorithm A whose output $A(1^n, r)$ for $r \sim \{0,1\}^*$ is distributed according to $\mathcal{D}_n$. We denote the probability of an element x over $\mathcal{D}_n$ by $\mathcal{D}_n(x) \in [0,1]$.

Theorem 1. *The following statements are equivalent.*

1. $\mathsf{EXP} \neq \mathsf{BPP}$.

2. **(Weak coding for Kt.)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*
$$\mathsf{Kt}(x) \leq \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}.$$

3. **(Non-trivial coding for Kt.)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $\mathcal{D}_n(x_n) \geq 1 - n^{-c}$ for every n. Then for infinitely many n, we have*
$$\mathsf{Kt}(x_n) \leq n - \omega(\log n).$$

As a consequence of this result, merely showing that Kt admits non-trivial coding requires a hardness assumption. This addresses Question 1 from Section 1.1.

The equivalence stated in Theorem 1 significantly strengthens a result from [Lee06, Chapter 5] showing that optimal coding for $\mathsf{K}^{\mathsf{poly}}$ yields $\mathsf{EXP} \neq \mathsf{BPP}$. In addition, our proof that Item 3 implies Item 1 is considerably simpler.

Theorem 1 also establishes an equivalence between *weak coding* and *non-trivial coding* for Kt. On the other hand, we observe in Section 3.2 that *near-optimal coding* for Kt implies the significantly stronger separation $\mathsf{DTIME}[2^{O(n)}] \nsubseteq \mathsf{i.o.BPTIME}[2^n]$. Consequently, in contrast to the equivalence between non-trivial coding and weak coding, we are unlikely to obtain an equivalence between weak-coding and near-optimal coding given our current knowledge of complexity theory, unless we can show how to boost the separation $\mathsf{EXP} \neq \mathsf{BPP}$ to a much stronger result.[1]

1.2.2 Coding for Non-Deterministic Time-Bounded Kolmogorov Complexity

Next, we establish an equivalence between coding with non-deterministic encodings and complexity lower bounds. We will need the following definition, which offers a natural extension of Kt to the setting of non-deterministic computations. For a string $x \in \{0,1\}^*$, the *non-deterministic time-bounded Kolmogorov complexity of x* is defined as

$$\mathsf{nKt}(x) \triangleq \min_{p\in\{0,1\}^*,\, t\in\mathbb{N}} \left\{ |p| + \lceil \log t \rceil \;\middle|\; \begin{array}{l} \bullet\ \forall w \in \{0,1\}^t, U(p,w) \text{ outputs } x \text{ or } \bot \text{ within } t \text{ steps} \\ \bullet\ \exists w \in \{0,1\}^t, U(p,w) \text{ outputs } x \text{ within } t \text{ steps} \end{array} \right\}.$$

The above definition is *equivalent* to a "local" notion of non-deterministic Kolmogorov complexity investigated in [AKRR11], which considers instead individual bits of x (see Appendix A).

Theorem 2. *The following statements are equivalent.*

1. $\mathsf{NEXP} \neq \mathsf{BPP}$.

[1] In fact, it seems plausible that near-optimal coding for Kt is *equivalent* to lower bounds of the form $\mathsf{DTIME}[2^{O(n)}] \nsubseteq \mathsf{BPTIME}[2^n]$. However, it is unclear to us how to establish this equivalence using our techniques, which are based on the theory of computational pseudorandomness.

2. **(Weak coding for nKt)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*

$$\mathsf{nKt}(x) \le \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}.$$

3. **(Non-trivial coding for nKt.)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $\mathcal{D}_n(x_n) \ge 1 - n^{-c}$ for every n. Then for infinitely many n, we have*

$$\mathsf{nKt}(x_n) \le n - \omega(\log n).$$

Moreover, the above holds if we replace NEXP with $\mathsf{EXP}^{\mathsf{NP}}$, and nKt with $\mathsf{Kt}^{\mathsf{NP}}$.[2]

As a consequence of this result, even if we could achieve non-trivial coding using non-deterministic encodings, a new complexity lower bound would follow. Indeed, Theorem 2 provides a new characterization of the NEXP versus BPP problem as a statement about the existence of succinct encodings.

1.2.3 Coding for Zero-Error Time-Bounded Kolmogorov Complexity

Finally, we introduce a natural zero-error variant of Kt complexity, which can also be seen as the restriction of rKt [Oli19] to errorless encodings. To the best of our knowledge, this definition has not been considered in previous work. For a string $x \in \{0,1\}^*$, we let

$$\mathsf{zKt}(x) \triangleq \min_{p\in\{0,1\}^*,\, t\in\mathbb{N}} \left\{ |p| + \lceil \log t \rceil \;\middle|\; \begin{array}{l} \bullet\ \forall r \in \{0,1\}^t,\, U(p,r) \text{ outputs } x \text{ or } \perp \text{ within } t \text{ steps} \\ \bullet\ \mathbf{Pr}_r[U(p,r) \text{ outputs } x \text{ within } t \text{ steps}] \ge \frac{2}{3} \end{array} \right\}.$$

In Section 5.3, we observe that the existing near-optimal coding result for rKt [LO21] yields zero-error encodings whenever the distribution $\mathcal{D}_n$ is *flat*, i.e., when it is uniformly distributed over a set $S \subseteq \{0,1\}^n$. In contrast, our next result indicates that it will be difficult to extend this zero-error coding theorem to *all* polynomial-time samplable distributions, even in the non-trivial coding regime.

Theorem 3. *The following statements are equivalent.*

1. $\mathsf{prZPEXP} \neq \mathsf{prBPP}$.

2. **(Weak coding for zKt)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*

$$\mathsf{zKt}(x) \le \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}.$$

3. **(Non-trivial coding for zKt)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $\mathcal{D}_n(x_n) \ge 1 - n^{-c}$ for every n. Then for infinitely many n, we have*

$$\mathsf{zKt}(x_n) \le n - \omega(\log n).$$

[2]Recall that $\mathsf{Kt}^{\mathsf{NP}}$ denotes the extension of Kt where the universal machine U has access to a SAT oracle.

Therefore, obtaining a zero-error version of existing coding results yields a new complexity separation. This addresses Question 2 from Section 1.1. Note that zero-error coding (Theorem 3) implies a stronger separation than non-deterministic coding (Theorem 2), which is expected since $\mathsf{nKt}(x) \leq \mathsf{zKt}(x)$ for every string x (i.e., it is more challenging to achieve a zero-error encoding).

For the interested reader, in Section 5.2 we investigate the possibility of achieving the stronger separation $\mathsf{ZPEXP} \neq \mathsf{BPP}$ from coding for zKt.

1.2.4 Complexity Separations and Meta-Complexity

Let MKtP be the following problem: Given $(x, 1^s)$, where $x \in \{0,1\}^*$ and $s \in \mathbb{N}$, decide whether $\mathsf{Kt}(x) \leq s$. We also consider a parametrized "gap" version of MKtP. Let $s_1, s_2 \colon \mathbb{N} \to \mathbb{N}$ be such that $s_1(n) < s_2(n)$ for every large n. Define $\mathsf{MKtP}[s_1, s_2]$ as the problem of deciding, given $x \in \{0,1\}^n$, whether $\mathsf{Kt}(x) \leq s_1(n)$ or $\mathsf{Kt}(x) \geq s_2(n)$. When $s_1(n) = n^{\varepsilon}$ and $s_2(n) = n - 1$, we might informally refer to the problem as Gap-MKtP.

Similarly, we can define analogous problems for nKt, $\mathsf{Kt}^{\mathsf{NP}}$, and zKt, denoted as MnKtP, $\mathsf{MKt}^{\mathsf{NP}}\mathsf{P}$, and MzKtP, respectively.

We identify these problems with their corresponding (promise) languages in a natural way.

The problem Gap-MKtP is complete for EXP under *non-uniform* polynomial-time reductions [ABK+06]. A similar result also holds for Gap-MnKtP, i.e., Gap-MnKtP is complete for $\mathsf{NEXP}/_{\mathsf{poly}}$ under *non-uniform* polynomial-time reductions [AKRR11]. These results imply that $\mathsf{EXP} \not\subseteq \mathsf{P}/_{\mathsf{poly}}$ (resp. $\mathsf{NEXP} \not\subseteq \mathsf{P}/_{\mathsf{poly}}$) if and only if $\mathsf{Gap\text{-}MKtP} \notin \mathsf{P}/_{\mathsf{poly}}$ (resp, $\mathsf{Gap\text{-}MnKtP} \notin \mathsf{P}/_{\mathsf{poly}}$).

On the other hand, it was also established that Gap-MKtP captures the hardness of EXP with respect to *uniform* randomized algorithms. That is, $\mathsf{EXP} \neq \mathsf{BPP}$ if and only if $\mathsf{Gap\text{-}MKtP} \notin \mathsf{prBPP}$ [ABK+06]. Here, we extend this result to the notion of nKt.

Theorem 4. *The following are equivalent.*

1. $\mathsf{NEXP} \neq \mathsf{BPP}$.
2. $\mathsf{MnKtP}[n^{\varepsilon}, n-1] \notin \mathsf{prBPP}$, *for all* $\varepsilon > 0$.

Moreover, the above holds if we replace MnKtP *with* $\mathsf{MKt}^{\mathsf{NP}}\mathsf{P}$*, and* NEXP *with* $\mathsf{EXP}^{\mathsf{NP}}$.[3]

For zero-error time bounded Kolmogorov complexity, we show that the problem of approximating zKt is at least as hard as solving every problem in prZPEXP with respect to two-sided error randomized algorithms.

Theorem 5. *If* $\mathsf{MzKtP}[n^{\varepsilon}, n-1] \in \mathsf{prBPP}$ *for some* $\varepsilon > 0$*, then* $\mathsf{prZPEXP} = \mathsf{prBPP}$.

Finally, we obtain an *unconditional* lower bound for approximating zKt against *zero-error* randomized algorithms.

Theorem 6. $\mathsf{MzKtP}[n^{\varepsilon}, n-1] \notin \mathsf{prZPTIME}\left[2^{\mathsf{polylog}(n)}\right]$*, for all* $\varepsilon > 0$.

Theorem 6 builds on a lower bound for approximating rKt from [Oli19] (i.e., $\mathsf{MrKtP} \notin \mathsf{BPP}$). Since zKt is an intermediate measure between Kt and rKt, in a sense, the result can be seen as progress towards showing that $\mathsf{MKtP} \notin \mathsf{P}$. The latter is a well-known open problem in meta-complexity (see, e.g., [ABK+06]).

[3] In fact, in all these results, the proof implicitly shows that the gap version of the problem is easy if and only if the non-gap version is easy. For instance, it is known that $\mathsf{Gap\text{-}MKtP} \notin \mathsf{prBPP}$ if and only if $\mathsf{MKtP} \notin \mathsf{BPP}$. This will also be the case for the equivalences established in this paper.

1.3 Summary of Equivalences and Concluding Remarks

For convenience of the reader, we summarize the equivalences established in this paper in Table 1. Note that, from the perspective of derandomization, our results identify the minimal complexity-theoretic assumptions required to obtain coding for different measures of Kolmogorov complexity.

Complexity Separation	**Coding Theorem**	**Meta-Complexity**
$\mathsf{EXP} \neq \mathsf{BPP}$	Weak/Non-Trivial Coding for Kt	$\text{Gap-}\mathsf{MKtP} \notin \mathsf{prBPP}$
$\mathsf{NEXP} \neq \mathsf{BPP}$	Weak/Non-Trivial Coding for nKt	$\text{Gap-}\mathsf{MnKtP} \notin \mathsf{prBPP}$
$\mathsf{EXP}^{\mathsf{NP}} \neq \mathsf{BPP}$	Weak/Non-Trivial Coding for $\mathsf{Kt}^{\mathsf{NP}}$	$\text{Gap-}\mathsf{MKt}^{\mathsf{NP}}\mathsf{P} \notin \mathsf{prBPP}$
$\mathsf{prZPEXP} \neq \mathsf{prBPP}$	Weak/Non-Trivial Coding for zKt	$\Longrightarrow \text{Gap-}\mathsf{MzKtP} \notin \mathsf{prBPP}$

Table 1: Summary of Equivalences: In each row, the three items are equivalent, except for the last row, where the complexity separation and the coding theorem are equivalent, and they imply that probabilistic polynomial-time algorithms cannot approximate zKt ($\text{Gap-}\mathsf{MzKtP} \notin \mathsf{prBPP}$).

As mentioned above, the equivalence between $\mathsf{EXP} \neq \mathsf{BPP}$ and $\text{Gap-}\mathsf{MKtP} \notin \mathsf{prBPP}$ included in the first row of Table 1 was established in [ABK+06]. It is unclear to us how to prove that $\text{Gap-}\mathsf{MzKtP} \in \mathsf{prBPP}$ from $\mathsf{prZPEXP} = \mathsf{prBPP}$, which would provide the equivalence between all items in the last row of Table 1.

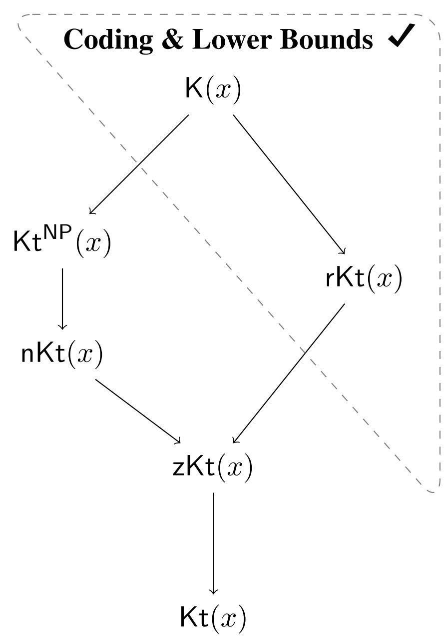

Figure 1: An arrow from a Kolmogorov complexity measure κ_1 to κ_2 indicates that $\kappa_1(x) \leq \kappa_2(x)$ for every string x. Our results show that, for each measure κ, the existence of weak coding and a corresponding complexity separation against BPP are equivalent (see Table 1). In particular, for $\kappa \in \{\mathsf{K}, \mathsf{rKt}\}$, both coding and lower bounds are known, while for $\kappa \in \{\mathsf{Kt}^{\mathsf{NP}}, \mathsf{nKt}, \mathsf{zKt}, \mathsf{Kt}\}$, these remain longstanding challenges.

