Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 92 (2025)

Complexity-Theoretic Universal Inductive Inference

Shuichi Hirahara* Mikito Nanashima†

Abstract

Solomonoff's theory of universal inductive inference (Inf. Control., 1964) provides a framework for predicting a future observation from past ones generated by an arbitrary randomized Turing machine. The theory is founded on the notion of resource-unbounded Kolmogorov complexity, and thus Solomonoff's approach cannot be realized as a finite-step algorithm.

In this paper, we develop a *complexity-theoretic* counterpart of Solomonoff's theory. We construct a *polynomial-time* universal inductive inference algorithm that extrapolates a sequence of symbols generated by any unknown t-time randomized Turing machine in time polynomial in t, assuming that time-bounded Kolmogorov complexity can be computed in average polynomial time. Previously, it was not even known whether distributional learning for all polynomial-size circuits—an i.i.d. analogue of inductive inference—is feasible if NP is easy on average. Moreover, without any unproven assumption, we characterize a distribution of sequences for which there exists an efficient inductive inference algorithm by the notion of prequential compression. We also construct an *optimal* efficient inductive inference algorithm that performs as well as any other efficient algorithms.

Our universal inductive inference algorithm relies on (1) a new algorithmic proof of a chain rule for time-bounded algorithmic information, and (2) an online algorithm that boosts the "confidence" of our inductive inference algorithm.

*National Institute of Informatics, Japan. s_hirahara@nii.ac.jp
†Institute of Science Tokyo, Japan. nanashima@comp.isct.ac.jp

ISSN 1433-8092

Contents

1 Introduction

Inductive inference, introduced by Solomonoff [Sol64a; Sol64b], is a foundational concept underlying the process of knowledge acquisition. In Solomonoff's formulation, the problem is to extrapolate a long sequence of symbols: given a sequence $x_1, x_2, \ldots, x_{i-1}$, the task is to predict the next symbol x_i for each $i \in \mathbb{N}$. This type of reasoning arises in many real-world contexts. Scientists infer laws of nature from empirical observations; artificial intelligence systems learn patterns from data; and animals develop instincts that help them recognize danger in their environment. All of these can be viewed as forms of inductive inference, namely using observed data to predict the future. Developing an algorithmic foundation for inductive inference is therefore a central challenge in science.

Solomonoff's theory presents an elegant framework for *universal* inductive inference. The goal is to extrapolate, according to a single inference principle, any sequence generated by an unknown randomized Turing machine. His approach bases the inference rule on *Kolmogorov complexity*, where the Kolmogorov complexity $\mathrm{K}(x)$ of a string x is the length of a shortest program that outputs x. Roughly speaking, given a prefix $\boldsymbol{x}_{<i} = (x_1, \ldots, x_{i-1})$, Solomonoff's universal inference assigns probability $p_{\mathsf{sol}}(x_i \mid \boldsymbol{x}_{<i}) \propto \sum_y 2^{-\mathrm{K}(\boldsymbol{x}_{<i}, x_i y)}$ to each candidate continuation x_i. Despite its conceptual simplicity, this scheme already achieves the universal inference guarantee described above. Solomonoff's inductive inference has profoundly influenced subsequent research and has been used as a foundational building block in the theory of universal artificial intelligence [see, e.g., MF98; Hut05; HQC24].

Complexity-Theoretic Implementation of Solomonoff's Theory. There is a major caveat in Solomonoff's original approach, as he noted himself [Sol64a, Section 3.1.2.1]. The scheme depends on $\mathrm{K}(\cdot)$, which is *provably* uncomputable. Thus, Solomonoff's algorithm for inductive inference cannot be implemented as an algorithm that halts in finite steps. To address this issue, in his seminal paper, he suggested replacing K with what is now called *time-bounded* Kolmogorov complexity. For a given time bound $t \in \mathbb{N}$, the t-time-bounded Kolmogorov complexity $\mathrm{K}^t(x)$ of a string x is the length of a shortest program that prints x within time t. This suggests a complexity-theoretic variant that assigns mass $p^t_{\mathsf{sol}}(x_i \mid \boldsymbol{x}_{<i}) \propto \sum_y 2^{-\mathrm{K}^t(\boldsymbol{x}_{<i}, x_i y)}$, for which analogous guarantees hold for all *efficiently* generated sequences by the same argument. However, even under the assumption that K^t is efficiently computable, whether Solomonoff's inductive inference can be efficiently implemented remains a fundamental open question. This motivates the following question.

Question 1.1. *Can we establish a complexity-theoretic counterpart of Solomonoff's theory of inductive inference? Specifically, assuming the existence of an efficient algorithm that approximates time-bounded Kolmogorov complexity, is there an efficient algorithm for universal inductive inference that works for every efficiently generated sequence?*

A difficulty of resolving this question affirmatively stems from the fact that the problem of approximating K^t, denoted by GapMINKT, is currently in an NP-intermediate status. Hirahara [Hir18; Hir20b] showed that GapMINKT $\in \mathsf{P}$ if $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ (NP is easy on average), which implies that NP-completeness of GapMINKT would establish the equivalence between worst- and average-case complexities of NP—the major open problem in the theory of average-case complexity, known as excluding Heuristica from Impagliazzo's five worlds [Imp95]. Although it is easy to implement a time-bounded variant of Solomonoff's inductive inference under the assumption that $\mathsf{P} = \mathsf{NP}$, the assumption in Question 1.1 is much weaker, and in particular, weaker than $\mathsf{DistNP} \subseteq \mathsf{AvgP}$. Bogdanov and Trevisan [BT06] showed that any worst-case problem nonadaptively reducible to DistNP is in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$, which poses a significant challenge to Question 1.1 as well as to ruling out Heuristica.

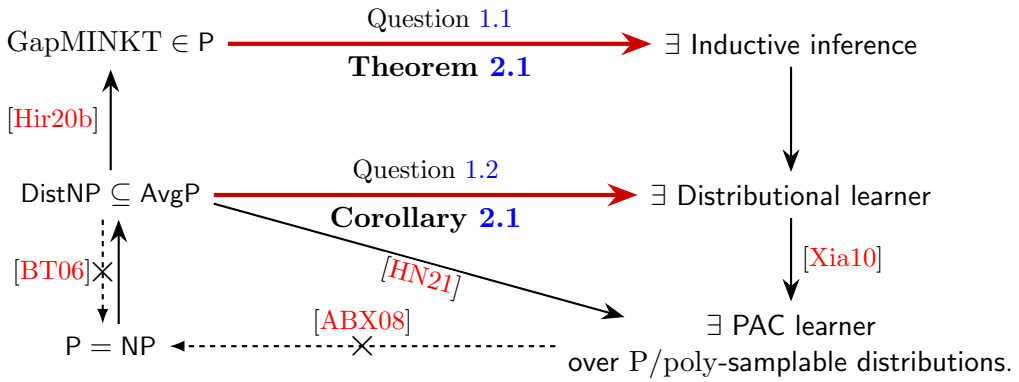

Figure 1: Solid arrows denote known implications. "∃ Inductive inference" in the figure refers to the statement that there exists a polynomial-time algorithm for inductive inference of every efficiently generated sequence. Dashed arrows with a cross mark (×) represent implications that cannot be established via certain classes of proof techniques ([BT06; ABX08]). Red arrows indicate the new results of this paper.

Currently, it is not even known if distributional learning—an i.i.d. analogue of inductive inference—is feasible if DistNP ⊆ AvgP. *Distributional learning*, introduced by Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie [KMRRSS94], is the task of learning an unknown distribution $\mathcal{D}$ using independent and identically distributed (i.i.d.) samples from $\mathcal{D}$. This is a special case of inductive inference where each symbol x_i is promised to be drawn independently from the same distribution $\mathcal{D}$, whereas inductive inference must handle arbitrary dependent data sequences. Thus, the following is an open question much weaker than Question 1.1.

Question 1.2. *Assuming that* NP *is easy on average, is there an efficient algorithm for distributional learning of every* P/poly*-samplable distribution?*

Which classes of learning problems admit NP-hardness proofs has been a central topic in computational learning theory. On a positive side, NP-hardness of PAC learning can be proved when the class of hypotheses is restricted [PV88], the description length of hypotheses is restricted [Hir22a], or the sample complexity is limited [BKST25]. On a negative side, Applebaum, Barak, and Xiao [ABX08] showed that NP-hardness of PAC learning cannot be established via non-adaptive reductions unless NP ⊆ coAM. Hirahara and Nanashima [HN21] showed that PAC learning of P/poly (polynomial-size circuits) over P/poly-samplable distribution is efficiently solvable under the assumption that DistNP ⊆ AvgP, indicating that proving NP-hardness of this learning task is at least as hard as ruling out Heuristica.[1] Xiao [Xia10] proved that PAC learning over P/poly-samplable distributions is reducible to distributional learning, and thus the former is easier than the latter. Question 1.2 lies at the frontier of this line of research and in the intersection of computational learning theory, average-case complexity theory, and meta-complexity theory, asking whether distributional learning is as hard as solving NP or not; see Figure 1 for the summary.

2 Our Results

In this paper, we provide an affirmative answer to Question 1.1 and hence to Question 1.2, thereby establishing a complexity-theoretic counterpart of Solomonoff's theory of universal inductive in-

[1] A follow-up work of [HN21] by Goldberg and Kabanets [GK23] improved the assumption of DistNP ⊆ AvgP to GapMINKT ∈ BPP.

ference (Section 2.1). Moreover, in Section 2.2, we establish a characterization of distributions of sequences for which there exists a polynomial-time inductive inference algorithm *without any unproven complexity-theoretic assumption*.

2.1 Complexity-Theoretic Universal Inductive Inference Using GapMINKT

We first formalize the notion of complexity-theoretic universal inductive inference. Our goal is to design an algorithm that, given a size parameter s and a time parameter t, predicts the next symbol x_i (encoded as a binary string of length n) at almost all positions i of the sequence generated by any unknown t-time randomized Turing machine of description length s. We call such an inference algorithm *universal* because the *single* algorithm works for *every* sequence generated by an arbitrary randomized Turing machine. We adopt accuracy and confidence parameters (ϵ, δ) as in the standard PAC model [Val84] and measure accuracy by the total variation distance: two distributions X and Y are said to be *ϵ-close*, denoted $X \equiv_\epsilon Y$, if the total variation distance between X and Y is at most ϵ. An inference algorithm is considered to be accurate at the i-th round if the distribution of the symbol x'_i predicted by the algorithm is ϵ-close to the conditional distribution of x_i given $x_{<i}$. A formal definition follows.

Definition 2.1 (Complexity-Theoretic Universal Inductive Inference)**.** *Fix parameters $n, s, t \in \mathbb{N}$ and accuracy/confidence parameters $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$. Let L be a randomized algorithm that takes the parameters $\mathsf{param} = (n, s, t, \epsilon^{-1}, \delta^{-1})$ and $i-1$ preceding strings $\boldsymbol{x}_{<i} = (x_1, \ldots, x_{i-1}) \in (\{0,1\}^n)^{i-1}$, and outputs a string in $\{0,1\}^n$. We write $L(\boldsymbol{x}_{<i}; \mathsf{param})$ for the output distribution of L on input $\boldsymbol{x}_{<i}$, where the randomness is over the internal randomness of L.*

Let Π be an arbitrary randomized Turing machine described by an s-bit string that, in time at most t, generates at least m strings $x_1, \ldots, x_m$ with each $x_j \in \{0,1\}^n$. Let $(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ denote the joint distribution of the m strings (not necessarily i.i.d.). For a function $\sigma : \mathbb{N}^4 \to \mathbb{N}$, we say that L solves complexity-theoretic universal inductive inference *with round complexity σ if, for every such Π and every $m \geq \sigma(s, n, \epsilon^{-1}, \delta^{-1})$,*

$$\Pr_{i \sim [m],\, \boldsymbol{x}_{<i}} \left[L(\boldsymbol{x}_{<i}; \mathsf{param}) \equiv_\epsilon \left(\mathcal{X}_i \mid \mathcal{X}_{<i} = \boldsymbol{x}_{<i} \right) \right] \;\geq\; 1 - \delta,$$

where i is uniform over $[m] := \{1, \ldots, m\}$, $\mathcal{X}_{<i} = (\mathcal{X}_1, \ldots, \mathcal{X}_{i-1})$, and the probability is over the choice of i and the randomness of Π used to generate $\boldsymbol{x}_{<i}$.

Our universal inductive inference algorithm is based on an efficient algorithm that approximates t-time-bounded Kolmogorov complexity K^t. We consider an assumption slightly weaker than GapMINKT $\in \mathsf{P}$—the assumption that there exists a randomized polynomial-time algorithm $\widetilde{K}$ such that for every input $x \in \{0,1\}^*$ of length n and every time bound $t \in \mathbb{N}$,

$$\mathrm{K}(x) \;\leq\; \widetilde{K}(x, 1^t) \;\leq\; \mathrm{K}^t(x) + O(\log n)$$

with high probability over the internal randomness of $\widetilde{K}$.[2] We denote this assumption by GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP. This worst-case assumption is equivalent to the existence of an errorless average-case algorithm for time-bounded Kolmogorov complexity with respect to the uniform distribution [Hir18; Hir20b; Hir20a; GKLO22], and, in particular, it follows from DistNP $\subseteq$ AvgBPP (see Appendix E).

Our main theorem is stated as follows.

[2] The difference $\mathrm{K}^t(x) - \mathrm{K}(x)$ is known as (basic) computational depth, and, in general, can be as large as n [AFMV06].

Theorem 2.1. *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then complexity-theoretic universal inductive inference is solvable in time*[3] $\mathsf{poly}(t)$ *with round complexity* $s \cdot \widetilde{O}\big(n^2\epsilon^{-6}\delta^{-5}\big)$.

In fact, the round complexity can be significantly improved, with the caveat that the inference algorithm below needs to know the total number of rounds, and its running time is larger.

Theorem 2.2. *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then when the total number of rounds* m *is explicitly provided to the inference algorithm, complexity-theoretic universal inductive inference is solvable in time* $t^{O(\delta^{-1})}$ *with round complexity* $O(s\,\epsilon^{-2}\delta^{-1})$.

In Appendix B, we observe that any (even time-unbounded) universal inductive inference algorithm requires $\Omega\big(s\,\epsilon^{-2} + s\,\delta^{-1}\big)$ rounds. Thus, for a constant confidence $\delta^{-1} = O(1)$, Theorem 2.2 provides a polynomial-time algorithm for complexity-theoretic universal inductive inference with the optimal round complexity $O(s\epsilon^{-2})$, which matches the information-theoretic lower bound up to a constant factor.

Using the results above, one can immediately answer Question 1.2 affirmatively, although with a large sample complexity or a long running time. We actually obtain an efficient distributional learner with sample complexity $O\big(s\,\epsilon^{-2}\,\log\delta^{-1}\big)$ via a non-trivial reduction that boosts the confidence of a distributional learner constructed from Theorem 2.2.

Corollary 2.1. *If* DistNP $\subseteq$ AvgBPP*, then there exists a randomized algorithm that for every distribution* $\mathcal{D}$ *sampled by a polynomial-time sampler of description length* s*, given as input* m *i.i.d. samples drawn from* $\mathcal{D}$*, outputs a circuit whose output distribution is* ϵ*-close to* $\mathcal{D}$ *with probability* $1-\delta$ *in time* $\mathsf{poly}(s, \epsilon^{-1}, \delta^{-1})$*, where* $m = O\big(s\,\epsilon^{-2}\,\log\delta^{-1}\big)$.

Furthermore, Corollary 2.1 improves the sample complexity of the prior work [HN21] from $\tilde{O}(s^3\epsilon^{-8}\log\delta^{-1})$ to $O(s\epsilon^{-2}\log\delta^{-1})$ for PAC learning over P/poly-samplable distributions. This follows from the reduction of Xiao [Xia10], which converts a distributional learner to a PAC learner while preserving sample complexity. However, it should be noted that the result of [HN21] and ours are actually incomparable because their learning algorithm solves *agnostic learning*, which is more general than PAC learning. Whether agnostic learning can be reduced to inductive inference remains open. It is an important open question to develop a common learning framework that unifies agnostic learning and inductive inference.

2.2 Unconditional Polynomial-Time Inductive Inference

It is of great importance to understand for which sequences inductive inference is possible in polynomial time *without any unproven complexity-theoretic assumption*. We establish a characterization of distributions over binary sequences for which there exists a polynomial-time inductive inference algorithm with constant accuracy and confidence by the notion of prequential compression [Daw84; DV99; Grü07].

We introduce the notion of (n,ϵ)-*prequential* q^t-*compression* for a distribution $\mathcal{X}$ over $\{0,1\}^N$. We partition a sequence $x \sim \mathcal{X}$ into m blocks $(y_1,\dots,y_m)$ of length n (and hence $m = \lceil N/n \rceil$ and the last block may be shorter than n). We measure the coding length of y_j given $y_{<j} \coloneqq (y_1,\dots,y_{j-1})$ using the information content q^t with respect to the t-time-bounded universal distribution Q^t [IL90; HN23]. Let $\mathrm{Q}^{t,z}$ be the t-*time-bounded universal distribution* with an advice string z, i.e., the output

[3] The running time depends only on t because t dominates other parameters under the requirement that an unknown Turing machine generates at least m symbols in time t, where $m \geq \Omega(s\epsilon^{-1}\delta^{-1})$.

distribution of a universal Turing machine U on a uniformly random input $\pi \sim \{0,1\}^t$ and an advice string z when run for t steps (see Section 4 for the formal definition). Define

$$\mathrm{q}^t(x \mid z) \coloneqq -\log \mathrm{Q}^{t,z}(x),$$

where $\mathrm{Q}^{t,z}(x)$ is the probability that x is sampled from $\mathrm{Q}^{t,z}$. When z is the empty string, we simply write $\mathrm{q}^t(x)$ instead of $\mathrm{q}^t(x \mid z)$. The information content $\mathrm{q}^t(x)$ of x with respect to Q^t is approximately equal to the t-time-bounded probabilistic Kolmogorov complexity $\mathsf{pK}^t(x)$ of a string x [GKLO22], which measures the shortest length of a program that prints x in time t given a public random string [HN23].

Definition 2.2. *For a function $t\colon \mathbb{N} \to \mathbb{N}$, we say that a distribution $\mathcal{X}$ over $\{0,1\}^N$ is (n,ϵ)-*prequential q^t-compressible *if*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}\left[\sum_{j=1}^{m} \mathrm{q}^{t(|y_{<j}|)}(y_j \mid y_{<j})\right] \le \epsilon N + \mathrm{H}(\mathcal{X}),$$

where $m \coloneqq \lceil N/n \rceil$ and y_j denotes the concatenation of the i-th bit of x for all $i \in \mathbb{N}$ such that $n(j-1) < i \le \min\{nj, N\}$, and $\mathrm{H}(\mathcal{X})$ denotes the Shannon entropy of a distribution $\mathcal{X}$.

Blier and Ollivier [BO18] empirically observed that deep learning models can compress the training data very well—even when accounting for the huge parameters of deep learning models—using the prequential compression with respect to the probability distribution induced by a deep learning model. Definition 2.2 is independent of a specific model and measures the coding length with respect to the t-time-bounded universal distribution. Using this definition, we provide a mathematical justification of the empirical observation of [BO18]: a distribution $\mathcal{X}$ of sequences is prequential compressible *if and only if* there exists an efficient inductive inference algorithm for $\mathcal{X}$. This indicates that the empirical observation of [BO18] is not only limited to deep learning models but also is a universal phenomenon applicable to every model.

Theorem 2.3 (see also Theorem 6.1)**.** *There exists a sequence $\{\mathcal{L}_{c,\epsilon}\}_{c,\epsilon^{-1}\in\mathbb{N}}$ of randomized polynomial-time algorithms such that for every family $\mathcal{X} = \{\mathcal{X}^N\}_{N\in\mathbb{N}}$ of distributions $\mathcal{X}^N$ over $\{0,1\}^N$, the following are equivalent.*

1. *For every constant $\epsilon > 0$, there exist constants $n \in \mathbb{N}$ and $c \in \mathbb{N}$ such that for all large $N \in \mathbb{N}$, $\mathcal{X}^N$ is (n,ϵ)-prequential q^t-compressible for $t(m) \coloneqq m^c$.*
2. *For all sufficiently small constants $\epsilon > 0$ and all $n \in \mathbb{N}$ with $n \ge (1/\epsilon)^{1.01}$, for all large $N \in \mathbb{N}$, $\mathcal{X}^N$ is (n,ϵ)-prequential rK^t-compressible for the constant function $t = N^{O(1)}$. Here, $\mathsf{rK}^t(x)$ is the t-time-bounded Kolmogorov complexity of x, i.e., the shortest length of a randomized program that prints x in time t with high probability.*
3. *For every constant $\epsilon > 0$, there exists a randomized polynomial-time algorithm L that predicts the next bits of $\mathcal{X}^N$ given its prefix with accuracy ϵ and confidence ϵ for all sufficiently large N. That is,*
$$\Pr_{\substack{i\sim[N]\\ x\sim\mathcal{X}^N}}\left[L(x_{<i}) \equiv_\epsilon \left(\mathcal{X}_i^N \mid \mathcal{X}_{<i}^N = x_{<i}\right)\right] \ge 1-\epsilon,$$
where $\mathcal{X}_i^N$ denotes the i-th bit of $\mathcal{X}^N$.
4. *Item 3 holds for every constant $\epsilon > 0$, for some $c \in \mathbb{N}$ and for $L \coloneqq \mathcal{L}_{c,\epsilon}$.*

This characterization carries a strong conceptual message: in the prequential setting, efficient prediction is equivalent to efficient compression.

The connection between compression and prediction has a long history in statistics and information theory. A line of recent work in machine learning revisits this perspective, typically using the direction from compression to prediction as a coding-theoretic interpretation or as an empirical evaluation principle for specific model classes [BO18; BLH23; DRDCGMGWAOHV24], rather than providing theoretical guarantees. In particular, Delétang, Ruoss, Duquenne, Catt, Genewein, Mattern, Grau-Moya, Wenliang, Aitchison, Orseau, Hutter, and Veness [DRDCGMGWAOHV24] explicitly note that there is no strong guarantee that a good compression rate leads to good autoregressive samples. There are also theorem-level results for restricted statistical model classes; for example, Han, Jiang, and Wu [HJW24] derive prediction guarantees for hidden Markov and renewal processes. In contrast, Theorem 2.3 provides an unconditional complexity-theoretic theorem in the polynomial-time regime: prequential compression with respect to time-bounded universal distributions *characterizes* the existence of a randomized polynomial-time predictor.

Note that the characterization in Theorem 2.3 concerns next-bit prediction with constant accuracy (equivalently, prediction of blocks of constant length). Extending this equivalence to polynomial-time prediction and compression for arbitrary n-bit blocks remains an important open problem.

2.3 Perspective: Towards a Practical Inductive Inference

A salient feature of the inductive inference algorithm $\mathcal{L} = \{\mathcal{L}_{c,\epsilon}\}$ of Theorem 2.3 is that the algorithm runs in polynomial time without any unproven assumption and is defined independently of a specific distribution $\mathcal{X}$ over sequences. Thus, in principle, the algorithm $\mathcal{L}$ can be implemented and run in practice. This is an *optimal* inductive inference algorithm in the sense that if there exists some efficient inference algorithm for $\mathcal{X}$, then $\mathcal{L}$ can also solve the inference task for $\mathcal{X}$. Moreover, it has the intriguing property that if GapMINKT $\in \mathsf{P}$ holds, then $\mathcal{L}$ solves the inductive inference task for any distribution $\mathcal{X}$ samplable by any efficient randomized Turing machine of small description length.[4]

One of the central themes in machine learning has been to understand the practical success of deep learning models, such as large language models, from a mathematical point of view. Our results provide a mathematical foundation for constructing efficient inference algorithms in a theoretically grounded manner. An intriguing direction for future research is to investigate the relationship between our optimal inference algorithm $\mathcal{L}$ and large language models. Although our algorithm $\mathcal{L}$ may be too slow to be practical due to a large hidden constant (originating from a universal Turing machine), it is theoretically optimal in the sense of Theorem 2.3. In contrast, large language models perform well in practice but may not be theoretically optimal. Is there an inference algorithm that can take the best of both worlds?[5]

2.4 Related Work

An implementation of Solomonoff's universal theory in a complexity-theoretic realm was first suggested by Impagliazzo and Levin [IL90]. A large part of their paper was devoted to proving that $\mathsf{DistNP} \not\subseteq \mathsf{HeurBPP}$ if and only if $\mathsf{NP} \times \{\mathcal{U}\} \not\subseteq \mathsf{HeurBPP}$ (i.e., the uniform distribution $\mathcal{U}$ is the hardest input distribution for NP among all the polynomial-time samplable distributions). It was alluded in the last section of their paper that a "universal inductive inference" is feasible if and only if

[4] This follows from Theorems 2.1 and 2.3.

[5] A related paper [GKKZ22] proposed a neural network architecture that can learn a constant-size program, but the architecture is not especially good in practice.

a one-way function—a fundamental cryptographic primitive—does not exist. Unfortunately, they did not provide the exact statement for universal inductive inference nor a proof for it, except that [IL90, Proposition 1] claims—without a proof—the equivalence between the non-existence of one-way functions and the existence of an efficient algorithm that approximates $\mathrm{Q}^t(x)$ on average when x is sampled from a t-time-samplable distribution. Because of this, their ideas were not well understood in the community. Meanwhile, Liu and Pass [LP20] established the equivalence between the non-existence of one-way functions and the existence of an efficient average-case algorithm for $\mathrm{K}^t(x)$ for a uniformly random input x. It was only recently that Hirahara and Nanashima [HN23] implemented the ideas of Impagliazzo and Levin [IL90], and proved the equivalence between the non-existence of one-way functions and the existence of an efficient *average-case universal* inductive inference. This algorithm solves inductive inference for a random sequence x generated in the following two steps: a constant-size randomized program Π_0 efficiently generates another efficient randomized Turing machine Π with an advice string, and Π generates a sequence x randomly. The average-case inference algorithm is required to succeed for most sequences x generated in this way, and is "average-case universal" in that it works for every *constant-size* program Π_0; however, it does not work for every environment Π. Indeed, the running time of their algorithm is characterized by $2^{O(\mathrm{cd}^t(\Pi))}$ in the worst case, where $\mathrm{cd}^t(\Pi) \coloneqq \mathrm{q}^t(\Pi) - \mathrm{K}(\Pi)$ is called the *computational depth of* Π, and there are exponentially many strings with computational depth $\Theta(|\Pi|)$ [AFMV06].

To summarize the results in the previous paragraph, the following are equivalent: (1) one-way functions do not exist; (2) there exists a polynomial-time algorithm that computes $\mathrm{K}^t(x)$ on average over a uniformly random x; and (3) there exists a polynomial-time average-case universal inductive inference algorithm. The implication from (2) to (3) establishes a complexity-theoretic counterpart of Solomonoff's inductive inference in the setting of *average-case complexity*. This differs from Theorem 2.1, which establishes an analogous result in the setting of *worst-case complexity*. Arguably, our inference algorithm is universal in the same spirit of Solomonoff's universal inductive inference because it works for random sequences generated by *every* environment Π, as long as the number of rounds is sufficiently larger than the description length of Π. It should also be noted that the previous results [IL90; LP20; HN23] provide error-prone-average-case to error-prone-average-case reductions, whereas ours provide worst-case to errorless-average-case reductions (because $\mathrm{GapK^t}$-vs-$\mathrm{K} \in \mathsf{pr\text{-}BPP}$ is equivalent to the existence of an errorless average-case polynomial-time algorithm for K^t). It is an important open question whether the gap between error-prone and errorless average-case complexities can be closed [HS22; HN22].

Naor and Rothblum [NR06] introduced the task of *learning adaptively changing distributions* in the setting of average-case complexity, and characterized its hardness by the existence of one-way functions. The task is a special case of the average-case inductive inference task where the algorithm is allowed to choose a single round i so that the output distribution x'_i of the algorithm is statistically close to the conditional distribution of x_i given $x_{<i}$. The inference algorithm of Theorem 2.1 solves a task more general than the worst-case variant of learning adaptively changing distributions: The inference algorithm succeeds in almost all rounds i, not just one round chosen by an algorithm.

3 Technical Overview

In this section, we present the key ideas underlying our inference algorithm. Section 3.1 introduces a key lemma for Theorems 2.1 and 2.2. We present an exposition of Solomonoff's inductive inference in Section 3.2. We sketch the proofs of Theorem 2.3, Theorem 2.2, and Theorem 2.1 in Sections 3.3 to 3.5, respectively.

3.1 Algorithmic Proof of Chain Rule

Our characterization (Theorem 2.3) suggests that efficient inductive inference is impossible for any prequential-incompressible distributions, and that it is *necessary* to construct a prequential compressor for inductive inference to be feasible. This is indeed the approach we (need to) take, and a key lemma behind Theorems 2.1 and 2.2 enables us to construct a prequential compressor for every distribution generated by an unknown efficient Turing machine under the assumption that $\mathrm{GapK}^{\mathrm{t}}\text{-vs-K} \in \mathsf{pr\text{-}BPP}$. The lemma, stated below, establishes a time-bounded analogue of the chain rule for K [ZL70]. In fact, Theorems 2.1 and 2.2 hold under the weaker assumption that the chain rule below holds; see Remark 7.1.

Lemma 3.1 (Chain rule for q^t)**.** *If* $\mathrm{GapK}^{\mathrm{t}}\text{-vs-K} \in \mathsf{pr\text{-}BPP}$*, then for every string* $\boldsymbol{x} = (x_1, \ldots, x_m)$*, where each* $x_i \in \{0,1\}^{\leq n}$*, and for every* $t \geq n + m$*, we have*

$$\sum_{i=1}^{m} \mathrm{q}^{\mathsf{poly}(t)}(x_i \mid \boldsymbol{x}_{<i}) \leq \mathrm{q}^t(\boldsymbol{x}) + m \cdot O(\log t),$$

where $\boldsymbol{x}_{<i} = (x_1, \ldots, x_{i-1})$.

This lemma demonstrates the existence of a prequential compressor that takes $\boldsymbol{x}_{<i}$ and x_i as input and compresses x_i so that the total coding length is at most $\mathrm{q}^t(\boldsymbol{x}) + m \cdot O(\log t)$. Note that this is optimal up to the additive term $m \cdot O(\log t)$ because $\mathrm{q}^t(\boldsymbol{x})$ is the lower bound of the coding length of any t-time decoding algorithm.

The special case of $m = 2$ is referred as symmetry of information. Symmetry of information for time-bounded Kolmogorov complexity [LM93; LW95] has been studied in the literature of meta-complexity (e.g., [Hir21; Hir22b; GK21; HILNO23; Ila23; HLO24; HLN24]). Lemma 3.1 generalizes symmetry of information for time-bounded Kolmogorov complexity proved in [Hir22b; GK21].

In fact, a straightforward extension of the previous proofs yields an additive term $m^2 \cdot O(\log t)$, which is insufficient for our applications. At a high level, the difficulty is that a prequential compressor needs to efficiently estimate $k_i \approx \mathrm{q}^{\mathsf{poly}(t)}(x_i \mid \boldsymbol{x}_{<i})$ in order to carry out a hybrid argument; see Appendix A for details. Avoiding this quadratic loss is one of our main technical contributions. We achieve this by presenting an *algorithmic* proof of the chain rule: We present an efficient algorithm that, given $x_1, \ldots, x_i$ as input, outputs a value k_i such that for every $i \in [m]$,

$$k_i \geq \mathrm{q}^{\mathsf{poly}(t)}(x_i \mid \boldsymbol{x}_{<i})$$

and

$$\sum_{i=1}^{m} k_i \leq \mathrm{q}^t(\boldsymbol{x}) + m \cdot O(\log t).$$

Lemma 3.1 follows immediately from this algorithmic proof. In fact, the values $(k_1, \ldots, k_m)$ algorithmically estimated will be crucial for the proof of Theorem 2.1. The proof of the chain rule can be found in Section 5.

We mention in passing that our algorithmic proof of the chain rule is fundamentally different from the previous proofs [Hir22b; GK21] in that our prequential compressor makes *adaptive* queries to an oracle that solves $\mathrm{GapK}^{\mathrm{t}}$-vs-K. This is particularly important in the literature of average-case complexity because it may bypass the limits of *nonadaptive* reductions shown by Bogdanov and Trevisan [BT06].

Independently of this work, Kabanets and Kolokolova [KK25] presented the equivalence between a conditional variant of a chain rule for $\mathsf{pK}^{\mathsf{poly}}$ and the existence of an efficient algorithm that approximates the conditional $\mathsf{pK}^{\mathsf{poly}}$-complexity.

From the Chain Rule to Prequential Compression. Prequential compressibility of every distribution $\mathcal{X}$ sampled by an efficient Turing machine Π of description length s immediately follows from the chain rule (Lemma 3.1) and the coding property of $\mathrm{q}^{\mathsf{poly}}$. The *coding property of* $\mathrm{q}^{\mathsf{poly}}$, also known as the domination property, states that for every x in the support of $\mathcal{X}$,

$$\mathrm{q}^t(x) \leq -\log \Pr[\mathcal{X} = x] + s + O(1)$$

when t is sufficiently larger than the running time of Π. This follows from the simple fact that a uniformly random program coincides with the description of Π with probability 2^{-s}, in which case the universal Turing machine simulates exactlyΠ; see Proposition 4.4. Taking the expectation of the inequality over $x \sim \mathcal{X}$ and applying Lemma 3.1, we obtain (up to negligible terms)

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}\left[\sum_i \mathrm{q}^{\mathsf{poly}(t)}(x_i \mid x_{<i})\right] \lesssim \mathop{\mathbb{E}}_{x\sim\mathcal{X}}\left[\mathrm{q}^t(x)\right] \lesssim \mathrm{H}(\mathcal{X}) + s,$$

which shows that $\mathcal{X}$ is prequential compressible. According to Theorem 2.3, this suffices for efficient inductive inference with constant accuracy and confidence. To explain why this is the case, we need to review Solomonoff's fundamental ideas.

3.2 Review: An Exposition of Solomonoff's Inductive Inference

Our inference algorithms build on Solomonoff's inductive inference [Sol64a] and its complexity-theoretic implementation in the average-case setting [IL90; HN23]. A time-bounded variant of Solomonoff's inductive inference can be viewed as a form of Bayesian inference with prior Q^t. Given the previously observed data stream $\boldsymbol{x}_{<i} = (x_1, \ldots, x_{i-1})$, the inference algorithm samples a prediction $\tilde{x}_i$ for the next symbol from the conditional (posterior) distribution of Q^t given the prefix $\boldsymbol{x}_{<i}$. Following [IL90; HN23], we refer to this sampling process as *universal extrapolation*. We also call the conditional prefix $\boldsymbol{x}_{<i}$ the *context* of the universal extrapolation, following a similar notion used in the literature on large language models.

The reason why universal extrapolation solves the task of universal inductive inference can be elegantly explained by the chain rule for the KL divergence. Let $\mathcal{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_m)$ be the joint distribution of m strings produced by an unknown t-time sampler Π with description size s. By the coding property of q^t,

$$\mathrm{KL}\big(\mathcal{X} \,\big\|\, \mathrm{Q}^t\big) \coloneqq \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{X}}\left[\log \frac{\Pr[\mathcal{X} = \boldsymbol{x}]}{\Pr[\mathrm{Q}^t = \boldsymbol{x}]}\right] \leq s + O(1).$$

By the chain rule for KL divergence (see Lemma 4.1),

$$\frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{X}}\left[\mathrm{KL}\big(\mathcal{X}_i \mid \mathcal{X}_{<i} = \boldsymbol{x}_{<i} \,\big\|\, \mathrm{Q}_i^t \mid \mathrm{Q}_{<i}^t = \boldsymbol{x}_{<i}\big)\right] = \frac{1}{m}\mathrm{KL}\big(\mathcal{X} \,\big\|\, \mathrm{Q}^t\big) \leq \frac{O(s)}{m},$$

where Q_i^t denotes the marginal distribution of the i-th string generated by Q^t. Hence, if $m \gg s$, the amortized KL divergence from the true conditional distribution to the inferred one at each position i vanishes asymptotically. By Pinsker's inequality (Proposition 4.1), this implies that the conditional distribution $(\mathcal{X}_i \mid \mathcal{X}_{<i} = \boldsymbol{x}_{<i})$ is statistically close to the distribution $(\mathrm{Q}_i^t \mid \mathrm{Q}_{<i}^t = \boldsymbol{x}_{<i})$ of universal extrapolation with context $\boldsymbol{x}_{<i}$, and thus sampling from the latter solves the task of inductive inference.

Previous work: inference for computationally shallow inputs. The previous work [IL90; HN23] in the literature presented an efficient average-case algorithm for universal extrapolation assuming the non-existence of one-way functions (which is weaker than GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP). In fact, building on [AFMV06; AF09], Hirahara and Nanashima [HN23] constructed a worst-case algorithm that samples from $(\mathrm{Q}_i^t \mid \mathrm{Q}_{<i}^t = \boldsymbol{x}_{<i})$ on input $\boldsymbol{x}_{<i}$ and runs in time $2^{O(\mathrm{cd}^t(\boldsymbol{x}_{<i})+\log t)}$, where $\mathrm{cd}^t(x)$ denotes the computational depth of x defined as

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

The *slow growth law* (see Lemma 4.6)states that no algorithm can rapidly increases computational depth, and hence the algorithm of [HN23] is efficient on most instances generated by an efficient constant-size program. However, this algorithm is no better than a brute force search when the input x is computationally deep, i.e., $\mathrm{cd}^t(x) \approx |x|$.

3.3 Polynomial-Time Inductive Inference from Prequential Compression

A new idea of this paper, significantly different from the previous ideas in the literature [IL90; HN23], is to shrink the context $\boldsymbol{x}_{<i}$ by "moving its prefix to an advice string". This (simple but important) idea underlies all the results of this paper, and we illustrate it by sketching the proof of Theorem 2.3.

To prove the implication from Item 1 to Item 3 of Theorem 2.3, we need to construct a polynomial-time inductive inference algorithm $\mathcal{L}$ that predicts the next bits of a (w,ϵ)-prequential q^t-compressible distribution $\mathcal{X}$ over $\{0,1\}^N$. The inference algorithm $\mathcal{L}$ takes an input $\boldsymbol{x}_{<i} = (x_1,\dots,x_{i-1}) \in \{0,1\}^{i-1}$, partitions it into consecutive blocks $y_1,\dots,y_j$ so that $|y_1| = \cdots = |y_{j-1}| = w$ and $|y_j| < w$, and then executes the "advised universal extrapolation" with context y_j and advice $y_{<j} \coloneqq (y_1,\dots,y_{j-1})$. The *advised universal extrapolation* refers to sampling z from the t-time-bounded universal distribution $\mathrm{Q}^{t,y_{<j}}$ with the advice string $y_{<j}$ conditioned that y_j is the prefix of z.

The significant advantage of advised universal extrapolation over the standard one is that it can be implemented in polynomial time when the context length is $O(\log n)$: indeed, one can use rejection sampling, that is, repeatedly sample from $\mathrm{Q}^{t,y_{<j}}$ until the condition that y_j is a prefix is satisfied, which happens with probability $2^{-|y_j|-O(1)}$. The disadvantage of advised universal extrapolation is that it may not solve the task of inductive inference for all efficiently samplable distributions, yet we prove it for all prequential compressible distributions $\mathcal{X}$.

To see why $\mathcal{L}$ is an inductive inference algorithm for a prequential compressible distribution $\mathcal{X}$, let Y_j denote the j-th block of $x \sim \mathcal{X}$ for each $j \in [m]$ and observe that

$$\begin{aligned}\frac{1}{m}\sum_{j=1}^{m} \mathrm{KL}\big(Y_j \mid Y_{<j} \,\big\|\, \mathrm{Q}^{t,Y_{<j}} \mid Y_{<j}\big) &= \frac{1}{m}\left(\mathbb{E}\left[\sum_{j=1}^{m} \mathrm{q}^t(Y_j \mid Y_{<j})\right] - \mathrm{H}(\mathcal{X})\right)\\ &\le \frac{\epsilon N}{m} \le \epsilon w,\end{aligned}$$

where the equality holds by the definition of the KL divergence, the first inequality follows from the definition of the (w,ϵ)-prequential q^t-compression, and the last inequality holds because $m = \lceil N/w \rceil$. This shows that the amortized KL divergence from $(Y_j \mid Y_{<j})$ to $(\mathrm{Q}^{t,Y_{<j}} \mid Y_{<j})$ is at most ϵw. Applying the chain rule for the KL divergence to each block of length w as in Section 3.2, the KL divergence from $(\mathcal{X}_i \mid \mathcal{X}_{<i})$ to advised universal extrapolation is bounded by ϵ, which establishes the accuracy of advised universal extrapolation.