In contrast to the equivalences described in Table 1, for the two-sided error notion of time-bounded Kolmogorov complexity rKt, we know *unconditionally* that:

- $\mathsf{BPEXP} \not\subseteq \mathsf{BPP}$ (see, e.g., [BFS09] and references therein);
- a near-optimal coding theorem holds [LO21]; and
- $\mathsf{Gap\text{-}MrKtP} \notin \mathsf{prBPP}$ [Oli19].

Our work highlights that this is not a coincidence, i.e., these different statements are intimately related (see Figure 1). Moreover, our results provide an equivalence between coding (i.e., the existence of succinct encodings) and the hardness of the corresponding meta-computational problem (i.e., the task of deciding if a succinct encoding exists). In other words, they answer affirmatively Question 3 stated in Section 1.1.

Finally, our results show that any non-trivial compression (even with nondeterminism) would imply new separations in complexity theory and advance our understand of the power and limits of randomness in computation. It would be worthwhile to investigate whether this perspective can be combined with other techniques and employed as a concrete method for establishing new lower bounds.

1.4 Techniques

In this section, we explain the main ideas behind our proofs. We start off with our results for Kt complexity. We then discuss the non-deterministic and zero-error settings, which require additional ideas and more elaborate proofs. In particular, the techniques we develop to establish our results in the context of zero-error Kolmogorov complexity might be of independent interest.

Equivalence Between Coding for Kt and $\mathsf{EXP} \neq \mathsf{BPP}$. We first describe how to obtain $\mathsf{EXP} \neq \mathsf{BPP}$ from a non-trivial coding theorem for Kt. Note that, by a standard padding argument, it suffices to show that $\mathsf{EE} \neq \mathsf{BPE}$, where $\mathsf{EE} \triangleq \mathsf{DTIME}[2^{2^{O(n)}}]$ and $\mathsf{BPE} \triangleq \mathsf{BPTIME}[2^{O(n)}]$. Our goal is then to diagonalize against BPE within EE. The first observation is that if we have a non-trivial coding theorem for Kt, then the truth table of every language in BPE on n-bit inputs will have Kt complexity strictly less than 2^n, for infinitely many n. To see this, consider any language $L \in \mathsf{BPE}$ and a sampler A that, on input 1^N, aims to output the N-bit truth table of $L^{=n}$, where $n = \log N$,[4] by running a probabilistic machine for computing L on every input in $\{0,1\}^n$. It is not hard to see that $A(1^N)$ can be implemented to run in time $\mathsf{poly}(N)$ and outputs $\mathsf{tt}(L^{=n})$ with probability at least $1 - 1/\mathsf{poly}(N)$. Then, by invoking the non-trivial coding theorem for Kt on this sampler, we get that for infinitely many n, $\mathsf{Kt}(\mathsf{tt}(L^{=n})) \leq 2^n - \omega(n)$. Note that this holds for every $L \in \mathsf{BPE}$. To diagonalize against all such L, we define a language L_{hard} whose 2^n-bit truth table has Kt complexity at least $2^n - 1$ for all n. Since one can compute an N-bit string with Kt complexity at least $N - 1$ in time $\mathsf{poly}(2^N)$ using exhaustive search, it follows that L_{hard} is computable in EE.

To derive a non-trivial coding theorem for Kt from $\mathsf{EXP} \neq \mathsf{BPP}$, the main idea is to use the hardness-vs-randomness framework to construct a *pseudorandom generator* (PRG). More specifically, by classical results in [IW01, TV07], we obtain that if $\mathsf{PSPACE} \neq \mathsf{BPP}$, then for every $b, c > 0$, there exists a PRG G that takes a short seed of length $n^{1/b}$, runs in time $2^{O(n^{1/b})}$, and outputs a longer string of length n^c that can fool any n^b-time algorithm D, for infinitely many n. More formally:

$$\left| \Pr_{z \sim \{0,1\}^{n^{1/b}}}[D(G(z)) = 1] - \Pr_{u \sim \{0,1\}^{n^c}}[D(u) = 1] \right| \leq \frac{1}{n^b}.$$

[4] For simplicity, let's assume that N is always a power of two.

Let $\mathcal{D} \triangleq \{\mathcal{D}_n\}$ be a polynomial-time samplable distribution family and A be its sampler, i.e, $A(u)$ is distributed according to $\mathcal{D}_n$ for uniformly random $u \sim \{0,1\}^{n^c}$, where $c > 0$ is some constant. Let $x \in \mathsf{Support}(\mathcal{D}_n)$ be the string for which we aim to find a short encoding. (For simplicity, let's assume that each $\mathcal{D}_n$ is supported on $\{0,1\}^n$.) First observe that the weak coding theorem holds trivially on a given n-bit string x if $\mathcal{D}_n(x) < 1/n^{1/\varepsilon}$, since in this case the desired encoding bound is larger than the length of the string. Therefore, we can assume without loss of generality that $\mathcal{D}_n(x) \geq 1/n^{1/\varepsilon}$.

Consider the function D_x, defined as $D_x(y) = 1$ if and only if $A(y) = x$. Note that $\mathbf{Pr}_u[D_x(u) = 1] \geq 1/n^{1/\varepsilon}$. Using the pseudorandom property of G (with $b > 1/\varepsilon$ chosen sufficiently large), it follows that:

$$\mathop{\mathbf{Pr}}_{z \sim \{0,1\}^{n^{1/b}}}[D_x(G(z)) = 1] \geq \mathop{\mathbf{Pr}}_{u \sim \{0,1\}^{n^c}}[D_x(u) = 1] - \frac{1}{n^b} > 0.$$

This implies the existence of some $z \in \{0,1\}^{n^{1/b}}$ such that $A(G(z)) = x$. Given the descriptions of A, G, and the seed z, x can be recovered in time $2^{O(n^{1/b})}$, yielding $\mathsf{Kt}(x) \leq O(n^{1/b}) \leq n^{\varepsilon}$.

However, there is an issue in the above argument: the function D_x depends on x, making it *non-uniform*, while the PRG G is designed to fool only *uniform* algorithms. The key observation is that the PRG obtained from [IW01, TV07] possesses a slightly stronger property: it not only fools uniform algorithms but in our case also fools D_x *with probability at least* $1 - 1/n^b$ *over* x *sampled from any* n^b*-time samplable distribution* (see Theorem 12). Since x is assumed to be sampled from $\mathcal{D}_n$ with probability at least $1/n^{1/\varepsilon}$, we conclude that G can successfully fool D_x in this case; otherwise, it would fail with probability at least $1/n^{1/\varepsilon} > 1/n^b$, contradicting the pseudorandomness guarantee.

The above requires assuming that $\mathsf{PSPACE} \neq \mathsf{BPP}$, while we only have $\mathsf{EXP} \neq \mathsf{BPP}$. We address this with a standard win-win argument. If $\mathsf{PSPACE} \neq \mathsf{BPP}$, then we are done. Otherwise, if $\mathsf{PSPACE} = \mathsf{BPP}$, our assumption that $\mathsf{EXP} \neq \mathsf{BPP}$ implies $\mathsf{EXP} \neq \mathsf{PSPACE}$. By the classical Karp–Lipton result [KL80], which states that if $\mathsf{EXP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$, then $\mathsf{EXP} = \mathsf{PSPACE}$, it follows that $\mathsf{EXP} \not\subseteq \mathsf{SIZE}[\mathsf{poly}]$. Using a different hardness-vs-randomness framework [BFNW93] (see Theorem 11), which allows us to produce pseudorandomness using the hard truth table of a language in EXP, this also yields an infinitely-often secure PRG with sub-polynomial seed length.[5] Such a PRG can be used to achieve weak coding for Kt as described in previous paragraphs.

Equivalence Between Coding for nKt and $\mathsf{NEXP} \neq \mathsf{BPP}$. To obtain $\mathsf{NEXP} \neq \mathsf{BPP}$ from a non-trivial coding theorem for nKt, one might consider resembling the proof used in the previous case. However, for this approach to work, we would need to be able to construct an N-bit string with high nKt-*complexity* in time $\mathsf{poly}(2^N)$, which is not clear how to achieve (even non-deterministically).[6] Here, we present a more sophisticated diagonalization argument that bypasses the need for this task. For simplicity, we describe how to obtain $\mathsf{NEXP} \neq \mathsf{BPP}$ from a *weak* coding theorem for nKt.

First of all, if we have a weak coding theorem for nKt, by a similar argument as described in the previous case, we get that the truth table of every language in BPE on n-bit inputs will have nKt-*complexity* less than $2^{\varepsilon n}$, for infinitely many n. This means one can *non-deterministically* generate these truth tables in time $2^{2^{\varepsilon n}}$ with at most $2^{\varepsilon n}$-bits of advice. This allows us to conclude that $\mathsf{BPE} \subseteq \mathsf{i.o.NTIME}\left[2^{2^{\varepsilon n}}\right]/_{2^{\varepsilon n}}$.

Now suppose, for the sake of contradiction, $\mathsf{NEXP} = \mathsf{BPP}$. Note that by the existence of NE-complete problems under linear-time reductions, this implies $\mathsf{NE} \subseteq \mathsf{BPTIME}[n^k]$ for some *fixed* $k > 0$. Then we

[5] In fact, the PRG obtained in this case can even fool non-uniform algorithms.

[6] Note that a naive algorithm for this task runs in time at least 2^{2^N}. In other words, we need to consider each candidate nondeterministic program running in time at most 2^N, and enumerating over all choices of the nondeterministic string to check that the program is suitable takes doubly exponential time.

have

$$\begin{aligned}
\mathsf{EE} &\subseteq \mathsf{BPE} && \text{(by padding and } \mathsf{NEXP} = \mathsf{BPP}\text{)}\\
&\subseteq \mathsf{i.o.NTIME}\left[2^{2^{\varepsilon n}}\right]/_{2^{\varepsilon n}} && \text{(by the previous paragraph)}\\
&\subseteq \mathsf{i.\,o.\,BPTIME}\left[2^{k\cdot\varepsilon\cdot n}\right]/_{2^{\varepsilon n}} && \text{(by padding and } \mathsf{NE} \subseteq \mathsf{BPTIME}[n^k]\text{)}\\
&\subseteq \mathsf{i.\,o.\,BPTIME}[2^{n}]/_{2^{\varepsilon n}} && \text{(by choosing } \varepsilon \leq 1/k\text{)}\\
&\subseteq \mathsf{i.\,o.\,DTIME}\left[2^{2^{n}}\right]/_{2^{\varepsilon n}} && \text{(by deterministic simulation)}
\end{aligned}$$

Note that we use the assumption $\mathsf{NEXP} = \mathsf{BPP}$ *twice* in the above. Finally, one can show by diagonalization that $\mathsf{EE} \notin \mathsf{i.\,o.\,DTIME}\left[2^{2^{n}}\right]/_{2^{\varepsilon n}}$, which gives a contradiction as desired.

The proof that weak coding for nKt follows from $\mathsf{NEXP} \neq \mathsf{BPP}$ is similar to the previous case. We use the hardness-vs-randomness framework (and a win-win argument) to construct a PRG that "hits" any string x sampled with probability at least $1/\mathrm{poly}(n)$. However, there are a couple of differences in this setting. First, in the win-win argument, we use the Karp–Lipton result for NEXP [IKW02] instead of the one for EXP. Second, to obtain weak coding for nKt, we require our PRG to be computable *non-deterministically* in the sense that there exists some good guess w that allows us to correctly compute the output of the PRG, while for all other bad guesses, we output $\perp$. While we don't know how to achieve this exactly, we can show that it is possible *with access to a small advice string*. This is because one can non-deterministically construct the truth table of a language in NEXP using a small amount of advice that indicates the number of positive instances, as observed for instance in [IKW02]. Such a PRG is sufficient for our purposes.

Equivalence Between Coding for zKt and $\mathsf{prZPEXP} \neq \mathsf{prBPP}$. The task of obtaining $\mathsf{prZPEXP} \neq \mathsf{prBPP}$ from a non-trivial coding theorem for zKt faces the same challenge as in the case of showing $\mathsf{NEXP} \neq \mathsf{BPP}$ from a coding theorem for nKt, since it is unclear how to construct an N-bit string with high zKt-*complexity* in time $\mathrm{poly}(2^N)$. On the other hand, the alternative approach used to show the latter can also be applied in this context. However, when using this approach in the case of NEXP, it relied on the fact that NEXP is a *syntactic* class, whereas ZPEXP is not (i.e., it is *semantic*). To address this issue, we consider the weaker conclusion that $\mathsf{prZPEXP} \neq \mathsf{prBPP}$ instead of $\mathsf{ZPEXP} \neq \mathsf{BPP}$.

A bigger challenge arises in showing that weak coding for zKt follows from $\mathsf{prZPEXP} \neq \mathsf{prBPP}$. Recall that in previous cases, we needed to use a Karp–Lipton result for either EXP or NEXP. However, we do not have such a Karp–Lipton result for *zero-error probabilistic classes*. In fact, obtaining Karp–Lipton theorems for probabilistic classes is known to be a challenging task in complexity theory. While there are known results showing some weak versions of such a theorem for ZPEXP (see [OS17]), they are not sufficient for our purpose here.

Our key observation is that we only need the Karp–Lipton result in one of the cases in our win-win argument. Specifically, we can consider two cases: $\mathsf{EXP} \neq \mathsf{BPP}$, in which we have weak coding for Kt and hence for zKt, and $\mathsf{EXP} = \mathsf{BPP}$. We show that in the latter case, we can indeed obtain a Karp–Lipton theorem for zero-error probabilistic classes. More specifically, we show that assuming $\mathsf{EXP} = \mathsf{BPP}$, if $\mathsf{ZPE}/_n \subseteq \mathsf{SIZE}[n^k]$ for some $k > 0$, then $\mathsf{prZPEXP} = \mathsf{prEXP}$ (see Lemma 35).

Now assume $\mathsf{prZPEXP} \neq \mathsf{prBPP}$, and suppose we are in the remaining case $\mathsf{EXP} = \mathsf{BPP}$ (which is equivalent to $\mathsf{prEXP} = \mathsf{prBPP}$). We get that $\mathsf{prZPEXP} \neq \mathsf{prEXP}$. By our aforementioned Karp–Lipton theorem, we obtain that $\mathsf{ZPE}/_n \nsubseteq \mathsf{SIZE}[n^k]$ for all k. Again, using the hardness-vs-randomness framework, this allows us to obtain an infinitely-often secure PRG with sub-polynomial seed length that is computable *probabilistically with zero error* using a small amount of advice. Proceeding similarly to previous proofs, this yields weak coding for zKt, as desired.

Complexity Separations and Meta-Complexity. We first describe the proof of Theorem 4. As mentioned in Section 1.2.4, it was shown in [ABK+06] that $\mathsf{EXP} = \mathsf{BPP}$ if and only if $\mathsf{Gap\text{-}MKtP} \in \mathsf{prBPP}$. The original proof relied on the fact that EXP admits instance checkers [BFL91], which are not available for NEXP. Here, we provide an alternative proof that does not use instance checkers.

For the direction that $\mathsf{NEXP} = \mathsf{BPP}$ implies $\mathsf{Gap\text{-}MnKtP} \in \mathsf{prBPP}$, it is not hard to see that $\mathsf{MnKtP} \in \mathsf{PSPACE}^{\mathsf{NEXP}}$, where the queries to the NEXP oracle are of polynomial size. It is then not difficult to show that the desired inclusion follows from $\mathsf{NEXP} = \mathsf{BPP}$. Indeed, we get the stronger conclusion that $\mathsf{MnKtP} \in \mathsf{BPP}$.

For the other direction, assume $\mathsf{NEXP} \neq \mathsf{BPP}$. Then, as shown in previous paragraphs, we obtain an infinitely-often secure PRG that is computable non-deterministically with a small amount of advice. Now suppose, for the sake of contradiction, that $\mathsf{Gap\text{-}MnKtP} \in \mathsf{prBPP}$. In that case, an efficient algorithm solving $\mathsf{Gap\text{-}MnKtP}$ could be used to break the security of the aforementioned PRG. This is because every output of such a PRG has small nKt-complexity, while a uniformly random string has high nKt-complexity.

The proof of Theorem 5 for $\mathsf{Gap\text{-}MzKtP}$ can be shown similarly, using a PRG that is computable probabilistically with zero error using a small advice. Such a PRG can be obtained under the assumption that $\mathsf{prZPEXP} \neq \mathsf{prBPP}$, as described in previous paragraphs.

Finally, for our unconditional lower bound in Theorem 6, a natural approach is to try to adapt the lower bound for rKt in the two-sided error setting from [Oli19] to the zero-error setting. The proof makes crucial use of techniques from pseudorandomness and of the properties of the reconstruction procedure of different PRGs. In order to adapt the original argument to the zero-error setting, it is necessary to obtain zero-error reconstruction routines for the corresponding PRGs. This, however, seems to be out of reach using current techniques (see [LPT24] for related results).

Instead, we show that if $\mathsf{Gap\text{-}MzKtP}$ can be solved by a zero-error randomized algorithm in quasi-polynomial time, then, using the *easy witness method* introduced by [Kab00], one can approximately "collapse" rKt and zKt (Lemma 40). This, in particular, implies that $\mathsf{Gap\text{-}MrKtP}$ can also be solved in quasi-polynomial time by a randomized algorithm. Using the known *unconditional* lower bound for $\mathsf{Gap\text{-}MrKtP}$ established in [Oli19], this leads to a contradiction.

Acknowledgements. We would like to thank Hanlin Ren for discussions related to the problem of showing that $\mathsf{ZPEXP} \not\subseteq \mathsf{ZPP}/_{O(\log n)}$. This work received support from the UKRI Frontier Research Guarantee Grant EP/Y007999/1 and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick.

2 Preliminaries

2.1 Time-Bounded Kolmogorov Complexity

Fix a time-efficient universal Turing machine U. For convenience of the reader, we collect below the main notions of time-bounded Kolmogorov complexity considered in this work.

Definition 7 (Kt [Lev84])**.** For a string $x \in \{0,1\}^*$ and an oracle $\mathcal{O} \subseteq \{0,1\}^*$, we let

$$\mathsf{Kt}^{\mathcal{O}}(x) \triangleq \min_{p \in \{0,1\}^*,\, t \in \mathbb{N}} \Big\{ |p| + \lceil \log t \rceil : U^{\mathcal{O},t}(p) = x \Big\}.$$

The notation $U^{\mathcal{O},t}(p)$ denotes that U computes for at most t steps. In the absence of $\mathcal{O}$, we simply write $\mathsf{Kt}(x)$.

Definition 8 (rKt [Oli19]). For a string $x \in \{0,1\}^*$, we let

$$\mathsf{rKt}(x) \triangleq \min_{p \in \{0,1\}^*,\, t \in \mathbb{N}} \left\{ |p| + \lceil \log t \rceil \colon \Pr_r[U^t(p,r) = x] \geq 2/3 \right\}.$$

Next, we define zero-error and nondeterministic analogues of these measures.

Definition 9 (zKt). For a string $x \in \{0,1\}^*$, we let

$$\mathsf{zKt}(x) \triangleq \min_{p \in \{0,1\}^*,\, t \in \mathbb{N}} \left\{ |p| + \lceil \log t \rceil \;\middle|\; \Pr_r[U^t(p,r) = x] \geq 2/3 \text{ and } \forall r,\ U^t(p,r) \in \{x, \perp\} \right\}.$$

Definition 10 (nKt). For $x \in \{0,1\}^*$, we let

$$\mathsf{nKt}(x) \triangleq \min_{p \in \{0,1\}^*,\, t \in \mathbb{N}} \{ |p| + \lceil \log t \rceil \mid \exists w, U^t(p,w) = x \text{ and } \forall w,\ U^t(p,w) \in \{x, \perp\} \}.$$

Note that, for every $x \in \{0,1\}^*$, we have $\mathsf{nKt}(x) \leq \mathsf{zKt}(x) \leq \mathsf{Kt}(x)$ and $\mathsf{rKt}(x) \leq \mathsf{zKt}(x) \leq \mathsf{Kt}(x)$. The relation between $\mathsf{rKt}(x)$ and $\mathsf{nKt}(x)$ is unclear. For an overview of probabilistic notions of Kolmogorov complexity and their applications, we refer to [LO22].

2.2 Pseudorandomness

For a finite set A, we write $x \sim A$ to denote that x is uniformly distributed over A.

Let $\mathcal{D}_n$ be a distribution supported over $\{0,1\}^n$. Let $\varepsilon \in [0,1]$. Finally, let $f \colon \{0,1\}^n \to \{0,1\}$. We say that $\mathcal{D}_n$ *ε-fools f* if

$$\left| \Pr_{x \sim \mathcal{D}_n}[f(x) = 1] - \Pr_{x \sim \{0,1\}^n}[f(x) = 1] \right| \leq \varepsilon.$$

For a function $H \colon \{0,1\}^\ell \to \{0,1\}^m$, we write $H(-)$ to denote the distribution induced by $H(y)$ for $y \sim \{0,1\}^\ell$.

Theorem 11 ([BFNW93]). *For every $\varepsilon > 0$ and $b \in \mathbb{N}$, there exist a polynomial time computable function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, $\delta < \varepsilon$ and $c \in \mathbb{N}$ such that the following holds.*

$$F \colon \{0,1\}^{2^{n^\delta}} \times \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b},$$

and if T is the truth table of a Boolean function on n^δ variables that has circuit complexity at least $n^{c\delta}$, then the generator $G^T(-) \triangleq F(T,-)$ (n^{-b})-fool every circuit of size at most n^b.

Theorem 12 ([IW01, TV07]). *Assume $\mathsf{PSPACE} \neq \mathsf{BPP}$. Then for every $\varepsilon > 0$ and $b \in \mathbb{N}$, there is a sequence $\{G_n\}_{n \in \mathbb{N}}$, where $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b}$ is computable in time $2^{O(n^\varepsilon)}$, such that the following holds. For every distribution family $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ of Boolean circuits samplable in time n^b, there are infinitely many $n \in \mathbb{N}$ such that with probability at least $1 - n^{-b}$ over C sampled from $\mathcal{C}_n$, G_n (n^{-b})-fools C.*

2.3 Complexity Theory and Diagonalization Against Advice

For the definition of standard notions, such as complexity classes with advice and promise classes, we refer to a textbook in complexity theory.

The following simple diagonalization lemma will be sufficient for our purposes.

Lemma 13. *Let $a(n), b(n), c(n), s(n)$ be time-constructible functions satisfying the following properties:*

1. $b^2(n) \cdot 2^{3c(n)} \cdot s^3(n) = o(a(n))$,

2. $c(n) + \log s(n) < 2^n$,

3. $s(n) = \omega(1)$,

4. $b(n) = \Omega(n)$.

Then we have $\mathsf{DTIME}[a(n)] \not\subseteq \mathsf{i.o.DTIME}[b(n)]/_{c(n)}$.

Proof. We define a language as follows. For input length n, define $l(n) = \lfloor \log(s(n) \cdot 2^{c(n)}) \rfloor + 1$. Item 2 guarantees that $l(n) \le 2^n$. We construct the length-$l(n)$ prefix of truth tables of the first $s(n)$ Turing machines with all possible length-$c(n)$ advice strings running in time $b(n)$. There are at most $s(n) \cdot 2^{c(n)}$ such prefixes, and since $2^{l(n)} > s(n) \cdot 2^{c(n)}$, we can enumerate over all length-$l(n)$ strings, and find the first string p outside this list. We then define the truth table of this language on input length n as $p0^{2^n - l(n)}$.

The first enumeration and simulation step takes time $s(n) \cdot 2^{c(n)} \cdot l(n) \cdot b(n) \cdot \log b(n)$. Using a naive search over all $l(n)$-bit strings, finding p takes time at most $s(n) \cdot 2^{c(n)} \cdot l(n) \cdot 2^{l(n)}$. By Item 1 and Item 4, this language is decidable in time $a(n)$. However, by our construction and Item 3, any Turing machine running in time $b(n)$ fails to decide this language with any length-$c(n)$ advice string for all large enough n. □

Recall that $\mathsf{EE} \triangleq \mathsf{DTIME}[2^{2^{O(n)}}]$ denotes the class of languages that can be decided in double exponential time, $\mathsf{E} \triangleq \mathsf{DTIME}[2^{O(n)}]$ denotes the class of languages that can be decided in single exponential time, and $\mathsf{EXP} \triangleq \mathsf{DTIME}[2^{n^{O(1)}}]$.

Corollary 14. *For any fixed $k \in \mathbb{N}$ and time-constructible $s(n) = \omega(1)$,* $\mathsf{EE} \not\subseteq \mathsf{i.o.DTIME}[2^{2^{kn}}]/_{2^n - s(n)}$.

Corollary 15. *For any fixed $k \in \mathbb{N}$,* $\mathsf{EXP} \not\subseteq \mathsf{i.o.SIZE}[n^k]$.

For a language $L \subseteq \{0,1\}^*$, we let $\mathsf{tt}(L^{=n}) \in \{0,1\}^{2^n}$ denote the string representing the truth table of L on inputs of length n.

3 Coding for Deterministic Time-Bounded Kolmogorov Complexity

3.1 Equivalence Between Coding for Kt and $\mathsf{EXP} \neq \mathsf{BPP}$

Theorem 1. *The following statements are equivalent.*

1. $\mathsf{EXP} \neq \mathsf{BPP}$.

2. **(Weak coding for Kt.)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*
$$\mathsf{Kt}(x) \le \left(\frac{1}{\mathcal{D}_n(x)} \cdot n \right)^{\varepsilon}.$$

3. **(Non-trivial coding for Kt.)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n \in \mathbb{N}}$ such that $\mathcal{D}_n(x_n) \ge 1 - n^{-c}$ for every n. Then for infinitely many n, we have*
$$\mathsf{Kt}(x_n) \le n - \omega(\log n).$$

Proof. We show the following implications.
(Item 2 $\Longrightarrow$ Item 3). This holds trivially.
(Item 3 $\Longrightarrow$ Item 1). This is shown by Lemma 16, stated and proved in Section 3.1.1.
(Item 1 $\Longrightarrow$ Item 2). This follows from Lemma 18, stated and proved in Section 3.1.2. □

3.1.1 $\mathsf{EXP} \neq \mathsf{BPP}$ from Non-Trivial Coding for Kt

Lemma 16. **(Item 3 $\Rightarrow$ Item 1 in Theorem 1).** *If non-trivial coding for* Kt *is true, then* $\mathsf{EXP} \neq \mathsf{BPP}$.