We briefly mention that a strong converse holds: the existence of an efficient inductive inference algorithm implies not only prequential q^t-compression but also prequential rK^t-compression (Item 3 $\Longrightarrow$ Item 2 in Theorem 2.3). In contrast to $\mathrm{q}^t(x)$ and $\mathsf{pK}^t(x)$, which essentially refer to the shortest length of a compressed string *in the presence of public random strings*, the t-time-bounded randomized Kolmogorov complexity $\mathsf{rK}^t(x)$ of x is the shortest length of a compressed string that can be decoded to x in time t with high probability over the internal randomness of the decoder; arguably, the latter is a more natural notion of compression. To compress a string in the sense of rK^t, we employ the pseudo-deterministic version of arithmetic coding, which was recently developed by Hirahara, Lu, and Nanashima [HLN24].

Details of Theorem 2.3 and its proof can be found in Section 6.

3.4 Efficient Advised Universal Extrapolation

In order to prove Theorem 2.2, we combine the new idea of shrinking contexts with the previous result of [HN23]. The rejection sampling algorithm for advised universal extrapolation sketched in Section 3.3 has the significant limitation that it is efficient only for short contexts. We overcome this limitation by inverting an auxiliary-input one-way function—whose security is weaker than that of one-way functions. An auxiliary-input one-way function is a family of functions indexed by an auxiliary input z that is hard to invert on average for some z. It is known that every auxiliary-input one-way function is efficiently invertible under the assumption that $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$. Applying the idea of [HN23] to auxiliary-input one-way functions, we can obtain an algorithm that executes advised universal extrapolation with context x and advice z in time $2^{O(\mathrm{cd}^t(x|z)+\log t)}$ for every x and every z, where $\mathrm{cd}^t(x \mid z) \coloneqq \mathrm{q}^t(x \mid z) - \mathrm{K}(x \mid z)$ denotes the conditional computational depth. We then use the chain rule for computational depth to guarantee that most blocks are computationally shallow given advice.

Corollary 3.1 (Chain rule for computational depth)**.** *If* $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$*, then for every string* $\boldsymbol{x} = (x_1, \dots, x_m)$*, where each* $x_i \in \{0,1\}^{\le n}$*, and for every* $t \ge n + m$*, we have*

$$\sum_{i=1}^{m} \mathrm{cd}^{\mathsf{poly}(t)}(x_i \mid \boldsymbol{x}_{<i}) \le \mathrm{cd}^t(\boldsymbol{x}) + m \cdot O(\log t).$$

This immediately follows from Lemma 3.1 together with $\mathrm{K}(\boldsymbol{x}) \le \sum_{i=1}^{m} \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + m \cdot O(\log t)$.

By the slow growth law, if the entire sequence $\boldsymbol{x}$ is generated by a t-time sampling algorithm with an s-bit description, then $\mathrm{cd}^{\mathsf{poly}(t)}(\boldsymbol{x}) \le O(s)$ holds with high probability. Therefore, if $m \ge s/\log t$, the chain rule implies that the expectation of $\mathrm{cd}^{\mathsf{poly}(t)}(x_i \mid \boldsymbol{x}_{<i})$ over a uniformly random position $i \sim [m]$ is at most $O(s)/m \le O(\log t)$. By Markov's inequality, for every $\delta \in (0,1]$,

$$\Pr_{i \sim [m]} \left[\mathrm{cd}^{\mathsf{poly}(t)}(x_i \mid \boldsymbol{x}_{<i}) \le \delta^{-1} \cdot O(\log t) \right] \ \ge \ 1 - \delta. \tag{1}$$

This means that the advised universal extrapolation algorithm runs in time $2^{O(\delta^{-1} \cdot \log t)} = t^{O(1/\delta)}$ for a $(1-\delta)$-fraction of blocks $i \sim [m]$.

In our inference algorithm of Theorem 2.2, we partition the sequence $x_1, \dots, x_\sigma \in \{0,1\}^n$ of symbols into blocks $y_1, \dots, y_m \in (\{0,1\}^n)^w$, where each block consists of w symbols. To predict the next symbol, we apply the advised universal extrapolation with context y_j and advice $\boldsymbol{y}_{<j}$. Applying Corollary 3.1 to the blocks $(y_1, \dots, y_m)$, the advised universal extrapolation works correctly for a $(1-\delta)$-fraction of blocks in time $t^{O(1/\delta)}$ if $m \ge s/\log t$. By arguments similar to Section 3.3, the "block-wise" KL divergence from the target distribution to the advised universal distribution is at

most $O(s)/m \leq O(\log t)$. Applying the chain rule for the KL divergence inside each block, the amortized per-symbol KL divergence to the advised universal extrapolation is at most $O(\log t)/w \leq \epsilon^2\delta$, where we choose $w \coloneqq \epsilon^{-2}\delta^{-1} \cdot O(\log t)$ so that this inequality holds. By Markov's inequality, the per-symbol KL divergence is at most ϵ^2 for a $(1-\delta)$-fraction of positions inside a block. By Pinsker's inequality, the total variation distance is at most ϵ.

Putting this together, the inference algorithm requires at least $m \geq s/\log t$ blocks of width $w = \epsilon^{-2}\delta^{-1} \cdot O(\log t)$. This condition is satisfied when the number σ of rounds is larger than $(s/\log t) \cdot w = O(s\epsilon^{-2}\delta^{-1})$. Details can be found in Section 7.

3.5 Fully Polynomial-Time Universal Inductive Inference

Next, we present the ideas for the proof of Theorem 2.1, which constructs a universal inductive inference algorithm that runs in *fully polynomial time* in all parameters. A high level idea is to boost the confidence level of the algorithm of Theorem 2.2, which runs in polynomial time if the confidence parameter δ is constant.

For the purpose of the exposition, we first make an ideal assumption. We assume that the conditional computational depth $\mathrm{cd}^t(x \mid y)$ is efficiently computable. This assumption helps to illustrate the ideas in Section 3.5.1, although the assumption is provably false because resource-unbounded Kolmogorov complexity is not computable. Removing this false assumption is one of our main technical contributions, which we describe in Section 3.5.2.

3.5.1 Confidence Boosting via Merge-Segmentation

Assuming that the conditional computational depth is efficiently computable, we obtain an *errorless* efficient algorithm for advised universal extrapolation: The algorithm either outputs a special symbol $\perp$ indicating the failure of an algorithm (when the input is computationally deep), or otherwise it correctly executes the advised universal extrapolation. Note that the universal extrapolation algorithm of [HN23] is *error-prone*, i.e., the algorithm does not know when it makes a mistake, because it is based on inverting one-way functions in the sense of error-prone average-case complexity.

As in Section 3.4, we partition the input sequence $x_1, \dots, x_\sigma$ into blocks $\boldsymbol{y} = (y_1, \dots, y_m)$. By Corollary 3.1, a $\frac{3}{4}$-fraction of blocks satisfy $\mathrm{cd}^t(y_j \mid \boldsymbol{y}_{<j}) \leq O(\log t)$. We call such a block *good*. For good blocks, the advised universal extrapolation algorithm successfully predicts the next symbols. Our idea is to iteratively use this argument to boost the confidence.

In more detail, our confidence boosting procedure starts with the initial sequence $\boldsymbol{y}^1 = (y_1, \dots, y_m)$ of blocks. Next, we consider a "resegmentation" of blocks. We call a sequence $\boldsymbol{z} = (z_1, \dots, z_b)$ of blocks a *resegmentation* of $\boldsymbol{y}$ if the concatenation of $\boldsymbol{z}$ is equal to that of $\boldsymbol{y}$. We construct the resegmentation $\boldsymbol{y}^2$ of $\boldsymbol{y}^1$ by merging all the consecutive good blocks. We repeat this process and, for each round r, construct a resegmentation $\boldsymbol{y}^{r+1}$ of $\boldsymbol{y}^r$ by merging all the consecutive good blocks. We stop the process after $R \coloneqq \log \delta^{-1}$ rounds.

Since a $\frac{3}{4}$-fraction of blocks in $\boldsymbol{y}^1$ are good, the number of blocks in $\boldsymbol{y}^2$ is at most $m/2$. Similarly, in each round, the number of blocks decreases by half. After $R = \log \delta^{-1}$ rounds, the number of blocks is at most δm. Thus, a $(1-\delta)$ fraction of the blocks in $\boldsymbol{y}$ are good in some round. This enables us to construct R inference algorithms, one of which is correct for a $(1-\delta)$ fraction of $\boldsymbol{y}$. Specifically, for each $r \in [R]$, the r-th inference algorithm uses the advised universal extrapolator with the advice of (some prefix of) $\boldsymbol{y}^r$. The final inference algorithm takes the uniform mixture of the predictions made by the R inference algorithms.

In the outline above, we have omitted many technical details. To mention a few of them: (1) Why is a $\frac{3}{4}$-fraction of $\boldsymbol{y}^r$ good? (2) How can we compute the resegmentation in an online manner?

(3) How can we take the uniform mixture of the R predictions? We briefly describe how to deal with these issues below.

The issue (1) can be addressed using the slow growth low. Let $\boldsymbol{y}^r = (y_1^r, \dots, y_b^r)$. Observe that $\boldsymbol{y}^1$ can be efficiently converted to $\boldsymbol{y}^r$ using $O(b \log m)$ bits of information that specifies the way of segmentation, from which the slow growth law implies

$$\mathrm{cd}^{\mathsf{poly}(t)}(\boldsymbol{y}^r) \;\le\; \mathrm{cd}^t(\boldsymbol{y}^1) + O(b \log m).$$

Hence, the amortized conditional computational depth of $\boldsymbol{y}^r$ per block remains $O\left((s + b \log m)/b\right) \le O(\log t)$, as long as $b \ge s/\log t$. This implies that a $\frac{3}{4}$-fraction of the merged blocks in $\boldsymbol{y}^r$ are good.

The issue (2) can be addressed as follows. We claim that, for any block y_i to be predicted, we can compute from $\boldsymbol{y}_{<i}$ a list of R resegmentations of $\boldsymbol{y}_{<i}$ such that one of them matches the resegmentation at the first round r in which y_i becomes good (if such a round exists). This is obtained by simulating the merging process for R rounds on $\boldsymbol{y}_{<i}$ (rather than on the entire sequence). The claim holds because, up to round r, the block y_i is not yet good, hence not merged; therefore y_i and the subsequent blocks do not influence the process on $\boldsymbol{y}_{<i}$, and the simulation coincides with the original process up to round r.

The issue (3) can be addressed by the property of the universal extrapolator of [IL90], which sequentially predicts the next bit. This enables us to combine such predictors on the fly.

3.5.2 Identifying Good Blocks with the Algorithmic Proof of Chain Rule

The remaining and most technical challenge is to remove the false assumption that conditional computational depth is efficiently computable, which enabled us to verify whether a block is good or not in the argument above.

We change the definition of good blocks so that one can efficiently check whether a block is good or not, using the algorithmic proof of the chain rule for q^t. We say that a block y_i is *good* if the coding length $\tilde{q}^{\mathsf{poly}}(y_i \mid \boldsymbol{y}_{<i})$ according to the universal extrapolation algorithm (i.e., the negative logarithm of the probability that the algorithm outputs y_i given $\boldsymbol{y}_{<i}$ as advice) does not exceed k_i computed from the algorithmic proof of the chain rule in Section 3.1, up to an additive constant. As mentioned in Section 3.1, each k_i is efficiently computable from $\boldsymbol{y}_{<i}$ and y_i. Moreover, $\tilde{q}^{\mathsf{poly}}(y_i \mid \boldsymbol{y}_{<i})$ is also computable in the worst case from $\boldsymbol{y}_{<i}$ and y_i because the universal extrapolator outputs the next-bit conditional probability.

We need to verify two properties: (completeness) why a constant fraction of blocks is good in each round of merging, and (soundness) why inference is accurate for good blocks.

Completeness. The completeness follows from the fact that a $\frac{3}{4}$-fraction of blocks in $\boldsymbol{y}^r = (y_1^r, \dots, y_b^r)$ have conditional computational depth at most $O(\log t)$. For such blocks y_i^r, the universal extrapolator correctly simulates the universal distribution $\mathrm{Q}^{\mathsf{poly}, \boldsymbol{y}^r_{<i}}$, implying that the assigned coding length $\tilde{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i})$ is close to $\mathrm{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i})$. By contrast, k_i computed from the algorithmic proof of the chain rule satisfies $k_i \ge \mathrm{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i})$ for each $i \in [b]$. Hence, at least a $\frac{3}{4}$-fraction of the blocks satisfy

$$\tilde{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i}) \approx \mathrm{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i}) \le k_i,$$

and are therefore good according to the new definition.

Soundness. To see the soundness, assume that the i-th block is good, i.e., $\tilde{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i}) \lesssim k_i$. By an argument similar to the one using Markov's inequality to obtain the tail bound for $\mathrm{cd}^{\mathsf{poly}}$ (Equation (1)), we may observe that $k_i \lesssim \mathrm{K}(y_i^r \mid \boldsymbol{y}^r_{<i})$ for almost all i. Thus we obtain $\tilde{q}^{\mathsf{poly}}(y_i^r \mid \boldsymbol{y}^r_{<i}) \lesssim \mathrm{K}(y_i^r \mid \boldsymbol{y}^r_{<i})$, which indicates that the universal extrapolator assigns the nearly optimal coding length. This enables us to bound the KL divergence.

Details can be found in Section 8.

3.6 Improving Time and Sample Complexity in IID Cases

The i.i.d. learning setting of Corollary 2.1 is a special case of the foregoing non-i.i.d. scenario. To obtain fully polynomial running time and sample complexity $O(s\epsilon^{-2}\log\delta^{-1})$, we run the algorithm from Theorem 2.2 for $O(\log\delta^{-1})$ rounds at a sufficiently small *constant* confidence error and then select a hypothesis by clustering. Concretely, among the $O(\log\delta^{-1})$ candidate samplers we identify the largest cluster of pairwise close distributions in statistical distance and output any member. The clustering step uses the algorithm of Naor and Rothblum [NR06] for approximating the statistical distance between samplers.

Organization of this paper. The remainder of this paper is organized as follows. In Section 4, we fix notation and present the necessary preliminaries. In Section 5, we give an algorithmic proof of the chain rule (Lemma 3.1) and derive Corollary 3.1 as a corollary. In Section 6, we establish the equivalence between prequential compression and efficient inductive inference. In Section 7, we establish Theorem 2.2 using the ideas introduced in Section 3.4. In Section 8, we prove Theorem 2.1 building on the confidence-boosting framework of Section 3.5. Finally, we derive Corollary 2.1 in Section 9.

4 Preliminaries

All logarithms are base 2 unless stated otherwise. Let $\langle , \rangle$ be a (standard) pairing function that maps $\mathbb{N}\times\mathbb{N}$ to $\mathbb{N}$.

We use the notation negl to represent some negligible function, i.e., for any polynomial p and sufficiently large $n\in\mathbb{N}$, it holds that $\mathsf{negl}(n) < 1/p(n)$. We also use the notation poly to refer to some polynomial.

For each $n\in\mathbb{N}$, define $[n] := \{1,2,\ldots,n\}$. For every $x,y\in\{0,1\}^*$, let $x\circ y$ denote the concatenation of x and y. For readability, we may omit the symbol $\circ$ and write simply xy.

For $k\le k'\le n$ and a string $x\in\{0,1\}^n$ with i-th bit denoted x_i, define $x_{[k]} := x_1\cdots x_k$ and $x_{[k:k']} := x_k\cdots x_{k'}$.

For a sequence of strings $x_1, x_2, \ldots, x_i, \ldots \in \{0,1\}^*$, we define $\boldsymbol{x}_{<i} := \langle x_1,\ldots,x_{i-1}\rangle$, i.e., the binary encoding of the sequence of the first $i-1$ strings. We also let $\boldsymbol{x}_{\le i} := \langle x_1,\ldots,x_i\rangle$. For a sequence of strings $\boldsymbol{x}$, we let $|\boldsymbol{x}|$ denote the length of its binary encoding.

We define the flattening operator $\flat$ as follows: for each $x^1 = \langle x^1_1,\ldots,x^1_{k_1}\rangle, \ldots, x^m = \langle x^m_1,\ldots,x^m_{k_m}\rangle$, where each $x^i_j\in\{0,1\}$, we set

$$\flat\langle x^1,\ldots,x^m\rangle := \langle x^1_1,\ldots,x^1_{k_1}, x^2_1,\ldots,x^2_{k_2}, \ldots, x^m_1,\ldots,x^m_{k_m}\rangle.$$

For $n\in\mathbb{N}$, we use the notation "for every $t\ge O(n)$" to mean that there exists a universal constant $c>0$ (independent of n) such that the statement holds for all $t\ge cn$. In particular, for every $x\in\{0,1\}^*$ and every $t\ge O(|x|)$, it holds that $\mathrm{K}^t(x)\le |x| + O(1)\le 2|x|$.

For any distribution $\mathcal{D}$, we use the notation $x\sim\mathcal{D}$ to refer to the sampling of x according to $\mathcal{D}$. For any finite set S, we use the notation $x\sim S$ to refer to the uniform sampling of x from S. For simplicity, we may identify a distribution $\mathcal{D}$ with a random variable drawn from $\mathcal{D}$.

For a distribution $\mathcal{D}$ and an oracle machine M, we write $M^{\mathcal{D}}$ to indicate that M has oracle access to $\mathcal{D}$, where each oracle query returns an independent sample $x\sim\mathcal{D}$.

For any distribution $\mathcal{D}$ over strings and any $x\in\{0,1\}^*$, let $\mathcal{D}(x)$ denote the probability that x is sampled according to $\mathcal{D}$. We also use $\mathcal{D}(x*)$ to represent

$$\mathcal{D}(x*) \;=\; \sum_{y\in\{0,1\}^*} \mathcal{D}(x\circ y).$$

Probability Theory. In this paper, we assume basic knowledge of probability theory, including the union bound, Markov's inequality, Jensen's inequality, and Hoeffding's inequality. For an event E where trials to determine whether E occurs are repeated efficiently, we say that an algorithm M performs the empirical estimation of the probability that E occurs with accuracy error $\varepsilon \in [0,1]$ and confidence error $\delta \in [0,1]$ if M computes a value v with $\Pr[E] - \varepsilon \leq v \leq \Pr[E] + \varepsilon$ with probability at least $1 - \delta$ over trials. By Hoeffding's inequality, only $O(\varepsilon^{-2} \log \delta^{-1})$ are needed for such estimation.

For any distributions $\mathcal{D}$ and $\mathcal{E}$, let $\Delta_{\mathrm{tv}}(\mathcal{D}, \mathcal{E})$ denote the total variation distance between $\mathcal{D}$ and $\mathcal{E}$. Let $\mathrm{KL}(\mathcal{D}||\mathcal{E})$ represent the KL divergence between two distributions $\mathcal{D}$ and $\mathcal{E}$.

We review conditional KL divergence and the chain rule for KL divergence.

Definition 4.1 (Conditional KL divergence)**.** *For random variables $(\mathcal{X}, \mathcal{X}')$ and $(\mathcal{Y}, \mathcal{Y}')$, the conditional KL divergence from $\mathcal{X}'|\mathcal{X}$ to $\mathcal{Y}'|\mathcal{Y}$ is defined as*

$$\mathrm{KL}((\mathcal{X}'|\mathcal{X})||(\mathcal{Y}'|\mathcal{Y})) = \mathop{\mathbb{E}}_{(x,x') \sim (\mathcal{X}, \mathcal{X}')} \left[\log \frac{\Pr[\mathcal{X}' = x' | \mathcal{X} = x]}{\Pr[\mathcal{Y}' = x' | \mathcal{Y} = x]} \right].$$

Lemma 4.1 (Chain rule for KL divergence [cf. CT06, Theorem 2.5.3])**.** *For any random variables $(\mathcal{X}, \mathcal{X}')$ and $(\mathcal{Y}, \mathcal{Y}')$, it holds that*

$$\mathrm{KL}(\mathcal{X}, \mathcal{X}'||\mathcal{Y}, \mathcal{Y}') = \mathrm{KL}(\mathcal{X}||\mathcal{Y}) + \mathrm{KL}((\mathcal{X}'|\mathcal{X})||(\mathcal{Y}'|\mathcal{Y})).$$

In particular, for any $m \in \mathbb{N}$ and any random variables $(\mathcal{X}^1, \ldots, \mathcal{X}^m)$ and $(\mathcal{Y}^1, \ldots, \mathcal{Y}^m)$,

$$\mathrm{KL}(\mathcal{X}^1, \ldots, \mathcal{X}^m||\mathcal{Y}^1, \ldots, \mathcal{Y}^m) = \sum_{i=1}^{m} \mathrm{KL}((\mathcal{X}^i|\mathcal{X}^1, \ldots, \mathcal{X}^{i-1})||(\mathcal{Y}^i|\mathcal{Y}^1, \ldots, \mathcal{Y}^{i-1})).$$

We also review Pinsker's inequality and its reverse.