Proof. For the sake of contradiction, suppose $\mathsf{EXP} = \mathsf{BPP}$. By a simple padding argument, this implies $\mathsf{EE} \subseteq \mathsf{BPE}$. Then it suffices to show the existence of a language $L_{\mathsf{hard}} \in \mathsf{EE}$ such that $L_{\mathsf{hard}} \notin \mathsf{BPE}$.

We first show the following claim.

Claim 17. *If non-trivial coding for* Kt *is true, then for every* $L \in \mathsf{BPE}$*, there are infinitely many* n *such that* $\mathsf{Kt}(\mathsf{tt}(L^{=n})) \leq 2^n - \omega(n)$.

Proof of Claim 17. Let $c > 0$ be the constant in the non-trivial coding theorem (Item 3 of Theorem 1).

Fix $L \in \mathsf{BPE}$. Let M be a $2^{O(cn)}$-time probabilistic Turing machine that computes L on each input of length n with error $\leq 2^{-n-cn}$. Such a machine can be obtained by using error reduction techniques.

Consider the distribution family $\mathcal{D} \triangleq \{\mathcal{D}_N\}$ where each $\mathcal{D}_N$ is defined by the following sampling procedure:

> On input 1^N, let $n \triangleq \lceil \log N \rceil$. Let S be the ordered set consisting of the lexicographically first N elements of $\{0,1\}^n$. For each $x \in S$ compute $b_x \triangleq M(x)$. Finally, output $\circ_{x \in S}\, b_x$, i.e., the concatenation of these bits.

Note that since M has exponentially small error for each input, by a union bound, we get that for every $N \in \mathbb{N}$, with probability at least $1 - 2^{-cn}$, $\mathcal{D}_N$ outputs the N-bit prefix of the truth table given by $L^{=n}$, i.e., $\mathsf{tt}(L^{=n})_{[1:N]}$, where $n = \lceil \log N \rceil$. Also note that $\mathcal{D}$ is polynomial-time samplable.

By applying non-trivial coding for Kt to $\mathcal{D}$, it follows that there are infinitely many N such that, for $n \triangleq \lceil \log N \rceil$,

$$\mathsf{Kt}\big(\mathsf{tt}(L^{=n})_{[1:N]}\big) \leq N - \omega(\log N) \leq N - \omega(n).$$

Fix any N such that the above holds, and let $(p, t) \in \{0,1\}^* \times \mathbb{N}$ be such that $|p| + \log t \leq N - \omega(n)$ and $U(p)$ outputs $\mathsf{tt}(L^{=n})_{[1:N]}$ within t steps. Consider the following procedure for generating $\mathsf{tt}(L^{=n})$.

> Given $(p, \mathsf{suffix} \triangleq \mathsf{tt}(L^{=n})_{[N+1,2^n]})$, we first run $U(p)$ to obtain $\mathsf{prefix} \triangleq \mathsf{tt}(L^{=n})_{[1:N]}$ and output $\mathsf{prefix} \circ \mathsf{suffix}$.

It is easy to see that the above procedure runs in time $t \cdot 2^{O(n)}$. This implies that

$$\begin{aligned}\mathsf{Kt}(\mathsf{tt}(L^{=n})) &\leq |p| + (2^n - N) + O(n) + \log\Big(t \cdot 2^{O(n)}\Big) \\ &\leq 2^n - \omega(n).\end{aligned}$$

This completes the proof of Claim 17. $\diamond$

We define the language L_{hard} as follows.

> On input $x \in \{0,1\}^n$, we first compute a string $T \in \{0,1\}^{2^n}$ such that $\mathsf{Kt}(T) > 2^n - 1$, as follows. We enumerate all pairs $(p, t) \in \{0,1\}^* \times \mathbb{N}$ such that $|p| + \lceil \log t \rceil \leq 2^n - 1$ and run $U(p)$ for at most t steps. This gives all the strings whose Kt-complexity are at most $2^n - 1$. We then let T be the lexicographically first 2^n-bit string that is not in the list. Finally, we output the x-th bit of T.

It is easy to see that $L_{\mathsf{hard}} \in \mathsf{EE}$. Also, by construction, we have that for all n, $\mathsf{Kt}(\mathsf{tt}(L_{\mathsf{hard}}^{=n})) > 2^n - 1$. It follows from Claim 17 that $L_{\mathsf{hard}} \notin \mathsf{BPE}$. $\square$

3.1.2 Weak Coding for Kt from $\mathsf{EXP} \neq \mathsf{BPP}$

Lemma 18. (Item 1 $\Rightarrow$ Item 2 in Theorem 1). *If $\mathsf{EXP} \neq \mathsf{BPP}$, then weak coding for Kt is true.*

We first show the following technical lemma.

Lemma 19. *If $\mathsf{EXP} \neq \mathsf{BPP}$, then for every $\varepsilon > 0$ and $b \in \mathbb{N}$, there is a sequence $\{G_n\}_{n\in\mathbb{N}}$, where $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b}$ is computable in time $2^{O(n^\varepsilon)}$, such that the following holds. For every distribution family $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ of Boolean circuits samplable in time n^b, there are infinitely many $n \in \mathbb{N}$ such that with probability at least $1 - n^{-b}$ over C sampled from $\mathcal{C}_n$, G_n (n^{-b})-fools C.*

Proof. Assume $\mathsf{EXP} \neq \mathsf{BPP}$. We consider two cases and show that the desired conclusion holds in each one of those cases.

Case 1: $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$. The desired pseudorandom generator follows directly from Theorem 12.

Case 2: $\mathsf{PSPACE} \subseteq \mathsf{BPP}$. Since we assume $\mathsf{EXP} \not\subseteq \mathsf{BPP}$, we have $\mathsf{EXP} \neq \mathsf{PSPACE}$ in this case. Recall that if $\mathsf{EXP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$ then $\mathsf{EXP} = \mathsf{PSPACE}$ [KL80]. Therefore, we have $\mathsf{EXP} \not\subseteq \mathsf{SIZE}[\mathsf{poly}]$, which further implies $\mathsf{E} \not\subseteq \mathsf{SIZE}[\mathsf{poly}]$. Let $L \in \mathsf{E}$ be a language that is not computable by any polynomial-size circuit.

Consider any $0 < \varepsilon < 1$ and $b \in \mathbb{N}$. Let F, $\delta < \varepsilon$ and $c \in \mathbb{N}$ be as provided by Theorem 11. By the property of F and the hardness of the language L, we have that, for infinitely many n, the generator $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^b$, defined as

$$G_n(-) \triangleq F\Big(\mathsf{tt}(L^{=n^\delta}), -\Big),$$

(n^{-b})-fools circuits of size at most n^b. Note that since $L \in \mathsf{E}$, $\mathsf{tt}(L^{=n^\delta})$ can be obtained in time $2^{O(n^\delta)}$. Also, F is polynomial-time computable. It follows that each G_n can be computed in time $2^{O(n^\varepsilon)}$. Finally, note that the above also yields the desired conclusion. $\square$

We are now ready to show Lemma 18.

Proof of Lemma 18. Assume $\mathsf{EXP} \neq \mathsf{BPP}$. The main idea is to use the pseudorandom generator G in Lemma 19 to "hit" any string x that is sampled with probability at least $1/\mathsf{poly}(n)$. That is, there is a seed $z \in \{0,1\}^{n^\varepsilon}$ such that $A(G(z)) = x$. Then x can be encoded using the short seed z. Details follow.

Let $\varepsilon > 0$ and $\{\mathcal{D}_n\}$ be a distribution family that admits a sampler A that, on input 1^n, runs in time at most n^c, for some constant $c \geq 1$.

Let $\{G_n \colon \{0,1\}^{n^{\varepsilon/2}} \to \{0,1\}^{n^b}\}$ be the sequence of generators in Theorem 12, where $b > c/\varepsilon$ is a constant specified later.

Consider the following distribution $\{\mathcal{C}_n\}$ of circuits:

> On input 1^n, we run $A(1^n)$ to obtain a string x. We then construct the circuit C_x such that $C_x(r) = 1$ if and only if $A(1^n; r) = x$. Finally, we output C_x.

First of all, note that by letting b be a sufficiently large constant, we get that $\{\mathcal{C}_n\}$ is samplable in time n^b. Then by Theorem 12 and the security of $\{G_n\}$, there are infinitely many n such that

$$\Pr_{C\sim\mathcal{C}_n}\Big[G_n\ (n^{-b})\text{-fools } C\Big] \geq 1 - n^{-b}. \tag{1}$$

Now fix any large enough n such that Equation (1) holds and consider any x in the support of D_n. Suppose $\mathcal{D}_n(x) < n^{-c/\varepsilon}$. Then we have

$$\mathsf{Kt}(x) \leq 2 \cdot n^c \leq \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon},$$

as desired.

Suppose $\mathcal{D}_n(x) \geq n^{-c/\varepsilon}$. Then by construction, we have that $\mathcal{C}_n$ samples C_x with probability at least $n^{-c/\varepsilon} > n^{-b}$. It follows from Equation (1) that G_n (n^{-b})-fools C_x; this is because otherwise the probability that G_n fails to be pseudorandom would be greater than n^{-b}. In particular, this means

$$\begin{aligned}\mathop{\mathbf{Pr}}_{z\sim\{0,1\}^{n^{\varepsilon/2}}}[C_x(G_n(z)) = 1] &\geq \mathop{\mathbf{Pr}}_{r\sim\{0,1\}^{n^b}}[C_x(r) = 1] - n^{-b} \\ &\geq n^{-c/\varepsilon} - n^{-b} > 0.\end{aligned}$$

It follows that there exists some $z \in \{0,1\}^{n^{\varepsilon/2}}$ such that $A(1^n; G_n(z)) = x$. From here, it is easy to show that $\mathsf{Kt}(x) \leq n^{\varepsilon}$, as desired. $\square$

3.2 Stronger Lower Bounds from Near-Optimal Coding for Kt

We say that near-optimal coding for Kt holds if for every polynomial-time sampler $A(1^n)$ and for every string $x \in \{0,1\}^n$, if x has probability $\geq \delta$ under $A(1^n)$ then $\mathsf{Kt}(x) = O(\log(1/\delta) + \log n)$.

Theorem 20. *Suppose that near-optimal coding for* Kt *holds. Then, for every* $c \geq 1$ *there is* $k \geq 1$ *and a language* $L \in \mathsf{DTIME}[2^{kn}]$ *such that* $L \notin \mathsf{i.o.BPTIME}[2^{cn}]$.

Proof. Fix a constant $c \geq 1$. We define a sampler $A(1^N)$ with $N \triangleq 2^n$ that randomly selects one of the first $\alpha(N) \triangleq \log\log N$ randomized Turing machines, runs it for 2^{2cn} steps on every string of length n, and outputs the corresponding truth table. We also assume that $A(1^N)$ boosts the success probability of the machine on a given input string by simulating it n^2 times and taking a majority vote, meaning that once a machine with bounded acceptance probabilities is selected, the corresponding truth table is produced with probability at least $9/10$.

Note that for every language $L' \in \mathsf{BPTIME}[2^{cn}]$ and for each large enough n, the truth table of L' on inputs of length n is output by $A(1^N)$ with probability at least $\delta \triangleq (9/10) \cdot (1/\log\log N) = \Omega(1/\log n)$. Consequently, by the near-optimal coding assumption, every truth table in $\mathsf{BPTIME}[2^{cn}]$ (a string of length $N = 2^n$) has Kt complexity at most $O(\log(1/\delta) + \log N) \leq c_1 \cdot n$, for a large enough constant c_1.

Finally, we can define a hard language $L \in \mathsf{DTIME}[2^{kn}]$ as follows. On an input string of length n, we find by diagonalization a string of length 2^n of Kt complexity $\geq c_2 n$, for $c_2 > c_1$, and compute according to the truth table encoded by this string. The latter can be done by exhaustive search in deterministic time 2^{kn}, for a large enough positive integer $k > c_2$. By the previous paragraph, we obtain that $L \notin \mathsf{BPTIME}[2^{cn}]$, which completes the proof. $\square$

We note that the elementary proof given above strengthens and simplifies [Lee06, Theorem 5.3.4].

4 Coding for Non-Deterministic Time-Bounded Kolmogorov Complexity

4.1 Equivalence Between Coding for nKt and $\mathsf{NEXP} \neq \mathsf{BPP}$

Theorem 21. *The following statements are equivalent.*

1. $\mathsf{NEXP} \neq \mathsf{BPP}$.

2. **(Weak coding for nKt)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*

$$\mathsf{nKt}(x) \leq \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}.$$

3. **(Non-trivial coding for nKt.)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $\mathcal{D}_n(x_n) \geq 1 - n^{-c}$ for every n. Then for infinitely many n, we have*

$$\mathsf{nKt}(x_n) \leq n - \omega(\log n).$$

Proof. We establish the following implications:
(Item 2 $\Longrightarrow$ Item 3). This follows immediately.
(Item 3 $\Longrightarrow$ Item 1). This is shown by Lemma 22, which is stated and proved in Section 4.1.1.
(Item 1 $\Longrightarrow$ Item 2). This follows from Lemma 25, which is stated and proved in Section 4.1.2. □

4.1.1 $\mathsf{NEXP} \neq \mathsf{BPP}$ from Non-Trivial Coding for nKt

Lemma 22. (Item 3 $\Rightarrow$ Item 1 in Theorem 21). *If non-trivial coding for nKt is true, then $\mathsf{NEXP} \neq \mathsf{BPP}$.*

Proof. We first show the following two claims.

Claim 23. *If non-trivial coding for nKt is true, then for every $L \in \mathsf{BPE}$, there are infinitely many n such that $\mathsf{nKt}(\mathsf{tt}(L^{=n})) \leq 2^n - \omega(n)$.*

Proof Sketch of Claim 23. The proof can be easily adapted from that of Claim 17, by replacing the use of non-trivial coding for Kt with that for nKt. ◇

Claim 24. *If non-trivial coding for nKt is true, then*

$$\mathsf{BPE} \subseteq \mathsf{i.o.NTIME}\left[2^{2^n - \omega(n)}\right]/_{2^n - \omega(n)}.$$

Proof of Claim 24. Fix $L \in \mathsf{BPE}$. First of all, by Claim 23, we have that there are infinitely many n such that $\mathsf{nKt}(\mathsf{tt}(L^{=n})) \leq 2^n - \omega(n)$. This means for infinitely many n, there exist a program p of size at most $2^n - \omega(n)$ such that for $t \triangleq 2^{2^n - \omega(n)}$,

- $\exists w \in \{0,1\}^t$, $U(p, w)$ outputs $\mathsf{tt}(L^{=n})$ within t steps, and
- $\forall w \in \{0,1\}^t$, $U(p, w)$ outputs $\mathsf{tt}(L^{=n})$ or $\perp$ within t steps

It is easy to see that for any n such that the above holds, given p as an advice, L on input length n can be solved non-deterministically in time $2^{2^n - \omega(n)}$, by guessing $w \in \{0,1\}^t$ and trying to use p to generate $\mathsf{tt}(L^{=n})$. ◇

We are now ready to show the lemma. Suppose

$$\mathsf{NEXP} = \mathsf{BPP}. \tag{2}$$

Note that by the existence of languages that are NE-complete under linear-time reductions, the above implies that there exists some $k > 0$ such that

$$\mathsf{NE} \subseteq \mathsf{BPTIME}[n^k]. \tag{3}$$

Next, we aim to derive a contradiction. By Equation (2) and padding, we have

$$\mathsf{EE} \subseteq \mathsf{BPE}. \tag{4}$$

By Claim 24, we get

$$\mathsf{BPE} \subseteq \text{i.o.}\mathsf{NTIME}\left[2^{2^n}\right]/_{2^n-\omega(n)}. \tag{5}$$

Now by using Equation (3) and a standard padding argument that incorporates the advice as an extra input string, we get that there exists some $k' > 0$ such that

$$\mathsf{NTIME}\left[2^{2^n}\right]/_{2^n-\omega(n)} \subseteq \mathsf{BPTIME}\left[2^{k'n}\right]/_{2^n-\omega(n)}. \tag{6}$$

Finally, by deterministic simulation of randomized algorithms, we get that there exists some $k > 0$ such that

$$\mathsf{BPTIME}\left[2^{k'n}\right]/_{2^n-\omega(n)} \subseteq \mathsf{DTIME}\left[2^{2^{kn}}\right]/_{2^n-\omega(n)}. \tag{7}$$

Equations (4) to (7) yield the existence of some $k > 0$ such that

$$\mathsf{EE} \subseteq \text{i.o.}\mathsf{DTIME}\left[2^{2^{kn}}\right]/_{2^n-\omega(n)}$$

However, the above contradicts Corollary 14. □

4.1.2 Weak Coding for nKt from $\mathsf{NEXP} \neq \mathsf{BPP}$

Lemma 25. (Item 1 ⇒ Item 2 in Theorem 21). *If $\mathsf{NEXP} \neq \mathsf{BPP}$, then weak coding for nKt is true.*

We rely on the following result, which is analogous to Lemma 19 but for the case of $\mathsf{NEXP} \neq \mathsf{BPP}$.

Lemma 26. *If $\mathsf{NEXP} \neq \mathsf{BPP}$, then for every $\varepsilon > 0$ and $b \in \mathbb{N}$, there is a sequence $\{G_n\}_{n\in\mathbb{N}}$, where $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b}$, such that the following holds. For every distribution family $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ of Boolean circuits samplable in time n^b, there are infinitely many $n \in \mathbb{N}$ such that with probability at least $1 - n^{-b}$ over C sampled from $\mathcal{C}_n$, G_n (n^{-b})-fools C. Moreover, each G_n can be computed non-deterministically with advice in the following sense: There exists a deterministic Turing machine M and a sequence of advice strings $a_n \in \{0,1\}^{n^\varepsilon}$ such that, given $z \in \{0,1\}^{n^\varepsilon}$ and $w \in \{0,1\}^{2^{n^\varepsilon}}$, $M(z, w; a_n)$ runs in time $2^{O(n^\varepsilon)}$. Also, for every $z \in \{0,1\}^{n^\varepsilon}$, the following hold:*

- *There exists $w \in \{0,1\}^{2^{n^\varepsilon}}$ such that $M(z, w; a_n) = G_n(z)$.*
- *For all $w \in \{0,1\}^{2^{n^\varepsilon}}$, $M(z, w; a_n) \in \{G_n(z), \perp\}$.*

Proof. The proof is similar to that of Lemma 26 but requires some crucial observations on the efficiency of computing the pseudorandom generator in the non-deterministic setting.

Assume $\mathsf{NEXP} \neq \mathsf{BPP}$. We consider two cases below.

Case 1: $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$. The desired pseudorandom generator follows directly from Theorem 12.

Case 2: $\mathsf{PSPACE} \subseteq \mathsf{BPP}$**.** By the assumption that $\mathsf{NEXP} \neq \mathsf{BPP}$, we get that $\mathsf{NEXP} \neq \mathsf{PSPACE}$ in this case. Then, by [IKW02], $\mathsf{NEXP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$ implies $\mathsf{NEXP} = \mathsf{MA} = \mathsf{PSPACE}$. Therefore, we obtain $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}[\mathsf{poly}]$ in this case.

Analogous to the proof of Lemma 26, given that $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}[\mathsf{poly}]$, we have a language $L \in \mathsf{NE}$ that is not computable by any polynomial-size circuit. Then for every $\varepsilon > 0$ and $b \in \mathbb{N}$, we get that the generator $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^b$, defined as $G_n(-) \triangleq F\Big(\mathsf{tt}(L^{=n^\delta}), -\Big)$, where F and $\delta > 0$ are as provided by Theorem 11, fools circuits of size at most n^b, for infinitely many n.

To show that each G_n can be computed non-deterministically with a small advice, it suffices to generate $\mathsf{tt}(L^{=n^\delta})$ in non-deterministic time $2^{O(n^\delta)}$ with n^δ bits of advice. This can be done, as observed in [IKW02, Lemma 1]. □

We now show Lemma 25.

Proof Sketch of Lemma 25. If $\mathsf{NEXP} \neq \mathsf{BPP}$, then by Lemma 26, we have an infinitely-often secure pseudorandom generator that is computable non-deterministically with a small advice string. Using an argument similar to the proof of Lemma 18, such a generator can be used to achieve weak coding for nKt. □

4.2 Equivalence Between Coding for $\mathsf{Kt}^{\mathsf{NP}}$ and $\mathsf{EXP}^{\mathsf{NP}} \neq \mathsf{BPP}$

Theorem 27. *The following statements are equivalent.*

1. $\mathsf{EXP}^{\mathsf{NP}} \neq \mathsf{BPP}$.
2. **(Weak coding for $\mathsf{Kt}^{\mathsf{NP}}$)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*
$$\mathsf{Kt}^{\mathsf{NP}}(x) \leq \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}.$$
3. **(Non-trivial coding for $\mathsf{Kt}^{\mathsf{NP}}$.)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $\mathcal{D}_n(x_n) \geq 1 - n^{-c}$ for every n. Then for infinitely many n, we have*
$$\mathsf{Kt}^{\mathsf{NP}}(x_n) \leq n - \omega(\log n).$$

Proof Sketch. The proof can be easily adapted from that of Theorem 1.

One difference arises in showing that $\mathsf{EXP}^{\mathsf{NP}} \neq \mathsf{BPP}$ implies weak coding for $\mathsf{Kt}^{\mathsf{NP}}$. Analogously to Lemma 19, we can show that if $\mathsf{EXP}^{\mathsf{NP}} \neq \mathsf{BPP}$, then for every $\varepsilon > 0$ and $b \in \mathbb{N}$, there exists an infinitely-often secure pseudorandom generator $G \triangleq \{G_n\}_{n\in\mathbb{N}}$, where each $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b}$ is computable in time $2^{O(n^\varepsilon)}$ *with access to an* NP *oracle*. Instead of using the Karp–Lipton theorem for EXP as in the proof of Lemma 19, we use the version for $\mathsf{EXP}^{\mathsf{NP}}$ [BH92], which states that if $\mathsf{EXP}^{\mathsf{NP}} \subseteq \mathsf{SIZE}[\mathsf{poly}]$, then $\mathsf{EXP}^{\mathsf{NP}} = \mathsf{PSPACE}$. □

5 Coding for Zero-Error Time-Bounded Kolmogorov Complexity

5.1 Equivalence Between Coding for zKt and $\mathsf{prZPEXP} \neq \mathsf{prBPP}$

Theorem 3. *The following statements are equivalent.*

1. $\mathsf{prZPEXP} \neq \mathsf{prBPP}$.

2. **(Weak coding for zKt)** *For any $\varepsilon > 0$ and any polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there are infinitely many $n \in \mathbb{N}$ such that for all $x \in \mathsf{Support}(\mathcal{D}_n)$,*

$$\mathsf{zKt}(x) \leq \left(\frac{1}{\mathcal{D}_n(x)} \cdot n\right)^{\varepsilon}.$$

3. **(Non-trivial coding for zKt)** *There exists a constant $c > 0$ such that the following holds. Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, satisfying that there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $\mathcal{D}_n(x_n) \geq 1 - n^{-c}$ for every n. Then for infinitely many n, we have*

$$\mathsf{zKt}(x_n) \leq n - \omega(\log n).$$

Proof. We present the following implications.
(Item 2 $\Longrightarrow$ Item 3). This holds trivially.
(Item 3 $\Longrightarrow$ Item 1). This is established by Lemma 28, stated and proved in Section 5.1.1.
(Item 1 $\Longrightarrow$ Item 2). This follows from Lemma 33, stated and proved in Section 5.1.2. □

5.1.1 $\mathsf{prZPEXP} \neq \mathsf{prBPP}$ from Non-Trivial Coding for zKt

Lemma 28. **(Item 3 ⇒ Item 1 in Theorem 3).** *If weak coding for zKt is true, then $\mathsf{prZPEXP} \neq \mathsf{prBPP}$.*

We need the following ingredients in our proof:

Claim 29. *If non-trivial coding for zKt is true, then for every $L \in \mathsf{BPE}$, there are infinitely many n such that $\mathsf{zKt}(\mathsf{tt}(L^{=n})) \leq 2^n - \omega(n)$.*

Proof Sketch of Claim 29. The proof can be easily adapted from that of Claim 17, by replacing the use of non-trivial coding for Kt with that for zKt. ◇

By the definition of zKt, we immediately have the following corollary:

Corollary 30. *If non-trivial coding for zKt is true, then*

$$\mathsf{BPE} \subseteq \mathsf{i.o.ZPTIME}[2^{2^n - \omega(n)}]/_{2^n - \omega(n)}.$$

Lemma 31. *If $\mathsf{prZPE} \subseteq \mathsf{prBPP}$, then there exists some $k > 0$, such that $\mathsf{prZPE} \subseteq \mathsf{prBPTIME}[n^k]$.*

Proof. We use the fact that there exists a "complete" problem for prZPE. More specifically, we define the promise problem $\Pi \triangleq (\mathcal{YES}, \mathcal{NO})$ as

$$\mathcal{YES} \triangleq \left\{(M, x, 1^t) \;\middle|\; M(x) \in \{1, \perp\} \wedge \mathbf{Pr}[M^{\leq 2^t}(x) = 1] \geq \frac{2}{3}\right\},$$
$$\mathcal{NO} \triangleq \left\{(M, x, 1^t) \;\middle|\; M(x) \in \{0, \perp\} \wedge \mathbf{Pr}[M^{\leq 2^t}(x) = 0] \geq \frac{2}{3}\right\},$$

where M is a randomized machine, and the notation $M^{\leq 2^t}$ denotes that we run it for at most 2^t steps. One can see that $\Pi \in \mathsf{prZPE}$. By our assumption, $\Pi \in \mathsf{prBPP}$, therefore there exists some k such that $\Pi \in \mathsf{prBPTIME}[n^k]$. Notice that each problem in prZPE can be reduced to Π in linear time, implying $\mathsf{prZPE} \subseteq \mathsf{prBPTIME}[n^k]$. □

Claim 32. *If* prZPE $\subseteq$ prBPP*, then there exists some* $k \in \mathbb{N}$ *such that for any time-constructible* $a(n)$*, we have* $\mathsf{ZPTIME}[2^{a(n)}]/_{a(n)} \subseteq \mathsf{BPTIME}[(n+a(n))^k]/_{a(n)}$.