Proposition 4.1 (Pinsker's inequality and its reverse; see [SV15])**.** *Let P and Q be probability distributions, and assume $\alpha := \min_x Q(x) > 0$. Then,*

$$\Delta_{\mathrm{tv}}(P, Q)^2 \leq \frac{\ln 2}{2} \mathrm{KL}(P \parallel Q) \leq \frac{1}{\alpha} \Delta_{\mathrm{tv}}(P, Q)^2.$$

Average-Case Complexity. Let $\mathcal{U} = \{\mathcal{U}_n\}_{n \in \mathbb{N}}$ denote the family of uniform distributions, where $\mathcal{U}_n$ is the uniform distribution over $\{0,1\}^n$.

A family of distributions $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$ over strings is said to be samplable if there exists a polynomial-time randomized algorithm D such that, for each $n \in \mathbb{N}$, the distribution of $D(1^n)$ is statistically identical to $\mathcal{D}_n$. Trivially, $\mathcal{U}$ is samplable.

We say that a randomized algorithm A solves a promise problem Π on errorless average over $\mathcal{D}$ with failure probability $\delta \in (0,1)$ if (1) A outputs $\Pi(x)$ or $\perp$ (which represents "failure") with probability at least $3/4$ over the choice of randomness for A for every $x \in \mathsf{Support}(\mathcal{D})$, and (2) the failure probability that $A(x)$ outputs $\perp$ overwhelmingly (i.e., with probability at least $3/4$) over the choice of $x \sim \mathcal{D}$ is bounded above by δ.

We say that a distributional problem $(\Pi, \{\mathcal{D}_n\}_{n \in \mathbb{N}})$ has an errorless heuristic scheme A if, for all $n, \delta^{-1} \in \mathbb{N}$, the randomized algorithm A given n and δ^{-1} in unary solves Π on errorless average over $\mathcal{D}_n$ with failure probability δ.

Let $\mathsf{Avg}_{\delta}\mathsf{BPP}$ and AvgBPP denote the classes of distributional problems that admit, respectively, an errorless heuristic algorithm with failure probability $\delta(n)$, and an errorless heuristic scheme.

Let DistNP denote the class of distributional problems $(L, \{\mathcal{D}_n\}_{n \in \mathbb{N}})$ such that $L \in \mathsf{NP}$, and $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$ is samplable.

Auxiliary-Input One-Way Functions. We introduce auxiliary-input one-way functions, a notion first proposed by Ostrovsky and Wigderson [OW93]. Informally, these are families of functions that are hard to invert in a weaker sense often encountered in cryptographic contexts.

An auxiliary-input function is a family of functions $f = \{f_z\}_{z\in\{0,1\}^*}$ indexed by binary strings z. We say that f is polynomial-time computable if each $f_z(x)$ is polynomial-time computable from (z, x).

Definition 4.2 (Auxiliary-Input One-Way Function)**.** *A polynomial-time computable auxiliary-input function $f = \{f_z\colon \{0,1\}^{\mathsf{poly}(|z|)} \to \{0,1\}^{\mathsf{poly}(|z|)}\}_{z\in\{0,1\}^*}$ is said to be an auxiliary-input one-way function if for every polynomial-time randomized algorithm A, there exist infinitely many $z \in \{0,1\}^*$ such that*

$$\Pr_{r,A}\left[f_z(A(z, f_z(r))) = f_z(r)\right] < \mathsf{negl}(|z|),$$

where $r \sim \{0,1\}^{\mathsf{poly}(|z|)}$ is a random seed.

4.1 Algorithmic Information and Meta-Complexity

We fix an arbitrary efficient universal Turing machine U, which we assume has the following structure: three read-only one-way tapes (for input, auxiliary input, and external randomness, respectively), one read/write working tape, and one write-only one-way output tape. We assume that U cannot write blank symbols to the output tape. For each input $x \in \{0,1\}^*$, auxiliary input $z \in \{0,1\}^*$, and randomness $r \in \{0,1\}^*$, we write $U^{t,z,r}(x)$ to denote the contents of the output tape after executing U on input x, auxiliary input z, and external randomness r, for t steps. When the time bound t is not considered, and the auxiliary input z and randomness r are the empty string, we omit the corresponding superscripts.

For simplicity, we may identify a Turing machine Π with its binary encoding used as input to the universal Turing machine U. We use the notation $|\Pi|$ to represent the length of the binary encoding of Π. Furthermore, we say that an s-size randomized Turing machine (or program) $\Pi \in \{0,1\}^{\le s}$ generates a stream $x_1, \ldots, x_m, \ldots \in \{0,1\}^*$ in time t to mean that $U^{t',z,r}(\Pi)$ outputs $x_1, \ldots, x_m$ as a prefix for $t' \ge t$ and $r \sim \{0,1\}^{t'}$ (recall that the output tape of U is one-way).

Kolmogorov Complexity. For each $t \in \mathbb{N}$, the *t-time-bounded Kolmogorov complexity* $\mathrm{K}^t(x \mid z)$ *of a string $x \in \{0,1\}^*$ given advice $z \in \{0,1\}^*$* is defined as the minimum $k \in \mathbb{N}$ such that there exists $\Pi \in \{0,1\}^k$ for which $U^z(\Pi)$ halts within t steps and outputs x. We also define the time-unbounded Kolmogorov complexity as $\mathrm{K}(x \mid z) = \lim_{t\to\infty} \mathrm{K}^t(x \mid z)$. If z is the empty string, we omit the notation "$\mid z$".

In the literature of Kolmogorov complexity [LV19], z is usually referred to as a "conditional string" because of its connection to the conditional algorithmic probability in a recursion-theoretic regime. In a complexity-theoretic regime, however, the distinction between a conditional string and a conditional algorithmic probability will be crucial. We thus refer to z as an *advice string* to avoid the confusion with the notion of conditioning in probability.

It is well known that K admits an optimal form of conditional coding as follows.

Proposition 4.2. *There exist a polynomial p and a constant c such that for all $t \in \mathbb{N}$, every t-time sampler Π for a distribution $\mathcal{D}$ over strings, and all $x \circ y$ in the support of $\mathcal{D}$,*

$$\mathrm{K}(y \mid x, \Pi) \le -\log \mathcal{D}(y\|x) + c \cdot \log t,$$

where $\mathcal{D}(y\|x)$ denotes the conditional probability that y is sampled after observing the prefix x.

Proof. See [LV19, Corollary 4.3.2]. □

Approximating Time-Bounded Kolmogorov Complexity. We define the GapK$^{\mathrm{t}}$-vs-K problem used in our assumption.

Definition 4.3 (GapK$^{\mathrm{t}}$-vs-K)**.** *For $c \geq 0$, the promise problem* $\mathrm{Gap}_c\mathrm{K}^{\mathrm{t}}\text{-vs-K} = (\Pi_{\mathrm{yes}}, \Pi_{\mathrm{no}})$ *is defined as follows:*

$$\Pi_{\mathrm{yes}} := \left\{(x, 1^s, 1^t) : \mathrm{K}^t(x) \leq s\right\}$$
$$\Pi_{\mathrm{no}} := \left\{(x, 1^s, 1^t) : \mathrm{K}(x) > s + c\log(t|x|)\right\}.$$

We write GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP to denote that $\mathrm{Gap}_c\mathrm{K}^{\mathrm{t}}\text{-vs-K} \in$ pr-BPP for some constant $c \geq 0$.

It is known that the assumption GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP follows from the average-case easiness of NP and implies the nonexistence of auxiliary-input one-way functions:

Theorem 4.1 ([Hir20b; GKLO22])**.** *If* DistNP $\subseteq$ AvgBPP*, then* GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP*.*

Lemma 4.2 ([cf. HS17])**.** *If* GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then there is no auxiliary-input one-way function.*

Lemma 4.2 shows one use of the assumption GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP. The other use is to prove a chain rule for q^t, shown in Section 5 (Lemmas 5.1 and 5.2). In fact, the assumption in Lemma 4.2 can be replaced by this chain rule; see Appendix C. Therefore, in our technique, we indeed invoke the assumption GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP only to obtain the chain rule via an algorithmic construction, and once established, the remaining inference relies only on that chain rule.

The following proposition is straightforward.

Proposition 4.3. *If* GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then there exist a randomized polynomial time algorithm $\tilde{K}$ and a constant $c \geq 0$ such that for every $x \in \{0,1\}^*$ and every $t \geq O(|x|)$,*

$$\Pr_{\tilde{K}}\left[\mathrm{K}(x) \leq \tilde{K}(x, 1^t) \leq \mathrm{K}^t(x) + c\log(t|x|)\right] \geq 2/3.$$

Proof. Let A be a randomized polynomial-time algorithm that solves $\mathrm{Gap}_c\mathrm{K}^{\mathrm{t}}\text{-vs-K} \in$ pr-BPP for some constant $c \geq 0$. By a standard repetition argument, we may assume that the error probability of A on input $(x, 1^s, 1^t)$ is at most $1/(6|x|)$.

Define the algorithm $\tilde{K}$ as follows: given $x \in \{0,1\}^*$ and $t \in \mathbb{N}$, it executes $A(x, 1^s, 1^t)$ for all $s \in [2|x|]$ to find the smallest s^* such that $A(x, 1^{s^*}, 1^t) = 1$. Then, it returns $s^* + c\log(t|x|)$.

By the union bound, with probability at least $2/3$, A returns the correct answer for all $s \in [2|x|]$. In that case, we have:

$$\mathrm{K}(x) \leq s^* + c\log(t|x|) \leq \mathrm{K}^t(x) + c\log(t|x|),$$

where the first inequality follows from $A(x, 1^{s^*}, 1^t) = 1$, and the second follows from $A(x, 1^{s^*-1}, 1^t) = 0$. □

Universal Distribution and (q-)Computational Depth [cf. IL90; HN23]. For each $t \in \mathbb{N}$, the *t-time-bounded universal distribution* $\mathrm{Q}^{t,z}$ *given advice* $z \in \{0,1\}^*$ is defined as the distribution of $U^{t,z}(w)$ for a uniformly random w chosen from $\{0,1\}^t$. If z is the empty string, we write $\mathrm{Q}^{t,z}$ as Q^t by omitting z. We define $\mathrm{q}^t(x \mid z) := -\log \mathrm{Q}^{t,z}(x)$, where recall that $\mathrm{Q}^{t,z}(x)$ denotes the probability that x is sampled according to $\mathrm{Q}^{t,z}$. We also define the *t-time-bounded* (q-)*computational depth of x given advice z* as $\mathrm{cd}^t(x \mid z) := \mathrm{q}^t(x \mid z) - \mathrm{K}(x \mid z)$.

Here we introduce some basic properties.

Proposition 4.4 (Domination Property [cf. HN23, Lemma 6.9])**.** *There exist a polynomial*[6] p *and a constant* c *such that for all* $t \in \mathbb{N}$*, every* t*-time sampler* Π *for a distribution* $\mathcal{D}$ *over strings, and all* x *in the support of* $\mathcal{D}$*,*

$$\mathrm{Q}^{p(t)}(x) \geq 2^{-c\cdot|\Pi|} \cdot \mathcal{D}(x).$$

Proposition 4.5. *There exists a constant* c *such that for all* $x, z \in \{0,1\}^*$ *and all* $t \geq c \cdot |x|$*, we have* $\mathrm{Q}^{t,z}(x) > 0$*.*

Proof. The claim follows from the fact that there exists a trivial t-time program that hardcodes x and outputs it directly, without using z at all. □

Proposition 4.6. *There exist a polynomial* τ *and a constant* c *such that for all* $t, t' \in \mathbb{N}$ *with* $t' \geq \tau(t)$ *and all* $x, z \in \{0,1\}^*$ *with* $t \geq c \cdot |x|$*,* $\mathrm{q}^{t'}(x \mid z) \leq \mathrm{q}^{t}(x \mid z) + c \log t$*. In particular,* $\mathrm{cd}^{t'}(x \mid z) \leq \mathrm{cd}^{t}(x \mid z) + c \log t$*.*

Proof. Let τ be a sufficiently large polynomial (with respect to the simulation overhead of U). For every program Π (given z) whose computation on U outputs x precisely at the t-th step, we can construct a program Π' (again given z) that that simulates Π for exactly t steps and then outputs x and halts. This simulation halts within $\tau(t)$ steps, so for all $t' \geq \tau(t)$ the computation of Π' is t'-time-bounded. Moreover, Π' can be described using $|\Pi| + O(\log(t|\Pi|)) \leq |\Pi| + O(\log t)$ bits, where we have assumed $|\Pi| \leq t$ since otherwise U would not be able to read the entire description of Π within t steps.

Thus, for every $t' \geq \tau(t)$

$$\mathrm{Q}^{t',z}(x) \geq 2^{-O(\log t)} \cdot \mathrm{Q}^{t,z}(x).$$

Taking negative logarithms yields the proposition. The claim about cd is immediately follows by subtracting $\mathrm{K}(x \mid z)$ from both sides. □

Proposition 4.7. *There exists a constant* c *such that for all* $x, z \in \{0,1\}^*$ *and all* $t \in \mathbb{N}$ *with* $t \geq c \cdot |x|$*,*

$$\mathrm{K}(x \mid z) \leq \mathrm{q}^{t}(x \mid z) + c \log t.$$

Proof. The proposition follows from Proposition 4.2, since the sampler for $\mathrm{Q}^{t,z}$ can be specified using only $O(\log t)$ bits given z. □

Probabilistic Kolmogorov Complexity [GKLO22]. For each $t \in \mathbb{N}$ and $\delta \in [0,1]$, we define the t-time-bounded probabilistic Kolmogorov complexity $\mathsf{pK}^t_\delta(x \mid z)$ of a string $x \in \{0,1\}^*$ given advice $z \in \{0,1\}^*$ as the minimum $k \in \mathbb{N}$ such that

$$\Pr_{r\sim\{0,1\}^t}\left[\exists \Pi \in \{0,1\}^{\leq k} \text{ s.t. } U^{t,z,r}(\Pi) \text{ halts within } t \text{ steps and outputs } x\right] \geq \delta.$$

By default, we set $\delta = 2/3$ and omit the subscript δ unless otherwise specified.

By definition, we obtain the following proposition:

Proposition 4.8. *For each* $t \in \mathbb{N}$ *and* $x \in \{0,1\}^*$*,*

$$\Pr_{r\sim\{0,1\}^t}\left[\mathrm{K}^t(x \mid r) \leq \mathsf{pK}^t(x)\right] \geq 2/3.$$

We also state a known relationship between pK^t and K, as shown in the following lemma.

[6] The polynomial overhead p in time arises from the simulation overhead of the universal Turing machine U.

Lemma 4.3 ([GKLO22, Lemma 18]). *For any $t, n \in \mathbb{N}$ with $t \geq O(n)$ and any $x \in \{0,1\}^{\leq n}$, $\mathrm{K}(x \mid t) \leq \mathsf{pK}^t(x) + \log t$.*

We observe that appending randomness to the advice does not significantly affect pK in expectation.

Proposition 4.9. *There exists a polynomial p such that for each $t, n \in \mathbb{N}$ with $t \geq O(n)$ and $x, y \in \{0,1\}^{\leq n}$,*

$$\mathsf{pK}^{p(t)}(x \mid y) \leq \mathop{\mathbb{E}}_{r \sim \{0,1\}^t} \left[\mathsf{pK}^t(x \mid y, r)\right] + O(\log t).$$

Proof. Let $v = \mathbb{E}_{r \sim \{0,1\}^t}[\mathsf{pK}^t(x \mid y, r)]$. Since $t \geq O(n)$, we may assume that $v \leq 2n$.

We first observe that

$$\Pr_r \left[\mathsf{pK}^t(x \mid y, r) \leq v + 4\right] \geq \frac{1}{2n}. \tag{2}$$

Indeed, if this inequality were false, then

$$v = \mathop{\mathbb{E}}_{r \sim \{0,1\}^t} \left[\mathsf{pK}^t(x \mid y, r)\right] > (v+4)\left(1 - \frac{1}{2n}\right) = v + 4 - \frac{2}{n} - \frac{v}{2n} \geq v + 1,$$

which is a contradiction.

Equation (2) implies that $\mathsf{pK}^{\mathsf{poly}(t)}_{1/3n}(x \mid y) \leq v + O(1)$. To see this, consider the first $2t$-bit of an external random string $r \sim \{0,1\}^{\mathsf{poly}(t)}$ is composed of two random strings $r_1, r_2 \sim \{0,1\}^t$. The event in Equation (2) holds for r_1 with probability at least $1/(2n)$; conditioned on this, by the definition of $\mathsf{pK}^t(x \mid y, r_1)$, there exists a program of length at most $v + 4$ that outputs x with probability at least $2/3$ over r_2. We can simulate such a program in $\mathsf{poly}(t)$ time by interpreting each bit of r_1 and r_2 as being read from two separate portions of r.

Goldberg, Kabanets, Lu, and Oliveira [GKLO22, Lemma 21] proved that the success probability of pK is easily amplified by standard repetition. Thus,

$$\mathsf{pK}^{O(n \cdot \mathsf{poly}(t))}(x \mid y) \leq \mathsf{pK}^{\mathsf{poly}(t)}_{1/3n}(x \mid y) + O(\log n) \leq v + O(\log t),$$

as desired. □

One main advantage of working with pK lies in the following coding theorem.

Theorem 4.2 (Optimal Coding for pK [LOZ22]). *There exists a polynomial p such that for every randomized Turing machine M that may take advice $z \in \{0,1\}^*$ and halts in time $t_M(z)$ and for every string $x \in \{0,1\}^*$ is the support of $M(z)$,*

$$\mathsf{pK}^{p(t_M(z))}(x \mid M, z) \leq -\log \Pr_M[x \leftarrow M(z)] + \log p(t_M(z)).$$

In particular,

$$\mathsf{pK}^{p(t_M(z))}(x \mid z) \leq O(|M|) - \log \Pr_M[x \leftarrow M(z)] + O(\log t_M(z)).$$

We now review the relationship between pK and q. These two complexity measures are essentially equivalent, up to an additive logarithmic term and a polynomial overhead in the time bound.

By applying Theorem 4.2 to the universal distribution Q^t, we immediately obtain the following upper bound:

Lemma 4.4. *There exists a polynomial p such that for all $x, z \in \{0,1\}^*$ and all $t \geq O(|x|)$,*

$$\mathsf{pK}^{p(t)}(x \mid z) \leq \mathrm{q}^t(x \mid z) + \log p(t).$$

A corresponding lower bound is also known, following from the domination property of the universal distribution:

Proposition 4.10 ([HN23, Proposition B.3]). *There exists a constant c such that for each $t \in \mathbb{N}$ and $x, z \in \{0,1\}^*$,*

$$\mathrm{q}^{ct}(x \mid z) \leq \mathsf{pK}^t(x \mid z) + c \log t.$$

Direct Product Generator We review the notion of the direct product generator, explicitly formulated in [Hir21].

Definition 4.4 (Direct Product Generator). *For $k\colon \mathbb{N} \to \mathbb{N} \cup \{0\}$ with $k(n) \leq 2n$, a k-direct product generator DP_k takes $x \in \{0,1\}^*$ and $z \in \{0,1\}^{2|x|^2}$ as input and outputs $z \circ \langle x, z_1 \rangle_{\mathbb{F}_2} \circ \cdots \circ \langle x, z_{k(|x|)} \rangle_{\mathbb{F}_2}$, where $\langle , \rangle_{\mathbb{F}_2}$ denotes the inner product in $\mathbb{F}_2$, and $z_i = z_{[(i-1)|x|+1:i|x|]}$ for each $i \in [k(|x|)]$.*

Note that, in the literature, the seed length is often defined as $k \cdot |x|$ for a parameter k. Here, we fix the seed length as $2|x| \cdot |x|$ independently of k by ensuring the seed is sufficiently long.