Proof. Fix $L \in \mathsf{ZPTIME}[2^{a(n)}]/_{a(n)}$, and let $M(x, r, s)$ be the corresponding Turing machine, where x is the input, r is the random string and s is the advice. We define a promise problem $\Pi \triangleq (\mathcal{YES}, \mathcal{NO})$ as follows:

$$\mathcal{YES} \triangleq \left\{ (x, s) \;\middle|\; \forall r, M(x, r, s) \in \{1, \bot\} \wedge \Pr_r[M(x, r, s) = 1] \geq \frac{2}{3} \right\},$$
$$\mathcal{NO} \triangleq \left\{ (x, s) \;\middle|\; \forall r, M(x, r, s) \in \{0, \bot\} \wedge \Pr_r[M(x, r, s) = 0] \geq \frac{2}{3} \right\}.$$

By definition, Π is in prZPE. By our assumption and Lemma 31, there exists some k such that $\Pi \in \mathsf{prBPTIME}[n^k]$. Let $M'(x, r, s)$ be the Turing machine witnessing such inclusion. Let $s_n \in \{0,1\}^{a(n)}$ be the advice string for M on input length n. Then we have

$$x \in L \to (x, s_n) \in \mathcal{YES} \to \Pr_r[M'(x, r, s_n) = 1] \geq \frac{2}{3},$$
$$x \notin L \to (x, s_n) \in \mathcal{NO} \to \Pr_r[M'(x, r, s_n) = 0] \geq \frac{2}{3}.$$

Since M' runs in time m^k, where $m = (n + a(n))$ is the length of (x, s_n), we conclude that $L \in \mathsf{BPTIME}[(n + a(n))^k]/_{a(n)}$. We finish our proof by observing that k is independent of $a(n)$ and L. $\diamond$

Proof of Lemma 28. For the sake of contradiction, assume non-trivial coding for zKt is true and prZPEXP = prBPP. Our assumption implies EXP $\subseteq$ BPP. By a padding argument, we have

$$\mathsf{EE} \subseteq \mathsf{BPE}.$$

By Corollary 30,

$$\mathsf{BPE} \subseteq \mathsf{i.o.ZPTIME}[2^{2^n - \omega(n)}]/_{2^n - \omega(n)}.$$

By Claim 32,

$$\mathsf{ZPTIME}[2^{2^{n-\omega(n)}}]/_{2^n-\omega(n)} \subseteq \mathsf{BPTIME}[2^{kn}]/_{2^n-\omega(n)} \subseteq \mathsf{DTIME}[2^{2^{kn}}]/_{2^n-\omega(n)}.$$

Combining these three inclusions gives us

$$\mathsf{EE} \subseteq \mathsf{i.o.DTIME}[2^{2^{kn}}]/_{2^n-\omega(n)}.$$

However, this contradicts the diagonalization result of Corollary 14. $\square$

5.1.2 Weak Coding for zKt from prZPEXP $\neq$ prBPP

Lemma 33. (Item 1 $\Rightarrow$ Item 2 in Theorem 1). *If* prZPEXP $\neq$ prBPP*, then weak coding for* zKt *is true.*

To prove Lemma 33, we need the following tools:

Lemma 34. *If* prZPEXP $\not\subseteq$ prEXP*, then for any* $\varepsilon > 0$*,* $\mathsf{BPP} \subseteq \mathsf{i.o.ZPTIME}[2^{n^\varepsilon}]/_{n^\varepsilon}$.

Proof. The proof relies on the easy witness method introduced by [Kab00].

We claim that it suffices to show the following statement:

If $\mathsf{prZPEXP} \not\subseteq \mathsf{prEXP}$, then for any $c \in \mathbb{N}$, there exists a Turing machine A, satisfying the following conditions:

1. For $r \in \{0,1\}^{2n}$ and $s \in \{0,1\}^n$, $A(r,s)$ runs in $2^{O(n)}$ steps using r as random string and s as advice and it either outputs some string in $\{0,1\}^{2^n}$ or $\perp$.
2. For infinitely many n, there exists some advice s_n such that $\mathbf{Pr}_r[A(r,s_n) = \perp] \leq 1/3$, and for every r such that $A(r,s_n) \neq \perp$, $A(r,s_n)$ is the truth table of an n-variable Boolean function with circuit complexity at least n^c.

In fact, if we have such a machine A, then for any $0 < \delta < \varepsilon$ and $c \in \mathbb{N}$, for infinitely many n, using n^δ bits of advice, A can generate the truth table of some n^δ-variable Boolean function with circuit complexity at least $n^{c\delta}$, succeeding with high probability with zero error. We can then plug this hard truth table into the generator of Theorem 11 to derandomize BPP. Since $n^\delta < n^\varepsilon$, $\mathsf{BPP} \subseteq \mathsf{i.o.ZPTIME}[2^{n^\varepsilon}]/_{n^\varepsilon}$. All that remains is to prove the above statement.

First, observe that our assumption implies $\mathsf{prZPTIME}[2^n] \not\subseteq \mathsf{prEXP}$, because otherwise a simple padding argument gives $\mathsf{prZPEXP} \subseteq \mathsf{prEXP}$. Let $\Pi \triangleq (\mathcal{YES}, \mathcal{NO})$ be a promise problem in $\mathsf{prZPTIME}[2^n] \backslash \mathsf{prEXP}$, and let $M(x,r)$ be the Turing machine witnessing such inclusion, where $x \in \{0,1\}^n$ is the input and $r \in \{0,1\}^{2^n}$ is the random string. We define a Turing machine $B(x)$ that "tries to derandomize M" as follows:

> On input $x \in \{0,1\}^n$, enumerate over all n-variable Boolean circuits C of size at most n^c, and compute $M(z, \mathsf{tt}(C))$, where $\mathsf{tt}(C) \in \{0,1\}^{2^n}$ is the string representing the truth table of circuit C. If all runs of M outputs $\perp$, then B outputs $\perp$; otherwise if $b \in \{0,1\}$ appeared as the output of one of the runs, B outputs b. (If both 0 and 1 appeared, B outputs arbitrarily.)

One can see that B runs in deterministic time $2^{n^{2c}}$. So by our assumption that $\Pi \notin \mathsf{prEXP}$, B cannot compute Π. In other words, there are infinitely many input lengths n where there exists some $x_n \in \mathcal{YES}_n \cup \mathcal{NO}_n$ such that $B(x_n) \neq \Pi(x_n)$. Since M makes zero error on the promised inputs, $B(x_n) = \perp$. By definition of B, for any r, if r is the truth table of an n-variable Boolean circuit of size at most n^c, then $M(x_n, r) = \perp$. Taking the contrapositive, if $M(x_n, r) \neq \perp$, then r is the truth table of an n-variable Boolean function with circuit complexity at least n^c. But if we sample r uniformly from $\{0,1\}^{2^n}$, then $\mathbf{Pr}_r[M(x_n,r) \neq \perp] \geq 2/3$. Hence we define the Turing machine A as follows:

> Given x_n as advice, A samples r from $\{0,1\}^{2^n}$ uniformly. If $M(x_n,r) \neq \perp$, then A outputs r; otherwise A outputs $\perp$.

It is not hard to see that A satisfies the two conditions stated above. □

We show the following Karp–Lipton theorem for zero-error probabilistic classes, under the assumption that $\mathsf{EXP} = \mathsf{BPP}$.

Lemma 35. *Suppose* $\mathsf{EXP} = \mathsf{BPP}$. *If* $\mathsf{ZPE}/_n \subseteq \mathsf{SIZE}[n^k]$ *for some* k*, then* $\mathsf{prZPEXP} = \mathsf{prEXP}$.

Proof. Assume $\mathsf{EXP} = \mathsf{BPP}$. For the sake of contradiction, suppose $\mathsf{ZPE}/_n \subseteq \mathsf{SIZE}[n^k]$ for some $k > 0$ and $\mathsf{prZPEXP} \neq \mathsf{prEXP}$. By Lemma 34, the latter implies that $\mathsf{BPP} \subseteq \mathsf{i.o.ZPTIME}[2^{n^\varepsilon}]/_{n^\varepsilon}$ for any $\varepsilon > 0$. Then we have $\mathsf{EXP} \subseteq \mathsf{BPP} \subseteq \mathsf{i.o.ZPE}/_n \subseteq \mathsf{i.o.SIZE}[n^k]$, which contradicts Corollary 15. □

Finally, we need the following analogue of Lemma 19, which gives an infinitely-often secure pseudo-random generator under the assumption that $\mathsf{prZPEXP} \neq \mathsf{prBPP}$.

Lemma 36. *If* prZPEXP $\neq$ prBPP*, then for every* $\varepsilon > 0$ *and* $b \in \mathbb{N}$*, there is a sequence* $\{G_n\}_{n\in\mathbb{N}}$*, where* $G_n\colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b}$*, such that the following holds. For every distribution family* $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ *of Boolean circuits samplable in time* n^b*, there are infinitely many* $n \in \mathbb{N}$ *such that with probability at least* $1 - n^{-b}$ *over* C *sampled from* $\mathcal{C}_n$*,* G_n (n^{-b})*-fools* C*. Moreover, each* G_n *can be computed probabilistically with zero error using a small advice in the following sense: There exists a deterministic Turing machine* M *and a sequence of advice strings* $a_n \in \{0,1\}^{n^\varepsilon}$ *such that, given* $z \in \{0,1\}^{n^\varepsilon}$ *and* $w \in \{0,1\}^{2^{n^\varepsilon}}$*,* $M(z, w; a_n)$ *runs in time* $2^{O(n^\delta)}$*. Also, for every* $z \in \{0,1\}^{n^\varepsilon}$*, the following hold:*

- $\mathbf{Pr}_{w\sim\{0,1\}^{2^{n^\varepsilon}}}[M(z, w; a_n) = G_n(z)] \geq 2/3.$
- *For all* $w \in \{0,1\}^{2^{n^\varepsilon}}$*,* $M(z, w; a_n) \in \{G_n(z), \perp\}$*.*

Proof. Assume prZPEXP $\neq$ prBPP. We consider two cases below.

Case 1: EXP $\neq$ BPP**.** The desired pseudorandom generator follows directly from Lemma 19.

Case 2: EXP $=$ BPP**.** Note that EXP $=$ BPP implies prEXP $=$ prBPP. Since we assume prZPEXP $\neq$ prBPP, it follows that prZPEXP $\neq$ prEXP. By Lemma 35, we have ZPE$/_n$ $\not\subseteq$ SIZE$[n^k]$ for every k. Then we can plug the hard truth table of some language in ZPE$/_n$ into the function F of Theorem 11 to get a pseudorandom generator. More specifically, for every $\varepsilon > 0$ and $b \in \mathbb{N}$, we consider the generator $G_n\colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^b$, defined as $G_n(-) \triangleq F\Big(\mathsf{tt}(L^{=n^\delta}), -\Big)$, where F and $\delta < \varepsilon$ are provided by Theorem 11, and L is a language in ZPE with linear advice that has circuit complexity at least $n^{c\delta}$, for some constant $c > 0$ specified in Theorem 11.

Finally, note that since $L \in$ ZPE$/_n$, $\mathsf{tt}(L^{=n^\delta})$ can be computed probabilistically with zero error in time $2^{O(n^\delta)}$ with n^δ bits of advice. $\square$

We are now ready to show Lemma 33.

Proof of Lemma 33. If prZPEXP $\neq$ prBPP, then by Lemma 36, we have an infinitely-often pseudorandom generator that is computable probabilistically with zero error using a small advice. Using a similar argument as in the proof of Lemma 18, such a generator can be used to achieve weak coding for zKt. $\square$

5.2 On Coding for zKt and ZPEXP $\neq$ BPP

Theorem 37. *Assume that* ZPEXP $\not\subseteq$ ZPP$/_{O(\log n)}$*. Then a near-optimal coding theorem for* zKt *implies that* ZPEXP $\not\subseteq$ BPP*.*

Proof. For the sake of contradiction, assume that ZPEXP $\subseteq$ BPP. Our goal is to derandomize BPP to ZPP$/_{O(\log n)}$. This contradicts the initial assumption that ZPEXP $\not\subseteq$ ZPP$/_{O(\log n)}$, which concludes the proof.

To achieve our goal, we adapt the proofs of Lemma 5.3.3 and Theorem 5.3.4 of [Lee06]. Let $L \in$ BPP, and let M be a randomized TM deciding L. Assume that on input length n, M uses m random bits for some $m > n$. We do error reduction over M, such that the resulting TM M' uses m^3 random bits, and the probability of error on a given input string is $O(2^{-m^2})$. If r is a string that leads to a wrong answer for x, then by Theorem 4.1.4 of [Lee06] for some polynomial p we have $\mathsf{CN}^p(r|x) \leq m^3 - O(m^2)$. Here $\mathsf{CN}^p(r|x)$ denotes (conditional) non-deterministic printing complexity, defined as the length of a shortest non-deterministic program w (simulated on a non-deterministic universal machine U_n) such that:

- $U_n(w, x)$ has at least one accepting path.

- $U_n(w, x)$ outputs r on every accepting path.
- $U_n(w, x)$ runs in at most $p(|r| + |x|)$ steps.

Note that $\mathsf{CN}^p(r) \leq m^3 - O(m^2) + n < m^3$, where we used that $m > n$, $|x| = n$, and $\mathsf{CN}^p(r|x) \leq m^3 - O(m^2)$. Therefore any m-bit string with CN^p complexity $\geq m^3$ will always be a good choice of the random string for M', no matter the choice of x.

We define y_l to be the lexicographically smallest string of length l satisfying $\mathsf{CN}^p(y_l) = l$. We then define a language R: $(1^l, i, b) \in R$ if and only if the i'th bit of y_l is b. Then one can see that $R \in \mathsf{EXP}$. By our assumption that $\mathsf{ZPEXP} \subseteq \mathsf{BPP}$, $R \in \mathsf{BPP}$. Therefore, given 1^l as input, by computing R and using error reduction, we can output y_l in polynomial time with probability $> 2/3$, implying $\mathsf{rKt}(y_l) = O(\log l)$.

Next, by defining an appropriate sampler that selects a random program of length at most $O(\log \ell)$ and simulates it for at most $2^{O(\log \ell)}$ steps (see, e.g., [LO21, Section 4.1]), we can output y_ℓ with probability at least $1/2^{O(\log \ell)}$. Invoking the assumed near-optimal coding theorem for zKt, we conclude that $\mathsf{zKt}(y_l) = O(\log l)$.

Since $\mathsf{zKt}(y_l) = O(\log l)$, using $O(\log l)$ bits of advice, we can compute y_l in polynomial time with zero error. Lastly, since y_{m^3} is always a good random string for M', we can compute L in polynomial time with zero error. This implies $\mathsf{BPP} \subseteq \mathsf{ZPP}/_{O(\log n)}$, as desired.[7] □

Both $\mathsf{RPEXP} \not\subseteq \mathsf{RP}/_{O(\log n)}$ and $\mathsf{BPEXP} \not\subseteq \mathsf{BPP}/_{O(\log n)}$ hold unconditionally [BFS09]. It is also known that $\mathsf{ZPEXP} \not\subseteq \mathsf{ZPP}$. However, achieving the separation in the zero-error case against $O(\log n)$ bits of advice is an interesting open problem.

5.3 Unconditional Near-Optimal zKt Coding for Flat Sources

A distribution $\mathcal{D}_n$ supported over $\{0,1\}^n$ is *flat* if there is a set $S \subseteq \{0,1\}^n$ such that $\mathcal{D}_n$ is uniformly distributed over S. In this section, we note that the near-optimal coding theorem for rKt established in [LO21] provides a near-optimal coding theorem for zKt if $\mathcal{D}_n$ is a flat polynomial-time samplable distribution. The result easily follows from the following more general statement.

Theorem 38. *Let $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$ be a polynomial-time samplable distribution. For every $n \geq 1$, let $\delta_n \in [0,1]$ be a parameter such that if $x \in \mathsf{Support}(\mathcal{D}_n)$ then $\mathcal{D}_n(x) \geq \delta_n$. Then*

$$\mathsf{zKt}(x) = O(\log(1/\delta_n) + \log n).$$

Proof Sketch. We explain why the proof of [LO21, Theorem 20] provides a zero-error encoding under the extra assumption that every element in the support of $\mathcal{D}_n$ has probability weight at least δ_n.

First, note that a key component of this proof is [LO21, Lemma 19], which shows how to efficiently isolate a string x from a collection W of strings using a short advice string v whose length depends on the logarithm of the size of W. This lemma will be used in a black-box way without modifications.

Now we proceed as in the proof of [LO21, Theorem 20]. We employ the following analogous definition for the set W, i.e.,

$$W \triangleq \{w \in \mathsf{Support}(\mathcal{D}_n) \mid \mathcal{D}_n(w) \geq \delta_n/32\},$$

which simplifies to $W = \mathsf{Support}(\mathcal{D}_n)$ under the assumption over the distribution. Note that $|W| \leq 1/\delta_n$. By [LO21, Lemma 19], there is a string u of length $O(\log(1/\delta_n) + \log n)$ such that the machine $M(1^n, u)$

[7] Alternatively, for any $L \in \mathsf{BPP}$, by reducing the error of a machine computing L to at most 2^{-n-1}, there exists a good choice of the random string that works for all inputs of length n. Then given 1^n one can find the first good random string r_n in exponential time, and by our assumption that $\mathsf{EXP} \subseteq \mathsf{BPP}$, it is easy to describe a sampler A such that $\mathbf{Pr}[A(1^n) = r_n] \geq 2/3$. Using near-optimal coding for zKt, we can conclude in a similar way that $L \in \mathsf{ZPP}/_{O(\log n)}$.

runs in deterministic time $\mathsf{poly}(n)$ and outputs a Boolean circuit that computes a function $H\colon \{0,1\}^n \to \{0,1\}^{O(\log 1/\delta_n)}$ such that

$$H(w) \neq H(w') \text{ for every distinct pair } w, w' \in W.$$

In particular, u and the value $H(x)$ allow us to identify x among any set S of strings with $S \subseteq W$.

Crucially, inspecting the remaining steps of the argument we are able to achieve a zero-error encoding because the set S constructed in the proof of [LO21, Theorem 20] is always a subset of W under the extra assumption on $\mathcal{D}_n$. Therefore, we either recover x when it is in S, or output "$\perp$" otherwise. ☐

Corollary 39. *Let $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution. If each $\mathcal{D}_n$ is flat, then for every $n \geq 1$ and for every string $x \in \mathsf{Support}(\mathcal{D}_n)$, we have*

$$\mathsf{zKt}(x) = O(\log(1/\mathcal{D}_n(x)) + \log n).$$

6 Complexity Separations and Meta-Complexity

6.1 Complexity of Approximating nKt Complexity

Theorem 4. *The following are equivalent.*

1. $\mathsf{NEXP} \neq \mathsf{BPP}$.
2. $\mathsf{MnKtP}[n^\varepsilon, n-1] \notin \mathsf{prBPP}$, *for all* $\varepsilon > 0$.

Moreover, the above holds if we replace MnKtP *with* $\mathsf{MKt^{NP}P}$*, and* NEXP *with* $\mathsf{EXP^{NP}}$.[8]

Proof. We show each implication below.

Item 1 ⇒ Item 2: For the sake of contradiction, suppose $\mathsf{NEXP} \neq \mathsf{BPP}$ and $\mathsf{MnKtP}[n^\varepsilon, n-1] \in \mathsf{prBPP}$, for some $\varepsilon > 0$.

Let A be a randomized polynomial-time algorithm for deciding $\mathsf{MnKtP}[n^\varepsilon, n-1]$ with exponentially small error. Let $c > 0$ be the constant such that A runs in time n^c on inputs of length n.

Let $b > c$ be a sufficiently large constant, and let $G \triangleq \{G_n : \{0,1\}^{n^{\varepsilon/2}} \to \{0,1\}^n\}$ be the sequence of generators provided by Lemma 26, such that for every distribution family $\{\mathcal{C}_n\}$ of Boolean circuits samplable in time n^b, there are infinitely many $n \in \mathbb{N}$ such that, with probability at least $1 - n^{-b}$ over C sampled from $\mathcal{C}_n$, G_n (n^{-b})-fools C.

Consider the following distribution family $\{\mathcal{C}_n\}$ of circuits, where each $\mathcal{C}_n$ is given by the following sampling procedure:

> On input 1^n, sample r uniformly at random from $\{0,1\}^{\mathsf{poly}(n)}$. Construct the circuit C_r such that $C_r(x) = A(x; r)$ for every $x \in \{0,1\}^n$. Finally, output C_r.

Note that $\mathcal{C}_n$ is samplable in time n^b if b is sufficiently large. By construction, for every $n \in \mathbb{N}$, with probability at least, say, $2/3$ over a circuit C sampled from $\mathcal{C}_n$, C correctly decides $\mathsf{MnKtP}[n^\varepsilon, n-1]$. In what follows, fix $n \in \mathbb{N}$ and such a circuit C.

[8] In fact, in all these results, the proof implicitly shows that the gap version of the problem is easy if and only if the non-gap version is easy. For instance, it is known that $\mathsf{Gap\text{-}MKtP} \notin \mathsf{prBPP}$ if and only if $\mathsf{MKtP} \notin \mathsf{BPP}$. This will also be the case for the equivalences established in this paper.

On the one hand, since G_n can be computed non-deterministically in time $2^{O(n^{\varepsilon/2})}$ with n^ε bits of advice, we have that for every $z \in \{0,1\}^{n^\varepsilon}$, $\mathsf{nKt}(G_n(z)) \leq n^\varepsilon$. It follows that

$$\Pr_{z \sim \{0,1\}^{n^{\varepsilon/2}}}[C(G_n(z)) = 1] = 1. \tag{8}$$

On the other hand, by a simple counting argument, for at least half of the x's in $\{0,1\}^n$, $\mathsf{nKt}(x) \geq \mathsf{K}(x) > n-1$. This implies

$$\Pr_{x \sim \{0,1\}^n}[C(x) = 0] \geq \frac{1}{2}. \tag{9}$$

Comparing Equations (8) and (9), we conclude that C is not fooled by G_n. This contradicts the pseudorandomness property of G.

Item 2 $\Rightarrow$ Item 1: We show the contrapositive. First, it is easy to see that $\mathsf{MnKtP} \in \mathsf{PSPACE}^{\mathsf{NEXP}}$. Indeed, given $(x, 1^s)$, one can enumerate every program p and time bound t such that $|p| + \log t \leq s$ and use an NEXP oracle to check the following conditions:

- $\forall w \in \{0,1\}^t$, $U(p,w)$ outputs x or $\perp$ within t steps.
- $\exists w \in \{0,1\}^t$, $U(p,w)$ outputs x within t steps.

Note that the queries made to the NEXP oracle are of size polynomial in the length of the input string $(x, 1^s)$, since $|p| + \log t \leq s$. Consequently, under the assumption that $\mathsf{NEXP} = \mathsf{BPP}$, the answer to each oracle query can be computed in BPP and therefore in polynomial space. This yields $\mathsf{MnKtP} \in \mathsf{PSPACE}$. Invoking $\mathsf{NEXP} \subseteq \mathsf{BPP}$ once again, we get that $\mathsf{MnKtP} \in \mathsf{BPP}$.

The above completes the proof of the equivalence between "$\mathsf{NEXP} \neq \mathsf{BPP}$" and "$\mathsf{MnKtP}[n^\varepsilon, n-1] \notin \mathsf{prBPP}$".

Similarly, the equivalence between "$\mathsf{EXP}^{\mathsf{NP}} \neq \mathsf{BPP}$" and "$\mathsf{MKt}^{\mathsf{NP}}\mathsf{P}[n^\varepsilon, n-1] \notin \mathsf{prBPP}$" can be shown using an infinitely-often secure pseudorandom generator $G = \{G_n\}_{n \in \mathbb{N}}$, where each $G_n \colon \{0,1\}^{n^\varepsilon} \to \{0,1\}^{n^b}$ is computable in time $2^{O(n^\varepsilon)}$ *with access to an* NP *oracle*. Again, such a pseudorandom generator can be obtained using an argument similar to the one in the proof of Lemma 19. $\square$

6.2 Complexity of Approximating zKt Complexity

6.2.1 Proof of Theorem 5

Theorem 5. *If* $\mathsf{MzKtP}[n^\varepsilon, n-1] \in \mathsf{prBPP}$ *for some* $\varepsilon > 0$*, then* $\mathsf{prZPEXP} = \mathsf{prBPP}$.

Proof. The proof can be adapted easily from that of (Item 1 $\Rightarrow$ Item 2) in Theorem 4. More precisely, assume $\mathsf{prZPEXP} \neq \mathsf{prBPP}$. Then, by Lemma 36, we get an infinitely-often secure pseudorandom generator that is computable probabilistically with zero error using a small amount of advice.

Suppose, for the sake of contradiction, that $\mathsf{MzKtP}[n^\varepsilon, n-1] \in \mathsf{prBPP}$ for some $\varepsilon > 0$. Then, an efficient algorithm solving $\mathsf{MzKtP}[n^\varepsilon, n-1]$ can be used to break the aforementioned pseudorandom generator. $\square$

6.2.2 Proof of Theorem 6

Theorem 6. $\mathsf{MzKtP}[n^{\varepsilon}, n-1] \notin \mathsf{prZPTIME}\big[2^{\mathsf{polylog}(n)}\big]$, *for all* $\varepsilon > 0$.

Proof. Towards a contradiction, assume that for some $\varepsilon > 0$ and $c \geq 1$ there is a zero-error probabilistic algorithm A running in time $2^{(\log n)^c}$ that separates n-bit strings x with $\mathsf{zKt}(x) \leq n^{\varepsilon}$ from those with $\mathsf{zKt}(x) \geq n^{1-\varepsilon}$. We make no assumption about the output behavior of A on the remaining inputs.

Lemma 40. *Assume the existence of an algorithm A as above. Then there exists a constant $\delta > 0$ for which the following holds. For every large enough n and for every string $y \in \{0,1\}^n$, if $\mathsf{rKt}(y) \leq n^{\delta}$ then $\mathsf{zKt}(y) \leq n^{\varepsilon}$.*

Proof. Let y be an n-bit string such that $\mathsf{rKt}(y) \leq n^{\delta}$, and let M be a program of length at most n^{δ} that outputs y with probability at least $2/3$ when running for at most $2^{n^{\delta}}$ steps. In short, we use the assumed algorithm A to randomly guess a random ℓ-bit string z of length $\mathsf{poly}(N)$ (where $N = 2^{n^{\delta}}$) and verify that it encodes a hard truth table. We can then instantiate a PRG based on z that allows us to transform M into a zero-error machine M' that outputs y.

In more detail, we aim to obtain a machine M' of length at most $n^{\varepsilon/2}$ that runs in time at most $2^{n^{\varepsilon/2}}$, outputs y with probability at least $2/3$, and always outputs either y or $\perp$. To derandomize a machine running in time N, we employ a PRG $G\colon \{0,1\}^{O(\log N)} \to \{0,1\}^N$ that fools N-size computations. The latter can be constructed with access to a string of length $\ell = \mathsf{poly}(N)$ of circuit complexity $\geq \ell^{\Omega(1)}$ [IW97]. In turn, since A is a zero-error algorithm on inputs $z \in \{0,1\}^{\ell}$ with $\mathsf{zKt}(z) \leq \ell^{\varepsilon}$, it is not hard to see that if we run A on a random input $z \in \{0,1\}^{\ell}$, with probability at least $1/2$ over the choice of z we have $\mathsf{zKt}(z) \geq \ell - 1$, and with probability at least $1/2$ over the internal randomness of A, we have $A(z) =$ "NO", meaning that A certifies that z is not a string with $\mathsf{zKt}(z) \leq \ell^{\varepsilon}$. (Crucially, no matter the internal randomness of A, it will never output "NO" on an ℓ-bit input string of zKt complexity at most ℓ^{ε}, since on such inputs it always outputs either "YES" or "$\perp$".) Since every string $z \in \{0,1\}^{\ell}$ with $\mathsf{Kt}(z) \geq \ell^{\varepsilon}$ encodes a truth table of circuit complexity at least $\ell^{\varepsilon/2} = \ell^{\Omega(1)}$, A can be used to generate and certify a hard truth table.