The following lemma captures the key property of the direct product generator. Intuitively, it transforms a string of high pK into a pseudorandom string against algorithms of bounded description size. (The lemma is stated in the contrapositive form.)

Lemma 4.5 (DP-reconstruction for pK [Hir20b]; see also [GKLO22, Lemma 22]). *There exists a polynomial p_{DP} such that for any $\epsilon \geq 0$, $n, k \in \mathbb{N}$, and $x \in \{0,1\}^n$, if D is a t_D-time randomized Turing machine that ϵ-distinguishes $\mathsf{DP}_k(x; z)$ from random, i.e.,*

$$\Pr_{z \sim \{0,1\}^{nk}, D}[D(\mathsf{DP}_k(x; z)) = 1] - \Pr_{w \sim \{0,1\}^{2n^2+k}, D}[D(w) = 1] > \epsilon,$$

then

$$\mathsf{pK}^{p_{\mathsf{DP}}(t_D, n, \epsilon^{-1})}(x \mid D) \leq k + \log p_{\mathsf{DP}}(t_D, n, \epsilon^{-1}).$$

Slow Growth Law. We review the slow growth law [Ben88; AFPS12; Hir23], which informally states that an efficient randomized algorithm cannot significantly increase the computational depth of the given input.

Specifically, we will use the following form, proved in [HN23, Lemma 6.15]. The proof in that work relativizes since it builds on a relativizing result from [Hir23, Lemma 8.13], so the result also holds in the presence of an advice string.

Lemma 4.6 ([HN23, Lemma 6.15]). *There exists a polynomial p such that for every $z \in \{0,1\}^*$, every t_Π-time randomized Turing machine Π that takes z as advice, every $i, t \in \mathbb{N}$ with $t \geq p(t_\Pi + |\Pi|)$, and every $\delta \in (0,1]$,*

$$\Pr_{x \sim \Pi(z)}\left[\mathrm{cd}^{p(t)}(x_{[i]} \mid z) \leq \mathrm{cd}^t(\Pi \mid z) + \log p(t\delta^{-1})\right] \geq 1 - \delta.$$

We state two corollaries that follow as special cases of Lemma 4.6.

Lemma 4.7. *There exist a polynomial p and constant $c > 0$ such that for every $x, z \in \{0,1\}^*$ and every $i, t \in \mathbb{N}$ with $t \geq p(|x|)$,*

$$\mathrm{cd}^{p(t)}(x_{[i]} \mid z) \leq \mathrm{cd}^t(x \mid z) + \log t + c.$$

Proof. This follows by applying Lemma 4.6 to a program that simply embeds x and outputs it (without using randomness). □

Proposition 4.11. *There exist a polynomial τ and a constant c such that for every $t \in \mathbb{N}$, every t-time sampler Π for a distribution $\mathcal{D}$ over strings (with $t \geq |\Pi|$), and every $\delta \in (0,1]$, we have*

$$\Pr_{x\sim\mathcal{D}}\left[\mathrm{cd}^{\tau(t)}(x) \leq c \cdot \left(|\Pi| + \log \delta^{-1} + \log t\right)\right] \geq 1 - \delta.$$

Proof. This follows since $\mathrm{cd}^{O(t)}(\Pi) \leq \mathrm{q}^{O(t)}(\Pi) \leq |\Pi| + O(1)$. □

4.2 Universal Extrapolation

For each distribution $\mathcal{D}$ over $\{0,1\}^*$, each $k \in \mathbb{N}$, and each $x \in \{0,1\}^*$, we define $\mathsf{Next}_k(x;\mathcal{D})$ as the conditional distribution over the k-bit prefix of a continuation of x, sampled according to $\mathcal{D}$. (If x is not in the support of the prefixs of $\mathcal{D}$, we treat it as a distribution over the empty string.) More formally, $\mathsf{Next}_k(x;\mathcal{D})$ is a distribution over $\{0,1\}^{\leq k}$. The probability that $y \in \{0,1\}^{\leq k}$ is sampled according to $\mathsf{Next}_k(x;\mathcal{D})$ is defined as the probability, over w sampled from $\mathcal{D}$, that the first $|x|+k$ bits of w (or w itself if $|w| < |x|+k$) are equal to xy, conditioned that x is a prefix of w.

Following the terminology of [HN23], we refer to the task of sampling from $\mathsf{Next}_k(x;\mathrm{Q}^t)$ on input a "context" x as *universal extrapolation*. We further consider an *advised* variant, in which the goal is to sample from $\mathsf{Next}_k(x;\mathrm{Q}^{t,z})$ on input the context x together with advice z.

Building on [HN23] via distributional inversion for $\mathrm{Q}^{t,z}$, and assuming no auxiliary-input one-way functions exist, we can perform advised universal extrapolation in time polynomial in the parameters and exponential only in the computational depth of the context. The formal statement is given below.

Lemma 4.8 (Universal Extrapolation Lemma)**.** *Assuming no auxiliary-input one-way functions exist, there is a polynomial-time randomized algorithm UE such that for all $k, t, \epsilon^{-1}, \alpha \in \mathbb{N}$ and all $z, x \in \{0,1\}^*$ with $\mathrm{cd}^t(x \mid z) \leq \alpha$,*

$$\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(x; z, 1^{\langle k,t,\epsilon^{-1},2^\alpha\rangle}\right),\ \mathsf{Next}_k\left(x;\mathrm{Q}^{t,z}\right)\right) \leq \epsilon.$$

Proof. The argument of [HN23, Theorem 8.1] applies verbatim to the advised setting: the construction and analysis are unchanged when z is provided as auxiliary input. See [HN23, Section 8] for details. □

Furthermore, the main result of [HN23] yields an inference algorithm whose running time is *exponential* in the description length s of a sampler for the target.

Proposition 4.12 (Corollary of [HN23, Theorem 9.1])**.** *If auxiliary-input one-way functions do not exist (indeed, it already suffices to assume the nonexistence of infinitely-often one-way functions), then the complexity-theoretic universal inductive inference is solvable in time $2^{O(s)} \cdot \mathsf{poly}(t, \epsilon^{-1}, \delta^{-1})$ with round complexity $O(s\,\epsilon^{-2}\delta^{-1})$.*

5 A Chain Rule for Time-Bounded Algorithmic Information

In this section we give an algorithmic proof of the chain rule for q^t; the formal statement is as follows.

Lemma 5.1. *If $\mathrm{GapK}^t\text{-vs-K} \in \mathsf{pr\text{-}BPP}$, then there exist a randomized polynomial-time algorithm CR, a constant $c > 0$, and a polynomial τ such that for every m-tuple $\boldsymbol{x} = (x_1, \ldots, x_m)$ of binary strings and every $t \geq m + n$:*

1. *For each* $i \in [m]$, $\mathsf{CR}(\boldsymbol{x}_{\le i}, 1^t)$ *outputs an integer* $k_i \in \mathbb{N}$.

2. *With probability at least* $1 - \mathsf{negl}(t)$ *over the internal randomness used to produce* $k_1, \dots, k_m$, *the following hold simultaneously:*

$$k_i \ge \mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) \quad \textit{for all } i \in [m],$$

and

$$\sum_{i=1}^{m} k_i \le \mathrm{q}^t(\boldsymbol{x}) + m\, c \log t.$$

Moreover, for every $\delta \in (0, 1]$,

$$\Pr_{i \sim [m]}\left[k_i \le \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + \delta^{-1} \cdot \left(\frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + c \log t\right)\right] \ge 1 - \delta.$$

Proof. Without loss of generality, we may assume that $t \ge O(|\boldsymbol{x}|)$. This is because, once the statement is established, for any t satisfying $m + n \le t \le O(|\boldsymbol{x}|)$, we can first apply it to $t' = \max\{\tau(t), O(|\boldsymbol{x}|)\}$ $(\le \mathsf{poly}(t))$ for the polynomial τ in Proposition 4.6, and then derive the statement for t from Proposition 4.6, namely,

$$\mathrm{q}^{t'}(\boldsymbol{x}) \le \mathrm{q}^t(\boldsymbol{x}) + O(\log t).$$

Since $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$, there exist a randomized polynomial-time algorithm $\tilde{K}$ and a constant c_0 such that for each $x \in \{0,1\}^*$ and $t \in \mathbb{N}$,

$$\Pr_{\tilde{K}}\left[\mathrm{K}(x) < \tilde{K}(x, 1^t) \le \mathrm{K}^t(x) + c_0 \log(t|x|)\right] \ge 2/3.$$

Consider an arbitrary $m \in \mathbb{N}$ and an m-tuple $\boldsymbol{x} = (x_1, \dots, x_m)$ of binary strings. Consider any $t \ge O(|\boldsymbol{x}|)$, as stated in the lemma. For each i, let $n_i = |x_i|$, and set $n = \max_i n_i$, $N = \sum_{i=1}^m 2n_i^2$, and $M = \sum_{i=1}^m 2n_i$. Then $t \ge m$ and $t \ge \sum_i n_i \ge n$. Moreover, $2t^2 \ge N + M$.

Let p_0 be the polynomial in Lemma 4.4. Let p_1 be a large enough polynomial we specify later. We consider a randomized algorithm D that is given $y \in \{0,1\}^{2t^2}$, selects $r \sim \{0,1\}^{p_0(t)}$, and outputs 1 if

$$\tilde{K}\left((r, y), 1^{p_1(t)}\right) \le |r| + |y| - 7 \,(= |r| + 2t^2 - 7);$$

outputs 0 otherwise.

Let $k_1, \dots, k_m \in \mathbb{N} \cup \{0\}$ be arbitrary parameters with $k_i \le 2n_i$ for each i. Observe that the length of $\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m)$, where $z_i \in \{0,1\}^{2n_i^2}$ in our formulation, is

$$\sum_{i=1}^{m}(2n_i^2 + k_i) = N + \sum_{i=1}^{m} k_i \le N + M \le 2t^2.$$

First, observe that D distinguishes $\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m) \circ y'$ from a truly random string if

$$\sum_{i=1}^{m} k_i \;\ge\; \mathrm{q}^t(x_1, \dots, x_m) + c_2 \cdot m \log t,$$

where $y' \sim \{0,1\}^{2t^2 - N - \sum_i k_i}$ and c_2 is a sufficiently large constant to be specified later.

Notice that we can efficiently compute $(r, \mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m) \circ y')$ from $r, x_1, \ldots, x_m$, $z_1, \ldots, z_m$, $k_1, \ldots, k_m$, and y'. Now, we choose p_1 and c_1 enough large so that for each $n, m, t \in \mathbb{N}$, $x_1, \ldots, x_m \in \{0,1\}^{\leq n}$, and for each $r, z_1, \ldots, z_m, y'$ (where $r \in \{0,1\}^{p_0(t)}$ and $z_i \in \{0,1\}^{2n_i^2}$ for each i, and $y' \in \{0,1\}^{2t^2 - N - \sum_i k_i}$),

$$\begin{aligned}
&\mathrm{K}^{p_1(t)}(r, \mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m) \circ y') \\
&\leq \mathrm{K}^{p_0(t)}(x_1, \ldots, x_m \mid r) + |r| + |z_1| + \cdots + |z_m| + |y'| + O(\log k_1 + \ldots + \log k_m) + O(\log t) \\
&\leq \mathrm{K}^{p_0(t)}(x_1, \ldots, x_m \mid r) + |r| + N + |y'| + c_1 \cdot m \log t.
\end{aligned}$$

For each i, suppose that $y_i = \mathsf{DP}_{k_i}(x_i; z_i)$. Recall that, with probability at least $2/3$, we have

$$\tilde{K}\Big((r, y_1 \circ \cdots \circ y_m \circ y'), 1^{p_1(t)}\Big) \;\leq\; \mathrm{K}^{p_1(t)}(r, y_1 \circ \cdots \circ y_m \circ y') + c_0 \log\big(p_1(t) \cdot (p_0(t) + 2t^2)\big).$$

In addition, by Proposition 4.8, when $r \sim \{0,1\}^{p_0(t)}$, $\mathrm{K}^{p_0(t)}(x \mid r) \leq \mathsf{pK}^{p_0(t)}(x)$ with probability at least $2/3$.

If both events occur, then for a sufficiently large constant c_2, we obtain

$$\begin{aligned}
\tilde{K}\Big((r, y_1 \circ \cdots \circ y_m \circ y'), 1^{p_1(t)}\Big) &\leq \mathrm{K}^{p_1(t)}(r, y_1 \circ \cdots \circ y_m \circ y') + c_0 \log\big(p_1(t) \cdot (p_0(t) + 2t^2)\big) \\
&\leq \mathrm{K}^{p_0(t)}(x_1, \ldots, x_m \mid r) + |r| + N + |y'| + c_1 m \log t \\
&\quad + c_0 \log\big(p_1(t) \cdot (p_0(t) + 2t^2)\big) \\
&\leq \mathsf{pK}^{p_0(t)}(x_1, \ldots, x_m) + |r| + N + |y'| + c_1 m \log t \\
&\quad + c_0 \log\big(p_1(t) \cdot (p_0(t) + 2t^2)\big) \\
&\leq \mathrm{q}^t(x_1, \ldots, x_m) + |r| + N + |y'| + c_2 m \log t - 7,
\end{aligned}$$

where the last inequality follows from Lemma 4.4.

Moreover, if $\sum_{i=1}^m k_i \geq \mathrm{q}^t(x_1, \ldots, x_m) + c_2 \cdot m \log t$, then

$$\begin{aligned}
\tilde{K}\Big((r, y_1 \circ \cdots \circ y_m), 1^{p_1(t)}\Big) &\leq \mathrm{q}^t(x_1, \ldots, x_m) + |r| + N + |y'| + c_2 m \log t - 7 \\
&\leq |r| + N + |y'| + \sum_{i=1}^m k_i - 7 \\
&= |r| + 2t^2 - 7.
\end{aligned}$$

Thus, we have

$$\Pr_{D, z_1, \ldots, z_m, y'}\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m) \circ y'\right) = 1\right] \geq \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}.$$

In contrast, we consider the case in which $y_i = w_i \sim \{0,1\}^{2n_i^2 + k_i}$ for each i. By the standard counting argument, with probability at least $1 - 2^{-5}$ over $r, w_1, \ldots, w_m, y'$,

$$\mathrm{K}(r, w_1 \circ \cdots \circ w_m \circ y') \geq |r| + |w_1 \circ \cdots \circ w_m \circ y'| - 6.$$

Since $\tilde{K}((r, w_1 \circ \cdots \circ w_m \circ y'), 1^t) > \mathrm{K}(r, w_1 \circ \cdots \circ w_m \circ y')$ with probability at least $2/3$, the union bound implies that

$$\Pr_{D, w_1, \ldots, w_m, y'}\left[D\left(w_1 \circ \cdots \circ w_m \circ y'\right) = 1\right] \leq \frac{1}{2^5} + \frac{1}{3} < \frac{7}{18}.$$

Thus, whenever $\sum_{i=1}^{m} k_i \geq \mathrm{q}^t(x_1, \ldots, x_m) + c_2 \cdot m \log t$, we have

$$\Pr\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m) \circ y'\right) = 1\right] - \Pr\left[D\left(w_1 \circ \cdots \circ w_m \circ y'\right) = 1\right] \geq \frac{1}{18}. \quad (3)$$

Now, we consider a randomized procedure K that generates $k_1, \ldots, k_m$ in the following inductive manner: Let i be the current round and assume that $k_1, \ldots, k_{i-1}$ have been already determined. For each $j \in [2n_i]$, the procedure K empirically estimates two probabilities

$$dp_j^i = \Pr_{z_1,\ldots,z_i,r,D}\left[D(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1}; z_{i-1}) \circ \mathsf{DP}_j(x_i; z_i) \circ r) = 1\right],$$

and

$$tr_j^i = \Pr_{z_1,\ldots,z_{i-1},w_i,r,D}\left[D(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1}; z_{i-1}) \circ w_i \circ r) = 1\right],$$

within additive accuracy $\pm 1/(216t)$ and negligible (in t) confidence error, where $|w_i| = 2n_i^2 + j$, and r is chosen so that the total length of the input becomes $2t^2$. Let $\tilde{dp}_j^{\,i}$ and $\tilde{tr}_j^{\,i}$ be the estimated values, respectively. Then, K determines k_i as the maximum $j \in [2n_i]$ satisfying that $\tilde{dp}_j^{\,i} - \tilde{tr}_j^{\,i} \leq 1/(54t)$ (if there is no such j, let $k_i = 0$). For each $i \in [m]$, let ρ_i denote the randomness used by K in the i-th round. Then, $k_1, \ldots, k_i$ are *deterministically* computable in polynomial time (in t) from $x_1, \ldots, x_i$, t, the description of D, and $\rho_1, \ldots, \rho_i$, according to the procedure K.

Fix any $\rho_1, \ldots, \rho_m$ such that all empirical estimations are performed successfully. Notice that they determine each value of k_i. Then, it must hold, for each $i \in [m]$,

$$dp_{k_i}^i - tr_{k_i}^i \leq \tilde{dp}_{k_i}^{\,i} - \tilde{tr}_{k_i}^{\,i} + \frac{2}{216t} \leq \frac{1}{54t} + \frac{1}{108t} = \frac{1}{36t}$$

and

$$dp_{k_i+1}^i - tr_{k_i+1}^i \geq \tilde{dp}_{k_i+1}^{\,i} - \tilde{tr}_{k_i+1}^{\,i} - \frac{2}{216t} > \frac{1}{54t} - \frac{1}{108t} = \frac{1}{108t}.$$

For notational simplicity, let $dp^i = dp_{k_i}^i$ and $tr^i = tr_{k_i}^i$ for each $i \in [m]$, and let

$$dp^0 := \Pr_{D,w_1,\ldots,w_m,r}\left[D\left(w_1 \circ \cdots \circ w_m \circ r\right) = 1\right].$$

Then, we can observe that

$$\begin{aligned}\Pr\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m) \circ y'\right) = 1\right] - \Pr\left[D\left(w_1 \circ \cdots \circ w_m \circ y'\right) = 1\right] \\ = dp^m - dp^0 = \sum_{i=1}^{m}(dp^i - dp^{i-1}) = \sum_{i=1}^{m}(dp^i - tr^i) \leq m \cdot \frac{1}{36t} = \frac{1}{36}.\end{aligned}$$

From Equation (3), it follows that

$$\sum_{i=1}^{m} k_i < \mathrm{q}^t(x_1, \ldots, x_m) + c_2 \cdot m \log t. \quad (4)$$

We will show that for a large enough polynomial p_2, the following holds for each $i \in [m]$:

$$\mathsf{pK}^{p_2(t)}(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{\rho}_{<i}) \leq k_i + \log p_2(t). \quad (5)$$

Assuming the inequalities above at first, we now proceed to derive the lemma by specifying CR.

Consider now the randomness of $\rho_1, \ldots, \rho_m$. Since the empirical estimations in K are performed with negligible confidence error,

$$\Pr_{\rho_1,\ldots,\rho_m}\left[\forall i \in [m],\ \mathsf{pK}^{p_2(t)}(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{\rho}_{<i}) \le k_i + \log p_2(t)\right] \ge 1 - \mathsf{negl}(t).$$

Thus, for each $i \in [m]$,

$$\mathop{\mathbb{E}}_{\rho_1,\ldots,\rho_{i-1}}\left[\mathsf{pK}^{p_2(t)}(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{\rho}_{<i})\right] \le \mathop{\mathbb{E}}_{\rho_1,\ldots,\rho_i}[k_i] + \log p_2(t) + 2n \cdot \mathsf{negl}(t).$$

From Proposition 4.9, for large enough polynomials p_3 and τ, for each $i \in [m]$,

$$\begin{aligned}
\mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) &\le \mathsf{pK}^{p_3(t)}(x_i \mid \boldsymbol{x}_{<i}) + O(\log t) \\
&\le \mathop{\mathbb{E}}_{\rho_1,\ldots,\rho_{i-1}}\left[\mathsf{pK}^{p_2(t)}(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{\rho}_{<i})\right] + O(\log t) \\
&\le \mathop{\mathbb{E}}_{\rho_1,\ldots,\rho_i}[k_i] + \log p_3(t), \qquad (6)
\end{aligned}$$

where the first inequality follows from Proposition 4.10.

By the same reasoning, it follows from Equation (4) that

$$\sum_{i=1}^{m} \mathop{\mathbb{E}}_{\rho_1,\ldots,\rho_i}[k_i] < \mathrm{q}^t(x_1, \ldots, x_m) + c_2 \cdot m \log t + 2mt \cdot \mathsf{negl}(t). \qquad (7)$$

The algorithm CR, given $\boldsymbol{x}_{\le i} = (x_1, \ldots, x_i)$ and 1^t, empirically estimates the value of $\mathbb{E}_{\rho_1,\ldots,\rho_i}[k_i]$ by executing K independently with $(x_1, \ldots, x_i)$, with accuracy error ± 1 and negligible confidence error. ($\mathsf{poly}(t)$ trials suffice by Hoeffding's inequality and the fact that each $k_i \in [0, 2t]$.) Let $\tilde{k}_i$ be the estimated value, i.e., if the empirical estimation succeeds then $|\tilde{k}_i - \mathbb{E}_{\rho_1,\ldots,\rho_i}[k_i]| \le 1$. Then CR outputs $\tilde{k}_i + 1 + \log p_3(t)$, which serves as k_i in the lemma.

The claim about CR is straightforward to verify. If the empirical estimation in CR succeeds (which occurs with probability at least $1 - \mathsf{negl}(t)$), then for each $i \in [m]$, from Equation (6) we have

$$\mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) \le \mathop{\mathbb{E}}_{\rho_1,\ldots,\rho_i}[k_i] + \log p_3(t) \le \tilde{k}_i + 1 + \log p_3(t).$$

In addition, from Equation (7),

$$\begin{aligned}
\sum_{i=1}^{m}\left(\tilde{k}_i + 1 + \log p_3(t)\right) &\le \sum_{i=1}^{m} \mathbb{E}[k_i] + 2m + m \log p_3(t) \\
&< \mathrm{q}^t(x_1, \ldots, x_m) + 2m + m \log p_3(t) + c_2 \cdot m \log t + 2mt \cdot \mathsf{negl}(t) \\
&\le \mathrm{q}^t(\boldsymbol{x}) + m \cdot c_3 \log t,
\end{aligned}$$

for a sufficiently large constant c_3.

The statement in the 'Furthermore' part of the lemma can also be verified under the same event that all empirical estimations succeed. By rearranging the above, we obtain

$$\begin{aligned}
\sum_{i=1}^{m}\left(\tilde{k}_i + 1 + \log p_3(t)\right) &\le \mathrm{q}^t(\boldsymbol{x}) + m \cdot c_3 \log t \\
&= \mathrm{K}(\boldsymbol{x}) + \mathrm{cd}^t(\boldsymbol{x}) + m \cdot c_3 \log t \\
&\le \sum_{i=1}^{m} \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + \mathrm{cd}^t(\boldsymbol{x}) + m \cdot c_3 \log t + m \cdot O(\log t).
\end{aligned}$$

Thus, for a sufficiently large constant c_4,

$$\mathop{\mathbb{E}}_{i\sim[m]}\left[\tilde{k}_i + 1 + \log p_3(t) - \mathrm{K}(x_i \mid \boldsymbol{x}_{<i})\right] \le \frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + c_4 \log t.$$

Recall that for each $i \in [m]$,

$$\tilde{k}_i + 1 + \log p_3(t) - \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) \ge \mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) - \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) \ge -c_5 \log t,$$

for a sufficiently large constant c_5 from Proposition 4.7. Trivially,

$$\mathop{\mathbb{E}}_{i\sim[m]}\left[\tilde{k}_i + 1 + \log p_3(t) - \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + c_5 \log t\right] \le \frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + (c_4 + c_5) \log t.$$

Since the quantity inside the expectation is nonnegative, an application of Markov's inequality yields that for every $\delta \in (0, 1]$,

$$\Pr_{i\sim[m]}\left[\tilde{k}_i + 1 + \log p_3(t) - \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + c_5 \log t \le \delta^{-1} \cdot \left(\frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + (c_4 + c_5) \log t\right)\right] \ge 1 - \delta.$$

Therefore, for any such $i \in [m]$,

$$\tilde{k}_i + 1 + \log p_3(t) \le \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + \delta^{-1} \cdot \left(\frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + (c_4 + c_5) \log t\right).$$

By choosing $c = \max\{c_3,\, c_4 + c_5\}$ in the lemma, we obtain the desired bound.

We complete the proof by providing the deferred proof of Equation (5). Recall that for each $i \in [m]$,

$$\Pr_{D,z_1,\ldots,z_i,r}\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1}; z_{i-1}) \circ \mathsf{DP}_{k_i+1}(x_i; z_i) \circ r\right) = 1\right]$$
$$- \Pr_{D,z_1,\ldots,z_{i-1},w_i,r}\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1}; z_{i-1}) \circ w_i \circ r\right) = 1\right] > \frac{1}{108t}. \quad (8)$$

Based on the above, we construct the following algorithm D_i that distinguishes $\mathsf{DP}_{k_i+1}(x_i; z_i)$ from truly random strings *given* D, $\boldsymbol{x}_{<i} = (x_1, \ldots, x_{i-1})$, and $\boldsymbol{\rho}_{<i} = (\rho_1, \ldots, \rho_{i-1})$: On input $y \in \{0,1\}^{2n_i^2 + k_i + 1}$, the distinguisher D_i first executes K with $\boldsymbol{x}_{<i}$ and randomness $\boldsymbol{\rho}_{<i}$ to obtain $k_1, \ldots, k_{i-1}$, and then outputs the same answer to

$$D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1}; z_{i-1}) \circ y \circ r\right),$$

where r is selected so that the total length of the input becomes $2t^2$.

Then, Equation (8) is rewritten as

$$\Pr_{D_i,z_i}\left[D_i\left(\mathsf{DP}_{k_i+1}(x_i; z_i)\right) = 1\right] - \Pr_{D_i,w_i}\left[D_i\left(w_i\right) = 1\right] \ge \frac{1}{108t}.$$

From Lemma 4.5,

$$\mathsf{pK}^{p_4(t)}(x_i \mid D_i) \le k_i + 1 + \log p_4(t),$$

for a large enough polynomial p_4. Thus, by taking p_2 large enough,

$$\begin{aligned}\mathsf{pK}^{p_2(t)}(x_i \mid \boldsymbol{x}_{<i}, \boldsymbol{\rho}_{<i}) &\le \mathsf{pK}^{p_4(t)}(x_i \mid D_i) + |D| + O(\log t)\\ &\le k_i + 1 + \log p_4(t) + |D| + O(\log t)\\ &\le k_i + \log p_2(t),\end{aligned}$$

as desired.

□

As an immediate corollary we obtain the following chain rule for q^t, which formally restates Lemma 3.1.

Lemma 5.2 (Chain rule for q^t)**.** *Assume* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*. Then there exist an absolute constant $c > 0$ and a polynomial τ such that for every m-tuple $\boldsymbol{x} = (x_1, \ldots, x_m)$ of binary strings of length at most n and every $t \geq m + n$,*

$$\sum_{i=1}^{m} \mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) \leq \mathrm{q}^t(\boldsymbol{x}) + m \cdot c \log t.$$

Proof. By Lemma 5.1 there is a randomized procedure CR such that, for some fixing of its random coins, it outputs integers $k_1, \ldots, k_m$ with the following properties: (i) $k_i \geq \mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i})$ for all $i \in [m]$, and (ii) $\sum_{i=1}^{m} k_i \leq \mathrm{q}^t(\boldsymbol{x}) + m \cdot c \log t$. Therefore,

$$\sum_{i=1}^{m} \mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) \leq \sum_{i=1}^{m} k_i \leq \mathrm{q}^t(\boldsymbol{x}) + m \cdot c \log t,$$

as claimed. □

As a further consequence we obtain a tail bound for the conditional computational depth.

Lemma 5.3 (Tail bound for conditional computational depth)**.** *Assume* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*. There exist an absolute constant $c > 0$ and a polynomial τ such that for every m-tuple $\boldsymbol{x} = (x_1, \ldots, x_m)$ of binary strings of length at most n, every $t \geq m + n$, and every $\delta \in (0, 1]$,*

$$\Pr_{i \sim [m]}\left[\mathrm{cd}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) \leq \delta^{-1} \cdot \left(\frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + c \log t\right)\right] \geq 1 - \delta.$$

Proof. Fix the random coins of CR as in the proof of Lemma 5.2, and let $k_1, \ldots, k_m$ be the resulting outputs. For these quantities we have

$$\begin{aligned}\Pr_{i \sim [m]}\left[\mathrm{cd}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i}) \leq \delta^{-1} \cdot \left(\frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + c \log t\right)\right] \\ \geq \Pr_{i \sim [m]}\left[k_i - \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) \leq \delta^{-1} \cdot \left(\frac{\mathrm{cd}^t(\boldsymbol{x})}{m} + c \log t\right)\right] \geq 1 - \delta,\end{aligned}$$

where the first inequality holds since $k_i \geq \mathrm{q}^{\tau(t)}(x_i \mid \boldsymbol{x}_{<i})$ for all $i \in [m]$. The claim follows. □

6 Prequential Compression and Inductive Inference

In this section, we prove Theorem 2.3. We start with the formal definition of time-bounded randomized Kolmogorov complexity.

Definition 6.1. *The t-time-bounded randomized Kolmogorov complexity of x given z is defined as*

$$\mathsf{rK}^t(x \mid z) \coloneqq \min\left\{|d| \;\middle|\; \Pr_{r \sim \{0,1\}^t}\left[U^{t,z,r}(d) \text{ halts and outputs } x\right] \geq \frac{2}{3}\right\}.$$

Theorem 6.1 (Restatement of Theorem 2.3)**.** *There exists a sequence $\{\mathcal{L}_{c,\epsilon}\}_{c,\epsilon^{-1} \in \mathbb{N}}$ of randomized polynomial-time algorithms such that for every family $\mathcal{X} = \{\mathcal{X}^N\}_{N \in \mathbb{N}}$ of distributions $\mathcal{X}^N$ over $\{0,1\}^N$, the following are equivalent.*

1. *For every constant $\epsilon > 0$, there exist constants $n \in \mathbb{N}$ and $c \in \mathbb{N}$ such that for all large $N \in \mathbb{N}$, $\mathcal{X}^N$ is (n, ϵ)-prequential q^t-compressible for $t(m) \coloneqq m^c$.*

2. *For all sufficiently small constants $\epsilon > 0$ and all $n \in \mathbb{N}$ with $n \geq (1/\epsilon)^{1.01}$, there exist constants $c \in \mathbb{N}$ and $N_0 \in \mathbb{N}$ such that $\mathcal{X}^N$ is (n, ϵ)-prequential rK^t-compressible for the constant function $t \equiv N^c$ and for all $N \geq N_0$. That is,*

$$\mathop{\mathbb{E}}_{x \sim \mathcal{X}^N} \left[\sum_{j=1}^{m} \mathsf{rK}^t(y_j \mid y_{<j}) \right] \leq \epsilon N + \mathrm{H}(\mathcal{X}^N),$$

   where $m \coloneqq \lceil N/n \rceil$ and y_j denotes the concatenation of the i-th bit of x for all $i \in \mathbb{N}$ such that $m(j-1) < i \leq \min\{mj, N\}$.

3. *For every constant $\epsilon > 0$, there exists a randomized polynomial-time algorithm L such that for all sufficiently large N,*

$$\Pr_{\substack{i \sim [N] \\ x \sim \mathcal{X}^N}} \left[L(x_{<i}) \equiv_\epsilon \left(\mathcal{X}_i^N \mid \mathcal{X}_{<i}^N = x_{<i} \right) \right] \geq 1 - \epsilon,$$

   where $\mathcal{X}_i^N$ denotes the i-th bit of $\mathcal{X}^N$.

4. *For every constant $\epsilon > 0$, there exists a constant $c \in \mathbb{N}$ such that for all sufficiently large N,*

$$\Pr_{\substack{i \sim [N] \\ x \sim \mathcal{X}^N}} \left[\mathcal{L}_{c,\epsilon}(x_{<i}) \equiv_\epsilon \left(\mathcal{X}_i^N \mid \mathcal{X}_{<i}^N = x_{<i} \right) \right] \geq 1 - \epsilon.$$

We first present a inductive inference algorithm for all prequential compressible distributions.

Lemma 6.1. *Let $n \in \mathbb{N}$, $c \in \mathbb{N}$, and $\epsilon^{-1} \in \mathbb{N}$ be constants such that ϵ is sufficiently small. There exists a randomized polynomial-time algorithm $\mathcal{L}_{c,\epsilon}^n$ such that for any family $\mathcal{X} = \{\mathcal{X}^N\}_N$ of (n, ϵ)-prequential q^t-compressible distributions $\mathcal{X}^N$ over $\{0,1\}^N$ for $t(m) \coloneqq m^c$, it holds that for all sufficiently large $N \in \mathbb{N}$,*

$$\Pr_{\substack{i \sim [N] \\ x \sim \mathcal{X}^N}} \left[\mathcal{L}_{c,\epsilon}^n(x_{<i}) \equiv_\delta \left(\mathcal{X}_i^N \mid \mathcal{X}_{<i}^N = x_{<i} \right) \right] \geq 1 - \delta,$$

where $\delta \coloneqq \epsilon^{1/4}$.

Proof. For notational simplicity, we simply write $\mathcal{X}^N$ instead of $\mathcal{X}$ below. For simplicity, we assume that n exactly divides N.

We define a randomized polynomial-time algorithm $\mathcal{L}_{c,\epsilon}^n$ as follows. Given as input $x_{<i} \in \{0,1\}^{i-1}$, it partitions $x_{<i}$ into consecutive blocks $y_1, \ldots, y_j$ so that $y_1 \circ \cdots \circ y_j = x$, $|y_{j'}| = n$ for all $j' < j$ and $|y_j| < n$, defines $t \coloneqq t(|y_{<j}|)$, runs the universal Turing machine $U^{t, y_{<j}}(d)$ for a random program $d \sim \{0,1\}^t$ to obtain its output w, and outputs the $(|y_j|+1)$-th bit of w if y_j is the prefix of w; otherwise, it repeats this trial up to $2^{O(n)} \cdot O(\log \epsilon^{-1})$ times; if all trials fail, output an arbitrary bit. Since n, ϵ, and c are constants, this algorithm runs in polynomial time.

Since $|y_j| < n$, each trial of $\mathcal{L}_{c,\epsilon}^n$ succeeds with probability at least $2^{-O(n)}$. Thus, the algorithm simulates the conditional distribution of the continuation of y_j under the t-time-bounded universal distribution $\mathrm{Q}^{t, y_{<j}}$ with probability at least $1 - \epsilon$. It follows that for every x,

$$\mathcal{L}_{c,\epsilon}^n(x_{<i}) \equiv_\epsilon \left(\mathrm{Q}_k^{t, y_{<j}} \;\middle|\; \mathrm{Q}_{<k}^{t, y_{<j}} = y_j \right),$$

where $k \coloneqq |y_j| + 1 \in [n]$ and $\mathrm{Q}_k^{t,y_{<j}}$ denotes the k-th bit of $\mathrm{Q}^{t,y_{<j}}$.

For each j, let Y_j be the random variable of the j-th block of $x \sim \mathcal{X}$. Since $\mathcal{X}$ is (n,ϵ)-prequential q^t-compressible, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{j\sim[m]}\left[\mathrm{KL}\big(Y_j \mid Y_{<j} \,\big\|\, \mathrm{Q}^{t,Y_{<j}} \mid Y_{<j}\big)\right] &= \frac{1}{m}\sum_{j=1}^{m}\left(\mathbb{E}\left[\mathrm{q}^t(Y_j \mid Y_{<j})\right] - \mathrm{H}(Y_j \mid Y_{<j})\right)\\ &= \frac{1}{m}\left(\mathbb{E}\left[\sum_{j=1}^{m}\mathrm{q}^t(Y_j \mid Y_{<j})\right] - \mathrm{H}(\mathcal{X})\right)\\ &\le \frac{\epsilon N}{m} \le \epsilon n.\end{aligned}$$

Fix $y_{<j}$ and define $Y'(y_{<j}) \coloneqq (Y_j \mid Y_{<j} = y_{<j})$. Applying the chain rule for KL divergence to Y'_j, we obtain

$$\begin{aligned}&\mathrm{KL}\big(Y_j \mid Y_{<j} = y_{<j} \,\big\|\, \mathrm{Q}^{t,Y_{<j}} \mid Y_{<j} = y_{<j}\big)\\ &= \mathrm{KL}\big(Y'(y_{<j}) \,\big\|\, \mathrm{Q}^{t,y_{<j}}\big)\\ &= \sum_{k=1}^{n}\mathrm{KL}\Big(Y'(y_{<j})_k \mid Y'(y_{<j})_{<k} \,\Big\|\, \mathrm{Q}_k^{t,y_{<j}} \mid \mathrm{Q}_{<k}^{t,y_{<j}}\Big),\end{aligned}$$

where the subscript k means the k-th bit of the string. Therefore,

$$\begin{aligned}&\mathop{\mathbb{E}}_{j\sim[m],k\sim[n]}\left[\mathrm{KL}\Big(\mathcal{X}_{n(j-1)+k} \mid \mathcal{X}_{<n(j-1)+k} \,\Big\|\, \mathrm{Q}_k^{t,Y_{<j}} \mid Y_{<j} \circ \mathrm{Q}_{<k}^{t,Y_{<j}}\Big)\right]\\ &= \frac{1}{n}\mathop{\mathbb{E}}_{j\sim[m],y_{<j}}\left[\sum_{k=1}^{n}\mathrm{KL}\Big(Y'(y_{<j})_k \mid Y'(y_{<j})_{<k} \,\Big\|\, \mathrm{Q}_k^{t,y_{<j}} \mid \mathrm{Q}_{<k}^{t,y_{<j}}\Big)\right]\\ &= \frac{1}{n}\mathop{\mathbb{E}}_{j\sim[m],y_{<j}}\left[\mathrm{KL}\big(Y_j \mid Y_{<j} = y_{<j} \,\big\|\, \mathrm{Q}^{t,Y_{<j}} \mid Y_{<j} = y_{<j}\big)\right]\\ &= \frac{1}{n}\mathop{\mathbb{E}}_{j\sim[m]}\left[\mathrm{KL}\big(Y_j \mid Y_{<j} \,\big\|\, \mathrm{Q}^{t,Y_{<j}} \mid Y_{<j}\big)\right]\\ &\le \epsilon.\end{aligned}$$

By Markov's inequality and the non-negativity of the KL divergence, with probability at least $1-\sqrt{\epsilon}$ over $(j,k) \sim [m]\times[n]$ (which corresponds to a uniformly random index $i = n(j-1)+k \sim [N]$) and $x \sim \mathcal{X}$,

$$\mathrm{KL}\Big(\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i} \,\Big\|\, \mathrm{Q}_k^{t,y_{<j}} \mid Y_{<j} \circ \mathrm{Q}_{<k}^{t,Y_{<j}} = x_{<i}\Big) \le \sqrt{\epsilon}.$$

From Pinsker's inequality,

$$\Delta_{\mathrm{tv}}\Big((\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i}), \Big(\mathrm{Q}_k^{t,y_{<j}} \;\Big|\; Y_{<j} \circ \mathrm{Q}_{<k}^{t,Y_{<j}} = x_{<i}\Big)\Big) \le \frac{\epsilon^{1/4}}{\sqrt{2}}.$$

By the triangle inequality for total variation distance, for every i, we have

$$\Delta_{\mathrm{tv}}\big((\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i}), \mathcal{L}^n_{c,\epsilon}(x_{<i})\big) \le \frac{\epsilon^{1/4}}{\sqrt{2}} + \epsilon.$$

Thus, we obtain

$$\Pr_{\substack{i\sim[N]\\ x\sim\mathcal{X}}}\left[\Delta_{\mathrm{tv}}\left((\mathcal{X}_i \mid \mathcal{X}_{<i}=x_{<i}), \mathcal{L}^n_{c,\epsilon}(x_{<i})\right) \le \frac{\epsilon^{1/4}}{\sqrt{2}}+\epsilon\right] \ge 1-\sqrt{\epsilon}.$$

The claim follows by observing that $\epsilon^{1/4}/\sqrt{2}+\epsilon \le \delta$ and $\sqrt{\epsilon} \le \delta$ for all sufficiently small $\epsilon > 0$. □

We will use the following pseudo-deterministic arithmetic coding.