Given a hard truth table z, since M outputs x with probability at least $2/3$ and G^z fools M, we have $\mathbf{Pr}_w[M(G^z(w)) = x] > 1/2$. Thus our zero-error machine M' computes as follows. First, it attempts to guess and certify a hard truth table using A. It outputs $\perp$ if it does not succeed. Otherwise, it cycles over every seed $w \in \{0,1\}^{O(\log N)}$ and outputs the most common string produced via $M(G^z(w))$.

Note that M' is indeed a zero-error encoding of x. It remains to bound the running time and description length of M'. Its description length is bounded by the descriptions of M, A, n, and ℓ, which is at most $n^{\delta} + O(1) + O(\log n) \leq n^{\varepsilon/2}$, assuming that $\delta \leq \varepsilon/3$. On the other hand, M runs in time at most $2^{n^{\delta}}$, A on an input of length $\ell = \mathsf{poly}(N)$ with $N = 2^{n^{\delta}}$ runs in time at most $2^{(\log \ell)^c} = 2^{n^{2\delta c}}$, and producing all outputs $G^z(w)$ with $w \in \{0,1\}^{O(\log N)}$ takes time at most $2^{O(\log N)} = 2^{O(n^{\delta})}$. Overall, the running time of M' is at most $2^{n^{\varepsilon/2}}$ if we take $\delta < \varepsilon/(4c)$. This completes the proof that $\mathsf{zKt}(y) \leq n^{\varepsilon}$ if $\mathsf{rKt}(y) \leq n^{\delta}$, provided that $\delta = \delta(\varepsilon) > 0$ is small enough and n is sufficiently large. $\square$

Let n be large enough. For a string $y \in \{0,1\}^*$, the following implications hold:

- If $\mathsf{rKt}(y) \leq n^{\delta}$, then $\mathsf{zKt}(y) \leq n^{\varepsilon}$.
- If $\mathsf{rKt}(y) \geq n-1$, then $\mathsf{zKt}(y) \geq n-1$.

The first implication follows from Lemma 40, while the second implication uses that $\mathsf{rKt}(y) \leq \mathsf{zKt}(y)$. As a consequence, the algorithm A decides $\mathsf{MrKtP}[n^{\delta}, n-1]$. In particular, we have $\mathsf{MrKtP}[n^{\delta}, n-1] \in \mathsf{prBPTIME}[n^{\mathsf{poly}(\log n)}]$. However, this contradicts the unconditional lower bound for $\mathsf{MrKtP}[n^{\delta}, n-1]$ established in [Oli19], which shows that this promise problem cannot be solved by a probabilistic algorithm that runs in quasi-polynomial time. $\square$

References

[ABK+06] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006.

[AF09] Luis Filipe Coelho Antunes and Lance Fortnow. Worst-case running times for average-case algorithms. In *Conference on Computational Complexity* (CCC), pages 298–303, 2009.

[AKRR11] 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.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Comput. Complex.*, 1:3–40, 1991.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BFS09] Harry Buhrman, Lance Fortnow, and Rahul Santhanam. Unconditional lower bounds against advice. In *International Colloquium on Automata, Languages and Programming* (ICALP), pages 195–209, 2009.

[BH92] Harry Buhrman and Steven Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In *Foundations of Software Technology and Theoretical Computer Science* (FSTTCS), pages 116–127, 1992.

[BLMP23] Marshall Ball, Yanyi Liu, Noam Mazor, and Rafael Pass. Kolmogorov comes to cryptomania: On interactive Kolmogorov complexity and key-agreement. In *Symposium on Foundations of Computer Science* (FOCS), pages 458–483, 2023.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Conference on Computational Complexity* (CCC), pages 10:1–10:24, 2016.

[GK23] Halley Goldberg and Valentine Kabanets. Improved learning from Kolmogorov complexity. In *Computational Complexity Conference* (CCC), pages 12:1–12:29, 2023.

[GK24] Halley Goldberg and Valentine Kabanets. Consequences of randomized reductions from SAT to time-bounded Kolmogorov complexity. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques* (APPROX/RANDOM), pages 51:1–51:19, 2024.

[GKLO22] Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. Probabilistic Kolmogorov complexity with applications to average-case complexity. In *Computational Complexity Conference* (CCC), pages 16:1–16:60, 2022.

[HIL+23] Shuichi Hirahara, Rahul Ilango, Zhenjian Lu, Mikito Nanashima, and Igor C. Oliveira. A duality between one-way functions and average-case symmetry of information. In *Symposium on Theory of Computing* (STOC), pages 1039–1050, 2023.

[Hir21] Shuichi Hirahara. Average-case hardness of NP from exponential worst-case hardness assumptions. In *Symposium on Theory of Computing* (STOC), pages 292–302, 2021.

[Hir22] Shuichi Hirahara. Symmetry of information from meta-complexity. In *Computational Complexity Conference* (CCC), pages 26:1–26:41, 2022.

[HKLO24] Shuichi Hirahara, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. Exact search-to-decision reductions for time-bounded Kolmogorov complexity. In *Computational Complexity Conference* (CCC), pages 29:1–29:56, 2024.

[HLN24] Shuichi Hirahara, Zhenjian Lu, and Mikito Nanashima. Optimal coding for randomized kolmogorov complexity and its applications. In *Symposium on Foundations of Computer Science* (FOCS), pages 369–378, 2024.

[HLO24] Shuichi Hirahara, Zhenjian Lu, and Igor C. Oliveira. One-way functions and pKt complexity. In *Theory of Cryptography* (TCC), pages 253–286, 2024.

[HN23] Shuichi Hirahara and Mikito Nanashima. Learning in Pessiland via inductive inference. In *Symposium on Foundations of Computer Science* (FOCS), pages 447–457, 2023.

[IKW02] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: exponential time vs. probabilistic polynomial time. *J. Comput. Syst. Sci.*, 65(4):672–694, 2002.

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In *Symposium on Theory of Computing* (STOC), pages 1575–1583, 2022.

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Symposium on Theory of Computing* (STOC), pages 220–229. ACM, 1997.

[IW01] Russell Impagliazzo and Avi Wigderson. Randomness vs time: Derandomization under a uniform assumption. *J. Comput. Syst. Sci.*, 63(4):672–688, 2001.

[Kab00] Valentine Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. In *Conference on Computational Complexity* (CCC), pages 150–157, 2000.

[KL80] Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Symposium on Theory of Computing* (STOC), pages 302–309, 1980.

[Lee06] Troy Lee. *Kolmogorov complexity and formula lower bounds*. PhD thesis, University of Amsterdam, 2006.

[Lev74] Leonid A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.

[Lev84] Leonid A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984.

[LMOP24] Zhenjian Lu, Noam Mazor, Igor C. Oliveira, and Rafael Pass. Lower bounds on the overhead of indistinguishability obfuscation. *IACR Cryptol. ePrint Arch.*, page 1524, 2024.

[LO21] Zhenjian Lu and Igor C. Oliveira. An efficient coding theorem via probabilistic representations and its applications. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 94:1–94:20, 2021.

[LO22] Zhenjian Lu and Igor C. Oliveira. Theory and applications of probabilistic Kolmogorov complexity. *Bull. EATCS*, 137, 2022.

[LORS24] Zhenjian Lu, Igor C. Oliveira, Hanlin Ren, and Rahul Santhanam. On the complexity of avoiding heavy elements. In *Symposium on Foundations of Computer Science* (FOCS), pages 2403–2412, 2024.

[LOZ22] Zhenjian Lu, Igor C. Oliveira, and Marius Zimand. Optimal coding theorems in time-bounded Kolmogorov complexity. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 92:1–92:14, 2022.

[LP20] Yanyi Liu and Rafael Pass. On one-way functions and Kolmogorov complexity. In *Symposium on Foundations of Computer Science* (FOCS), pages 1243–1254, 2020.

[LP23] Yanyi Liu and Rafael Pass. One-way functions and the hardness of (probabilistic) time-bounded Kolmogorov complexity w.r.t. samplable distributions. In *Annual Cryptology Conference* (CRYPTO), pages 645–673, 2023.

[LP24] Yanyi Liu and Rafael Pass. On one-way functions, the worst-case hardness of time-bounded Kolmogorov complexity, and computational depth. In *Theory of Cryptography* (TCC), pages 222–252, 2024.

[LP25] Yanyi Liu and Rafael Pass. Hardness along the boundary towards one-way functions from the worst-case hardness of time-bounded Kolmogorov complexity. In *Annual Cryptology Conference* (CRYPTO), 2025.

[LPT24] Jiatu Li, Edward Pyne, and Roei Tell. Distinguishing, predicting, and certifying: On the long reach of partial notions of pseudorandomness. In *Symposium on Foundations of Computer Science* (FOCS), pages 1–13, 2024.

[LS24] Zhenjian Lu and Rahul Santhanam. Impagliazzo's worlds through the lens of conditional Kolmogorov complexity. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 110:1–110:17, 2024.

[Oli19] Igor C. Oliveira. Randomness and intractability in Kolmogorov complexity. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 32:1–32:14, 2019.

[OS17] Igor C. Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Computational Complexity Conference* (CCC), pages 18:1–18:49, 2017.

[San23] Rahul Santhanam. An algorithmic approach to uniform lower bounds. In *Computational Complexity Conference* (CCC), pages 35:1–35:26, 2023.

[TV07] Luca Trevisan and Salil P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.

A Equivalence Between nKt and KNt

The following non-deterministic variant of time-bounded Kolmogorov complexity was introduced in [AKRR11].

Definition 41 (KNt [AKRR11])**.** For $x \in \{0,1\}^*$, define

$$\mathsf{KNt}(x) \triangleq \min_{p \in \{0,1\}^*,\, t \in \mathbb{N}} \{|p| + \lceil \log t \rceil \mid \forall i \le n+1, V(p,i,b) \text{ runs in time } t \text{ and accepts iff } x_i = b\},$$

where V is a fixed universal non-deterministic Turing machine.

For completeness, we observe here that the following equivalence holds.

Proposition 42. *For every $x \in \{0,1\}^*$, we have*

$$\mathsf{nKt}(x) = \mathsf{KNt}(x) \pm O(\log |x|).$$

Proof. Let $x \in \{0,1\}^n$ and $s \triangleq \mathsf{KNt}(x)$. First, we show that $\mathsf{nKt}(x) \le s + O(\log n)$.

Let p be a non-deterministic program and t a time bound such that $|p| + \lceil \log t \rceil = s$, and $V(p,i,b)$ runs in time t and accepts if and only if $x_i = b$, for all $i \le n+1$. We view V as a deterministic algorithm that has access to an additional tape holding the "guess" string. That is, for all $i \le n+1$, there exists $w_i^* \in \{0,1\}^t$ such that $V(p,i,x_i;w_i^*)$ accepts within t steps, and for all $w_i \in \{0,1\}^t$, $V(p,i,\neg x_i;w_i)$ rejects within t steps.

Consider the following procedure for outputting x non-deterministically:

> Given a guess w, we view it as $(y, w_1, w_2, \ldots, w_n)$, where $y \in \{0,1\}^n$ and each $w_i \in \{0,1\}^t$ for some t. We then check whether $V(p,i,y_i;w_i)$ accepts for all $i \in [n]$. If so, we output y; otherwise, we output $\perp$.

We first argue correctness. Consider the "correct" guess $w \triangleq (x, w_1^*, w_2^*, \ldots, w_n^*)$. It is easy to see, by the property of p, that the above procedure will output x when given w. Also, note that for any guess of the form $(y, w_1, w_2, \ldots, w_n)$, the procedure will only output y or $\perp$, and if $y \ne x$, then, again by the property of p, the procedure will output $\perp$ because in this case $V(p,i,y_i;w_i)$ will reject for at least one $i \in [n]$.

Also, note that given the program p and the number n, the above procedure can be implemented to run in time $t \cdot \mathsf{poly}(n)$. This implies

$$\mathsf{nKt}(x) \le |p| + O(\log n) + \log(t \cdot \mathsf{poly}(n)) \le s + O(\log n),$$

as desired.

Now let $x \in \{0,1\}^n$ and $s \triangleq \mathsf{nKt}(x)$. Next, we show that $\mathsf{KNt}(x) \le s + O(\log n)$.

Let p be a program and t a time bound such that $|p| + \lceil \log t \rceil = s$ and the following conditions hold:

- $\forall w \in \{0,1\}^t$, $U(p,w)$ outputs x or $\perp$ within t steps,
- $\exists w \in \{0,1\}^t$, $U(p,w)$ outputs x within t steps,

where U is a *deterministic* universal Turing machine.

We describe the following non-deterministic procedure:

> On an input (i,b) and a guess string w, we run $U(p;w)$ to obtain a string x. Accept if and only if $x_i = b$.

For an input (i,b), if $b = x_i$, then there exists some guess $w \in \{0,1\}^t$ such that $U(p;w)$ outputs x, and the above will accept. On the other hand, if $b \ne x_i$, then since $U(p;w)$ only outputs x or $\perp$ for all guesses $w \in \{0,1\}^t$, the above will reject.

Also, note that the above procedure can be implemented to run in time $t \cdot \mathsf{poly}(n)$. This yields

$$\mathsf{KNt}(x) \le |p| + O(\log n) + \log(t \cdot \mathsf{poly}(n)) \le s + O(\log n),$$

as desired. $\square$

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