Theorem 6.2 ([HLN24, Theorem 4.6]). *Let $\mathcal{D} = \{\mathcal{D}_k\}_{k\in\mathbb{N}}$ be a family of distributions where each $\mathcal{D}_k$ is supported on $\{0,1\}^{\ell(k)}$ for an efficiently computable function $\ell(\cdot)$. Suppose there exists a nonuniform advice sequence $\{\alpha_k\}$ such that, for each $k \in \mathbb{N}$,*

- *for any z, the next-bit distribution $\mathsf{Next}_1(z;\mathcal{D}_k)$ is polynomial-time samplable given z and α_k;*[7]
- *for each $b \in \{0,1\}$, $\Pr[\mathsf{Next}_1(z;\mathcal{D}_k)=b] \ge 1/\gamma(k)$ for a universal function γ.*

Then, there exists a polynomial τ such that, for each $k \in \mathbb{N}$ and every $x \in \mathsf{Support}(\mathcal{D}_k)$,

$$\mathsf{rK}^{\tau(k,|\alpha_k|)}(x \mid \alpha_k) \le -\log \mathcal{D}_k(x) + O(\log(k\,\gamma(k)\,\ell(k))).$$

We now present prequential compression from an inductive inference algorithm.

Lemma 6.2. *Let $\epsilon > 0$ and $\mathcal{X} = \{\mathcal{X}^N\}_{N\in\mathbb{N}}$. Assume that there exists a randomized polynomial-time algorithm L such that for all sufficiently large N,*

$$\Pr_{\substack{i\sim[N]\\ x\sim\mathcal{X}^N}}\left[L(x_{<i}) \equiv_\epsilon \left(\mathcal{X}^N_i \mid \mathcal{X}^N_{<i}=x_{<i}\right)\right] \ge 1-\epsilon.$$

Then, for every $n \in \mathbb{N}$, for $\delta := O\left(\epsilon^{1/6} + \frac{\log(n/\epsilon)}{n} + \sqrt{n\epsilon}\right)$, there exist constants $c \in \mathbb{N}$ and $N_0 \in \mathbb{N}$ such that $\mathcal{X}^N$ is (n,δ)-prequential rK^t-compressible for the constant function $t \equiv N^c$ and for all $N \ge N_0$.

Proof. For notational simplicity, we simply write $\mathcal{X}$ instead of $\mathcal{X}^N$. We construct a prequential compressor for $\mathcal{X}$ from the inference algorithm L. A high-level idea is that if $x \sim \mathcal{X}$ is partitioned into $m = \lceil N/n \rceil$ consecutive blocks of length n (the last block has length at most n), then most blocks are predictable, and such blocks can be compressed.

For every $j \in [m]$, let B_j denote the set of indices in the j-th block and $y_j \in \{0,1\}^{\le n}$ denote the j-th block for each $j \in [m]$. By Markov's inequality, with probability $1-\sqrt{\epsilon}$ over $i \sim [N]$, it holds that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}[\Delta_{\mathrm{tv}}(L(\mathcal{X}_{<i}), (\mathcal{X}_i \mid \mathcal{X}_{<i}=x_{<i}))] \le \sqrt{\epsilon} \tag{9}$$

Let G be the set of indices $j \in [m]$ such that Equation (9) holds for all $i \in B_j$. Then, the size of G is at least $m - \sqrt{\epsilon}N = m\cdot(1-\sqrt{\epsilon}n)$.

[7] In the original statement, they assume *next-bits predictability*, i.e., that every next-bit probability is approximable within fixed polynomial accuracy and confidence error. This property trivially follows from samplability by standard empirical estimation. Moreover, while their theorem is stated for the uniform computational model, the same proof applies to the nonuniform model, i.e., in the presence of an auxiliary input sequence $\{\alpha_k\}$.

Fix an index $j \in G$ and an index $i \in B_j$. We define P as the randomized algorithm that, given $x_{<i}$ as input, output $L(x_{<i})$ with probability $1 - 2\alpha$ and output a uniformly random bit $b \sim \{0,1\}$ with probability 2α, where we define $\alpha := \sqrt{\epsilon}$. Then we have

$$\Delta_{\mathrm{tv}}(P(x_{<i}), L(x_{<i})) \le 2\alpha.$$

By the triangle inequality for Δ_{tv} and Equation (9), we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}[\Delta_{\mathrm{tv}}(P(x_{<i}), (\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i}))] \le 3\sqrt{\epsilon}.$$

By Markov's inequality,

$$\Pr_{x\sim\mathcal{X}}\Big[\Delta_{\mathrm{tv}}(P(x_{<i}), (\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i})) \ge \epsilon^{1/3}\Big] \le 3\epsilon^{1/6}.$$

Applying Proposition 4.1, since $\Pr_P[P(x_{<i}) = b] \ge \alpha$ for every $b \in \{0,1\}$, we obtain

$$\Pr_{x\sim\mathcal{X}}\Big[\mathrm{KL}(\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i} \parallel P(x_{<i})) \le O(\epsilon^{2/3}/\alpha)\Big] \ge 1 - 3\epsilon^{1/6}.$$

and thus

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}[\mathrm{KL}(\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i} \parallel P(x_{<i}))] \le O\Big(\epsilon^{2/3}/\alpha\Big) + O\Big(\epsilon^{1/6}\Big) = O\Big(\epsilon^{1/6}\Big).$$

We now take the sum over all $i \in B_j$ and obtain

$$\sum_{i\in B_j} \mathop{\mathbb{E}}_{x\sim\mathcal{X}}[\mathrm{KL}(\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i} \parallel P(x_{<i}))] \le O\Big(n \cdot \epsilon^{1/6}\Big). \tag{10}$$

Fix an arbitrary index $j \in G$. Define C to be a randomized algorithm that recursively uses the predictor P to construct a string of length n; specifically, given y as input, C recursively calculates $b_i := P(y \circ b_1 \cdots b_{i-1})$ for each $i \in [n]$ and outputs $b_1 \cdots b_n \in \{0,1\}^n$. Let Y_j denotes the j-th block of $\mathcal{X}$. By the chain rule for KL and Equation (10), we have

$$\begin{aligned}
&\mathrm{KL}(Y_j \mid Y_{<j} \parallel C(Y_{<j})) \\
&= \mathop{\mathbb{E}}_{y_{<j}\sim Y_{<j}}[\mathrm{KL}(Y_j \mid Y_{<j} = y_{<j} \parallel C(y_{<j}))] \\
&= \mathop{\mathbb{E}}_{x\sim\mathcal{X}}\left[\sum_{i\in B_j} \mathrm{KL}(\mathcal{X}_i \mid \mathcal{X}_{<i} = x_{<i} \parallel P(x_{<i}))\right] \\
&\le O\Big(n \cdot \epsilon^{1/6}\Big).
\end{aligned}$$

For any $y_{<j}$ in the support of $Y_{<j}$, let $\mathcal{D}_{y_{<j}}$ denote the distribution induced by $C(y_{<j})$. Observe that

$$\begin{aligned}
&\mathbb{E}\big[-\log \mathcal{D}_{Y_{<j}}(Y_j)\big] \\
&= \mathrm{H}(Y_j \mid Y_{<j}) + \mathrm{KL}(Y_j \mid Y_{<j} \| C(Y_{<j})) \\
&\le \mathrm{H}(Y_j \mid Y_{<j}) + O\Big(n \cdot \epsilon^{1/6}\Big),
\end{aligned}$$

We apply Theorem 6.2 to $\mathcal{D}_{y_{<j}}$ under the parameter settings $k := \epsilon^{-1}$ (here we apply Item 3 only for $\epsilon^{-1} \in \mathbb{N}$), $\ell(k) := n$, and advice $\alpha_k := y_{<j}$. Then the assumptions of the theorem are satisfied with $\gamma(k) = 1/\alpha$ using P as the sampler for next bits. Hence, for every y_j in the support of $\mathcal{D}_{y_{<j}}$,

$$\begin{aligned}
\mathsf{rK}^{\mathsf{poly}(\epsilon^{-1}, |y_{<j}|)}(y_j \mid y_{<j}) &\le -\log \mathcal{D}_{y_{<j}}(y_j) + O(\log(k\,\gamma(k)\,\ell(k))) \\
&\le -\log \mathcal{D}_{y_{<j}}(y_j) + O(\log(n/\epsilon)).
\end{aligned}$$

Taking the expectation and summing over $j \in [m]$, we obtain

$$\begin{aligned}
&\sum_{j\in[m]} \mathbb{E}\Big[\mathsf{rK}^{\mathsf{poly}(N)}(Y_j \mid Y_{<j})\Big] \\
&\le \sum_{j\in G} \big(\mathbb{E}\big[-\log \mathcal{D}_{Y_{<j}}(Y_j)\big] + O(\log(n/\epsilon))\big) + \sum_{j\in[m]\setminus G} (n + O(1)) \\
&\le \sum_{j\in G} \Big(\mathrm{H}(Y_j \mid Y_{<j}) + O\Big(n\cdot\epsilon^{1/6} + \log(n/\epsilon)\Big)\Big) + m\sqrt{n}\epsilon\cdot(n+O(1)) \\
&\le \mathrm{H}(\mathcal{X}) + N\cdot O\bigg(\epsilon^{1/6} + \frac{\log(n/\epsilon)}{n} + \sqrt{n}\epsilon\bigg).
\end{aligned}$$

We define δ so that this is bounded by $\mathrm{H}(\mathcal{X}) + \delta N$, which completes the proof. □

We observe that q^t-compression implies rK^t-compression.

Lemma 6.3. *For all sufficiently large $n \in \mathbb{N}$ and all $\epsilon > 0$, the following holds. Let $\mathcal{X}$ be a (n,ϵ)-prequential rK^t-compressible distribution over $\{0,1\}^N$ for some $N \ge n^4$ and for some constant function $t \equiv N^c$. Then, $\mathcal{X}$ is (n,δ)-prequential $q^{t'}$-compressible for $t'(m) \coloneqq m^{c'}$ for some constant c' and $\delta \coloneqq \epsilon + O\Big(\frac{\log n}{n}\Big)$.*

Proof. We first claim that there exists a constant c_0 such that for all but finitely many x and z and all $t \ge |x| + |z|$,

$$\mathrm{q}^{t^{c_0}}(x \mid z) \le \mathsf{rK}^t(x \mid z) + O(\log|x|). \tag{11}$$

Let d^* be the string of length $\mathsf{rK}^t(x \mid z)$ such that $U^{t,r,z}(d^*)$ outputs x with probability $\frac{2}{3}$ over a random $r \sim \{0,1\}^t$. Consider the constant-size program that, given z as input, randomly chooses $s \sim [|x| + O(1)], d \sim \{0,1\}^s$, and $r \sim \{0,1\}^t$ and outputs $U^{t,r,z}(d)$. The output of this program is equal to x when $s = |d^*|$, $d = d^*$, and $U^{t,r,z}(d^*)$ outputs x, which happens with probability at least $\frac{1}{|x|+O(1)} \cdot 2^{-|d^*|} \cdot \frac{2}{3} = \Omega\big(2^{-|d^*|-\log|x|}\big)$. Thus, $\mathrm{Q}^{t^{c_0}}(x \mid z) \ge \Omega\Big(2^{-\mathsf{rK}^t(x|z)-\log|x|}\Big)$ for a sufficiently large constant c_0, which completes the proof of the claim.

We define $t'(m) \coloneqq m^{2c_0c}$. Applying Equation (11) to the blocks $(y_1,\dots,y_m)$ of $x \sim \mathcal{X}$, we have

$$\sum_{j=1}^{m} \mathrm{q}^{t'(|y_{<j}|)}(y_j \mid y_{<j}) \le \sum_{j=\sqrt{N}}^{m} \big(\mathsf{rK}^t(y_j \mid y_{<j}) + O(\log n)\big) + \sqrt{N}\cdot(n+O(1)),$$

where we used the fact that $t^{c_0} = N^{c_0c} \le j^{2c_0c} = t'(j) \le t'(|y_{<j}|)$ for every $j \ge \sqrt{N}$. Taking the expectation of the first term over $x \sim \mathcal{X}$, we obtain

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}\left[\sum_{j=\sqrt{N}}^{m} \big(\mathsf{rK}^t(y_j \mid y_{<j}) + O(\log n)\big)\right] \le \mathrm{H}(\mathcal{X}) + \epsilon N + m\cdot O(\log n)$$

by the (n,ϵ)-prequential rK^t-compressibility of $\mathcal{X}$. Since $m = \lceil N/n \rceil$, we obtain

$$m\cdot O(\log n) \le N\cdot O\bigg(\frac{\log n}{n}\bigg).$$

We also have $\sqrt{N} \cdot (n + O(1)) \le O(N/n)$ because $n^4 \le N$. Combining the inequalities above, we conclude that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{X}}\left[\sum_{j=1}^{m} \mathrm{q}^{t'(|y_{<j}|)}(y_j \mid y_{<j})\right] \le \mathrm{H}(\mathcal{X}) + N \cdot \left(\epsilon + O\left(\frac{\log n}{n}\right)\right).$$

□

Proof of Theorem 6.1. It is obvious that Item 4 implies Item 3.

The implication from Item 2 implies Item 1 follows from Lemma 6.3. Specifically, for a given constant $\epsilon > 0$, we choose $n = (1/\epsilon)^2$ in Item 2 and obtain that $\mathcal{X}^N$ is (n, ϵ)-prequential rK^t-compressible. By Lemma 6.3, $\mathcal{X}^N$ is (n, δ)-prequential $\mathrm{q}^{t'}$-compressible for $\delta = O\left(\frac{\log n}{n}\right) + \epsilon \le 2\epsilon$, where the last inequality holds for all sufficiently small $\epsilon > 0$, as desired.

To prove the implication from Item 1 to Item 3, assume that $\mathcal{X}^N$ is (n, ϵ)-prequential q^t-compressible for $t(m) \coloneqq m^c$. By Lemma 6.1, the algorithm $L = \mathcal{L}^n_{c,\epsilon^4}$ predicts the next bits of $\mathcal{X}^N$ with accuracy ϵ and confidence ϵ, which completes the proof of Item 3.

We prove the implication from Item 2 to Item 4 in a similar way. For each $\epsilon' > 0$ and c, we define $\epsilon \coloneqq (2\epsilon')^{1/4}$, $n \coloneqq (1/\epsilon)^2$ and $\mathcal{L}_{c,\epsilon'} \coloneqq \mathcal{L}^n_{c,\epsilon}$ using the inference algorithm of Lemma 6.1. Note that this choice of $\mathcal{L}_{c,\epsilon'}$ is independent of a specific distribution $\mathcal{X}$. Let $\epsilon' > 0$ be a given constant for which we aim to prove Item 4. By Item 2, $\mathcal{X}^N$ is (n, ϵ)-prequential rK^t-compressible for $t \equiv N^c$, where we choose $n \coloneqq (1/\epsilon)^2$. By Lemma 6.3, $\mathcal{X}^N$ is $(n, 2\epsilon)$-prequential $\mathrm{q}^{t'}$-compressible for some $t'(m) = m^{O(1)}$. By Lemma 6.1, $\mathcal{L}^n_{c,2\epsilon}$ predicts the next bits of $\mathcal{X}^N$ with accuracy and confidence $\delta = (2\epsilon)^{1/4}$ for some c. We set $\epsilon = (\epsilon')^4/2$ so that $\delta \le \epsilon'$, which shows that $\mathcal{L}_{c,\epsilon'} = \mathcal{L}^n_{c,2\epsilon}$ achieves the desired accuracy and confidence.

The implication from Item 3 to Item 2 follows from Lemma 6.2. Specifically, let ϵ be the constant that satisfies Item 3. Then $\mathcal{X}^N$ is (n, δ)-prequential rK^t-compressible for some constant function $t = \mathsf{poly}(N)$ and $\delta \coloneqq O\left(\epsilon^{1/6} + \frac{\log(n/\epsilon)}{n} + \sqrt{n\epsilon}\right)$. For all sufficiently small $\epsilon' > 0$ and $n \ge (1/\epsilon')^{1.01}$, define $\epsilon \coloneqq 1/n^6$. Then we have $\delta = O\left(\frac{\log n}{n}\right) = \frac{1}{n^{1-o(1)}} \le (\epsilon')^{1.01\cdot(1-o(1))} \le \epsilon'$, where we used that ϵ' is sufficiently small. It follows that $\mathcal{X}^N$ is (n, ϵ')-prequential rK^t-compressible, as desired. □

7 Inductive Inference via Advised Universal Extrapolation

In this section, we prove the following theorem via advised universal extrapolation.

Theorem 7.1 (Restatement of Theorem 2.2)**.** *If* $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$*, then when the total number of rounds m is explicitly provided to the inference algorithm, complexity-theoretic universal inductive inference is solvable in time $t^{O(\delta^{-1})}$ with round complexity $O(s\epsilon^{-2}\delta^{-1})$.*

7.1 KL Bound for Advised Universal Extrapolation

We first establish the following KL bound for advised universal extrapolation.

Lemma 7.1 (Advised Universal Extrapolation)**.** *If* $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$*, then there exist a constant c and a polynomial p satisfying the following for every $n, s, b, w \in \mathbb{N}$: Let Π be an s-size randomized Turing machine that produces at least $b \cdot w$ binary strings $x_1, \dots, x_{bw}, \dots \in \{0,1\}^n$ in t_Π $(\ge s)$ time. For each $i \in [b]$, let $\bar{x}^i = x_{(i-1)w+1} \circ \cdots \circ x_{iw}$. Then, for any $t \in \mathbb{N}$ with $t \ge p(t_\Pi)$,*

$$\mathop{\mathbb{E}}_{i\sim[b], j\sim[w]}\left[\mathrm{KL}\left(X^i_j | X^{\le i}_{<j} \middle\| \mathrm{Q}^{t, X^{<i}}_{[(j-1)n+1:jn]} | \mathrm{Q}^{t, X^{<i}}_{[(j-1)n]}\right)\right] \le \frac{c}{w} \cdot \left(\frac{s}{b} + \log t\right),$$

where $X^{<i}$, $X^i_{<j}$, *and* X^i_j *represent distributions of* $(\bar{x}^1, \ldots, \bar{x}^{i-1})$, $(x_{(i-1)w+1}, \cdots, x_{(i-1)w+j-1})$, *and* $x_{(i-1)w+j}$ *chosen according to* Π, *respectively, and* $X^{\leq i}_{<j} := (X^{<i}, X^i_{<j})$.

In particular, if $b \geq s/\log t$ *and* $w \geq 2c\epsilon^{-1}\delta^{-1}\log t$ *for* $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$,

$$\Pr_{i, x^{<i}, j, x^i_{<j}}\left[\mathrm{KL}\left(X^i_j | x^{\leq i}_{<j} \middle\| \mathsf{Next}_n\left(\bar{x}^i_{<j}; \mathrm{Q}^{t, x^{<i}}\right)\right) \leq \epsilon\right] \geq 1 - \delta,$$

where $i \sim [b]$, $j \sim [w]$, $x^{<i} \sim X^{<i}$, $x^i_{<j} \sim X^i_{<j}$, $x^{\leq i}_{<j} := (x^{<i}, x^i_{<j})$, *and* $X^i_j | x^{\leq i}_{<j}$ *represents the conditional distribution of* $x_{(i-1)w+j}$ *given* $X^{\leq i}_{<j} = x^{\leq i}_{<j}$.

Proof. Let τ be the polynomial in Lemma 5.2, and τ' be the polynomial in Proposition 4.6.

For each $i \in [b]$, let $\bar{X}^i$ be the distribution of $\bar{x}^i$, $x^{\leq i} = (\bar{x}^1), \ldots, \bar{x}^i$, and $X^{\leq i}$ be its distribution.

From Lemma 5.2, we obtain that for any $t \geq O(t_\Pi)$ (recall that $bw \leq t_\Pi$ since all strings are generated in t_Π time),

$$\sum_{i=1}^{b} \mathrm{q}^{\tau(t)}(\bar{x}^i, \mid x^{<i}) \leq \mathrm{q}^t(x^{\leq b}) + b \cdot O(\log t).$$

From Proposition 4.4, for a large enough polynomial τ'', we have that for any $t \geq \tau''(t_\Pi)$,

$$\begin{aligned}\mathrm{q}^t(\bar{x}^1, \ldots, \bar{x}^b) &\leq O(s) + O(\log t) - \log \Pr\left[X^{\leq b} = x^{\leq b}\right] \\ &= O(s) + O(\log t) - \sum_{i=1}^{b} \log \Pr\left[\bar{X}^i = \bar{x}^i \middle| X^{<i} = x^{<i}\right].\end{aligned}$$

From the two inequalities above, we have

$$\sum_{i=1}^{b} \left(q^t(\bar{x}^i \mid x^{<i}) + \log \Pr\left[\bar{X}^i = \bar{x}^i \middle| X^{<i} = x^{<i}\right]\right) \leq O(s) + b \cdot O(\log t).$$

From Proposition 4.6, we have for every $t' \geq \tau'(t)$

$$\mathrm{q}^{t'}(\bar{x}^i \mid x^{<i}) \leq \mathrm{q}^{\tau(t)}(\bar{x}^i \mid x^{<i}) + O(\log t).$$

Therefore, by taking large enough polynomial p, we obtain that for every $t \geq p(t_\Pi)$,

$$\sum_{i=1}^{b} \left(q^t(\bar{x}^i \mid x^{<i}) + \log \Pr\left[\bar{X}^i = \bar{x}^i \middle| X^{<i} = x^{<i}\right]\right) \leq O(s) + b \cdot O(\log t).$$

Notice that

$$q^t(\bar{x}^i \mid x^{<i}) + \log \Pr\left[\bar{X}^i = \bar{x}^i \middle| X^{<i} = x^{<i}\right] = \log \frac{\Pr\left[\bar{X}^i = \bar{x}^i \middle| X^{<i} = x^{<i}\right]}{\Pr[\mathrm{Q}^{t, x^{<i}} = \bar{x}^i]}.$$

Thus, by taking expectation over $\bar{X}^1, \ldots, \bar{X}^b$, we get

$$\sum_{i=1}^{b} \mathrm{KL}\left(\bar{X}^i | X^{<i} \middle\| \mathrm{Q}^{t, X^{<i}}\right) \leq O(s) + b \cdot O(\log t).$$

By taking a large enough constant $c > 0$,

$$\mathop{\mathbb{E}}_{i\sim[b]}\left[\mathrm{KL}\left(\bar{X}^i|X^{<i}\middle\|\mathrm{Q}^{t,X^{<i}}\right)\right] = \frac{1}{b}\sum_{i=1}^{b}\mathrm{KL}\left(\bar{X}^i|X^{<i}\middle\|\mathrm{Q}^{t,X^{<i}}\right) \le c\cdot\left(\frac{s}{b}+\log t\right). \tag{12}$$

For each $i \in [b]$, we apply the chain rule for KL divergence and obtain

$$\mathrm{KL}\left(\bar{X}^i|X^{<i}\middle\|\mathrm{Q}^{t,X^{<i}}\right) = \sum_{j=1}^{w}\mathrm{KL}\left(X^i_j|X^{\le i}_{<j}\middle\|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n+1:jn]}|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n]}\right).$$

Along with Equation (12), we derive the first part of the lemma as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{i\sim[b],j\sim[w]}\left[\mathrm{KL}\left(X^i_j|X^{\le i}_{<j}\middle\|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n+1:jn]}|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n]}\right)\right]\\
&= \frac{1}{w}\cdot\mathop{\mathbb{E}}_{i\sim[b]}\left[\sum_{j=1}^{w}\mathrm{KL}\left(X^i_j|X^{\le i}_{<j}\middle\|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n+1:jn]}|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n]}\right)\right]\\
&= \frac{1}{w}\cdot\mathop{\mathbb{E}}_{i\sim[b]}\left[\mathrm{KL}\left(\bar{X}^i|X^{<i}\middle\|\mathrm{Q}^{t,X^{<i}}\right)\right]\\
&\le \frac{c}{w}\cdot\left(\frac{s}{b}+\log t\right)
\end{aligned}$$

Next, we derive the second part of the lemma from the above.
If $b \ge s/\log t$ and $w \ge 2c\epsilon^{-1}\delta^{-1}\log t$ are satisfied for $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, then

$$\frac{c}{w}\cdot\left(\frac{s}{b}+\log t\right) \le \frac{\epsilon\delta}{2\log t}\cdot\left(\frac{s}{s/\log t}+\log t\right) = \epsilon\delta.$$

Therefore,

$$\mathop{\mathbb{E}}_{i\sim[b],j\sim[w]}\left[\mathrm{KL}\left(X^i_j|X^{\le i}_{<j}\middle\|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n+1:jn]}|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n]}\right)\right] \le \epsilon\delta.$$

Notice that, by the definition of conditional KL divergence,

$$\mathrm{KL}\left(X^i_j|X^{\le i}_{<j}\middle\|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n+1:jn]}|\mathrm{Q}^{t,X^{<i}}_{[(j-1)n]}\right) = \mathop{\mathbb{E}}_{x^{<i},x^i_{<j}}\left[\mathrm{KL}\left(X^i_j|\bar{x}^{\le i}_{<j}\middle\|\mathsf{Next}_n\left(x^i_{<j};\mathrm{Q}^{t,x^{<i}}\right)\right)\right].$$

Since KL divergence is always nonnegative, by Markov's inequality,

$$\Pr_{i,x^{<i},j,x^i_{<j}}\left[\mathrm{KL}\left(X^i_j|x^{\le i}_{<j}\middle\|\mathsf{Next}_n\left(\bar{x}^i_{<j};\mathrm{Q}^{t,x^{<i}}\right)\right) \le \epsilon\right] \ge 1-\delta,$$

as desired. □

7.2 Inductive Inference via Advised Universal Extrapolation

Now we provide the formal proof of Theorem 7.1.

Proof of Theorem 7.1. Let p_0 and c_0 be the maximum of the polynomials and constants in Lemmas 4.7, 5.3 and 7.1, respectively. For each $t \in \mathbb{N}$, let $t' := p_0(t)$ and $t'' := p_0(t')$ below. Without loss of generality, we assume that, $t'' \ge t' \ge 2$, $c_0 \ge 1$, and $c_0 \log t' \ge \log t''$ for each t.

For each $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ with $m \geq 17c_0^2 \cdot s\epsilon^{-2}\delta^{-1}$, let $b = \lceil s/\log t' \rceil$. We also define $w \in \mathbb{N}$ as the largest integer such that $b \cdot w \leq m$. Let $\tilde{m} := b \cdot w$ ($\leq m$). The proof is based on the advise universal extrapolation as in Lemma 7.1 on $\tilde{m}$ strings $x_1, \ldots, x_{\tilde{m}}$.

Below, we assume that $c_0 \log \delta^{-1} \leq s$. Otherwise, $2^s \leq \delta^{-c_0} = \mathsf{poly}(\delta^{-1})$. In this case, we can use universal extrapolation without any advice that works in time $\mathsf{poly}(2^s, \epsilon^{-1}) \leq \mathsf{poly}(\delta^{-1}, \epsilon^{-1})$ (this is exactly the same setting as [HN23]) to extrapolate a prefix string produced by an s-size program under Q^t with statistical error ϵ.

We first verify that a random position $i \sim [m]$ almost falls in $[\tilde{m}]$ (thus ignoring $x_{\tilde{m}+1}, \ldots, x_m$ does not much affect the confidence error). Since $m < b(w+1) = \tilde{m} + b$, we have

$$\Pr_{i \sim [m]}[i > \tilde{m}] = \frac{m - \tilde{m}}{m} < \frac{b}{m} \leq \left(\frac{s}{\log t'} + 1\right) \cdot \frac{1}{17c_0^2 \cdot s\epsilon^{-2}\delta^{-1}} \leq \frac{\delta}{17c_0^2\epsilon^{-2}\log t'} + \frac{\delta}{17c_0^2 \cdot s\epsilon^{-2}} \leq \frac{2}{17}\delta.$$

From the assumption, there is no auxiliary-input one-way function by Lemma 4.2 (or Proposition C.1) and thus there exists the polynomial-time randomized algorithm UE in Lemma 4.8.

Now we present the construction of the algorithm L based on UE. Given an input prefix $x_{<i}$ and parameters $\mathsf{param} = (n, m, s, t, \epsilon^{-1}, \delta^{-1})$, the algorithm L first computes $t', t'', w, b, \tilde{m}$ as defined above. If $i > \tilde{m}$, the algorithm halts with an arbitrary message (we ignore this case as discussed earlier). Otherwise, if $i \leq \tilde{m}$, the algorithm computes the unique pair $(i', j') \in [b] \times [w]$ such that $i = (i'-1) \cdot w + j'$, and outputs a sample from

$$\mathsf{UE}\left(\bar{x}^{i'}_{<j'};\, x^{<i'},\, 1^{\langle n, t'', 2\epsilon^{-1}, 2^{c_0}t' \cdot t'^{3c_0\delta^{-1}}\rangle}\right),$$

where $\bar{x}^{i'}_{<j'} = x_{(i'-1)w+1} \circ \cdots \circ x_{(i'-1)w+j'-1}$ and $x^{<i'} = (x_1 \circ \cdots \circ x_w, \ldots, x_{(i'-2)w+1} \circ \cdots \circ x_{(i'-1)w})$. (Note that $x_{<i}$ is parsed into $(x^{<i'}, \bar{x}^{i'}_{<j'})$.) Observe that $t'^{3c_0\delta^{-1}} = \mathsf{poly}(t^{\delta^{-1}})$ since $t' = p_0(t)$.

It is easily verified that L halts in polynomial time in $t^{\delta^{-1}}$ (recall that $t \geq m \geq \Omega(s\epsilon^{-2}\delta^{-1})$ and $t \geq n$). Below, we verify the correctness under the condition $i \leq \tilde{m}$. Under this condition, a random choice of i is regarded as random choices of $(i', j') \sim [b] \times [w]$.

We first observe the lower bound on w as follows: Since $(w+1)b > m$,

$$w > \frac{m}{b} - 1 \geq \frac{17c_0^2 \cdot s\epsilon^{-2}\delta^{-1}}{s/\log t' + 1} - 1 \geq 16c_0^2\epsilon^{-2}\delta^{-1}\log t'.$$

Since $b \geq s/\log t' \geq s/\log t''$ and $w \geq 16c_0^2\epsilon^{-2}\delta^{-1}\log t' \geq 2c_0 \cdot 2\epsilon^{-2} \cdot 4\delta^{-1} \cdot \log t''$, Lemma 7.1 implies

$$\Pr_{i', \bar{x}^{<i'}, j', x^{i'}_{<j'}}\left[\mathrm{KL}\left(X_i | x_{<i} \middle\| \mathsf{Next}_n\left(\bar{x}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{x}^{<i'}}\right)\right) \leq \frac{\epsilon^2}{2}\right] \geq 1 - \frac{\delta}{4}. \tag{13}$$

If the event above occurs, it holds that

$$\Delta_{\mathrm{tv}}\left(\mathsf{Next}_n\left(\bar{x}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{x}^{<i'}}\right), X_i | x_{<i}\right) \leq \sqrt{2^{-1} \cdot \mathrm{KL}\left(X_i | x_{<i} \middle\| \mathsf{Next}_n\left(\bar{x}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{x}^{<i'}}\right)\right)} \leq \frac{\epsilon}{2},$$

where the first inequality follows from Pinsker's inequality.

In addition, since $b \geq 2s/\log t' \geq (s + c_0 \log \delta^{-1})/\log t'$, Lemmas 4.7 and 5.3 implies that for every j',

$$\Pr_{i', x^{<i'}, x^{i'}_{<j'}}\left[\mathrm{cd}^{t''}(x^{i'}_{<j'} \mid \bar{x}_{<i}) \leq 3c_0\delta^{-1}\log t' + \log t' + c_0\right] \geq 1 - \frac{\delta}{3}. \tag{14}$$

When this occurs, by Lemma 4.8,

$$\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{x}^{i'}_{<j'}; x^{<i'}, 1^{\langle n,t'',2\epsilon^{-1},2^{c_0}t'\cdot t'^{3c_0}\delta^{-1}\rangle}\right), \mathsf{Next}_n\left(\bar{x}^{i'}_{<j'}; \mathrm{Q}^{t'',x^{<i'}}\right)\right) \le \frac{\epsilon}{2}.$$

Therefore, if both the events in Equations (13) and (14) occur,

$$\Delta_{\mathrm{tv}}\left(L\left(x_{<i};\mathsf{param}\right), X_i | x_{<i}\right) \le \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

By the union bound, we conclude that

$$\Pr_{i,x_{<i}}\left[\Delta_{\mathrm{tv}}\left(L\left(x_{<i};\mathsf{param}\right), X_i | x_{<i}\right) \le \epsilon\right] \ge 1 - \left(\frac{2}{17}\delta + \frac{\delta}{4} + \frac{\delta}{3}\right) > 1 - \delta.$$

□

Remark 7.1. Theorem 7.1 indeed follows from the chain rule for q^t (i.e., the statement in Lemma 5.2) without assuming GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP. Recall that the proof of Theorem 7.1 invokes the assumption GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP at three points: (i) to obtain a tail bound for cd^t (Lemma 5.3); (ii) to derive the KL bound for advised universal extrapolation (Lemma 7.1); and (iii) to invert auxiliary-input one-way functions (Lemma 4.2 and Proposition C.1). For the first two purposes, the assumption GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP is used only to establish the chain rule for q^t. For the last, by Proposition C.1, it also depends solely on the chain rule for q^t.

8 Fully Polynomial-Time Inductive Inference

In this section, we take the inference algorithm from Section 7 as a subroutine and prove the following main theorem.

Theorem 8.1 (Restatement of Theorem 2.1)**.** *If* GapK$^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then complexity-theoretic universal inductive inference is solvable in time* $\mathsf{poly}(t)$*, with round complexity* $s \cdot p(n, \epsilon^{-1}, \delta^{-1})$ *for some polynomial* $p = \tilde{O}\left(n^2\epsilon^{-6}\delta^{-5}\right)$*.*

Throughout Section 8, we use the following notation for next-block probabilities. For any $x, y \in \{0,1\}^*$,

$$\mathcal{D}(y \,\|\, x) \;=\; \Pr\left[\,\mathsf{Next}_{|y|}(x;\mathcal{D}) = y\,\right].$$

In particular,

$$\mathrm{Q}^{t,z}(y \,\|\, x) \;=\; \Pr\left[\,\mathsf{Next}_{|y|}(x;\mathrm{Q}^{t,z}) = y\,\right].$$

8.1 Next-Bit Generator from the Nonexistence of AIOWF

We first formalize the notion of a next-bit generator. We then construct such a generator for the time-bounded advised universal distribution $\mathrm{Q}^{t,z}$, obtained via Lemma 4.8 from the nonexistence of auxiliary-input one-way functions. Our construction satisfies the additional technical properties required to establish our KL-divergence bound.

Definition 8.1 (Next-Bit Generator)**.** *For* $n \in \mathbb{N}$*, a next-bit generator is a randomized algorithm* G *that, on input* $x \in \{0,1\}^*$ *and an advice string* α*, outputs a pair* (p_0, p_1) *of real values in* $[0,1]$ *such that*

$$p_0 + p_1 \le 1$$

for every input x*, advice* α*, and choice of internal randomness. We write* $G(x;\alpha) = (p_0, p_1)$*.*

Given $m \in \mathbb{N}$ and advice α, a generator G induces a distribution $\mathcal{D}_G^\alpha$ over $\{0,1\}^{\le m}$ defined, for each $x \in \{0,1\}^{\le m}$, by

$$\mathcal{D}_G^\alpha(x) \;=\; \left(\prod_{i=1}^{|x|} p_{x_i \,\|\, x_{<i}}\right) \cdot p_{\varepsilon \,\|\, x},$$

where $(p_{0\,\|\,y}, p_{1\,\|\,y}) \leftarrow G(y;\alpha)$ for each $y \in \{0,1\}^{<m}$ and we set $p_{\varepsilon\,\|\,y} := 1 - (p_{0\,\|\,y} + p_{1\,\|\,y})$ if $|y| < m$, and $p_{\varepsilon\,\|\,y} := 1$ if $|y| = m$. If G runs in polynomial time and its outputs are rationals with denominators that are powers of two, then $\mathcal{D}_G^\alpha$ is exactly samplable in time $\mathsf{poly}(m, |\alpha|)$ by drawing each next bit according to $(p_{0\,\|\,y}, p_{1\,\|\,y}, p_{\varepsilon\,\|\,y})$ using $\mathsf{poly}(m, |\alpha|)$ random bits. In our concrete construction below we ensure this property as stated in the lemma. Thus, a polynomial-time next-bit generator can be regarded as a polynomial-time sampler for its induced distribution.

We will use the following specific next-bit generator.

Lemma 8.1. *If there exists no auxiliary-input one-way function, then there exist a polynomial-time next-bit generator $\tilde{Q}$, a constant c, and a polynomial γ such that for every $t, \epsilon^{-1}, d \in \mathbb{N}$, every $z \in \{0,1\}^*$, and every $x \in \{0,1\}^*$ with $t \ge c|x|$, the following holds: if we denote by (p_0, p_1) the output of $\tilde{Q}(x; z, 1^t, 1^{\epsilon^{-1}}, 1^d)$, then for each $b \in \{0,1\}$:*

1. *$p_b \ge 1/\gamma(t, \epsilon^{-1}, d)$, and p_b is a rational whose denominator is a power of two;*
2. *if $\mathrm{cd}^t(x \mid z) \le \log d$ and $\mathrm{cd}^t(x \circ b \mid z) \le \log d$, then*

$$\Pr_{\tilde{Q}}\left[(1-\epsilon)\cdot \mathrm{Q}^{t,z}(b \,\|\, x) \;\le\; p_b \;\le\; (1+\epsilon)\cdot \mathrm{Q}^{t,z}(b \,\|\, x)\right] \;\ge\; 1 - \mathsf{negl}(t).$$

First, we prove the following technical lemma for proving Lemma 8.1.

Lemma 8.2. *There exist a polynomial p and a constant c such that for every $x, z \in \{0,1\}^*$, every $t \ge c \cdot |x|$, and every $b \in \{0,1\}$,*

$$\mathrm{Q}^{t,z}(b\|x) \ge \frac{1}{2^{\mathrm{cd}^t(x \circ b | z)} \cdot p(t)}.$$

Proof. Let c be a sufficiently large constant, larger than the constants in Propositions 4.5 and 4.7. For every $x, z \in \{0,1\}^*$, every $t \ge c|x|$, and every $b \in \{0,1\}$, we have

$$\mathrm{Q}^{t,z}(b\|x) = \frac{\mathrm{Q}^{t,z}(x \circ b*)}{\mathrm{Q}^{t,z}(x*)} = \frac{\mathrm{Q}^{t^2,z}(x \circ b)}{\mathrm{Q}^{t,z}(x*)} \cdot \frac{\mathrm{Q}^{t,z}(x \circ b*)}{\mathrm{Q}^{t^2,z}(x \circ b)},$$

since $\mathrm{Q}^{t^2,z}(x) > 0$ by Proposition 4.5.

Let $X \subseteq \{0,1\}^t$ be the set of π such that $U(\pi)$ produces x as a prefix within t steps. Then there exists a universal constant c_0 such that, for each $\pi \in X$, there exists $\mathsf{append}(\pi, b) \in \{0,1\}^*$ of length at most $|\pi| + c_0 \log t$ such that $U^z(\pi')$ outputs $x \circ b$ and halts within $\tau(t)$ steps. This follows since one can simulate π in t steps and output b when the $(|x|+1)$-st bit is produced. Notice that this position can be specified using $O(\log t)$ bits with a universal hidden constant. Furthermore, $\mathsf{append}(\cdot, b)$ can be taken to be injective.

We only consider the case $t^2 \ge c_0 \log t$ by choosing c sufficiently large (which implies a lower bound on t). Let $Xb \subseteq \{0,1\}^{t^2}$ be the set of π such that $U^z(\pi)$ produces $x \circ b$ and halts in t^2 steps. By the above argument, for every $\pi \in X$, it holds that $\mathsf{append}(\pi, b) \in Xb$. Let $\mathsf{append}(X, b) = \{\mathsf{append}(\pi, b) : \pi \in X\}$. Then $\mathsf{append}(X, b) \subseteq Xb$.

Thus, for some universal polynomial $p_0(t) = t^{c_0}$,

$$\begin{aligned}
\mathrm{Q}^{t^2,z}(x \circ b) &\geq \sum_{\pi' \in Xb} 2^{-|\pi'|} \\
&\geq \sum_{\pi' \in \mathsf{append}(X,b)} 2^{-|\pi'|} \\
&\geq \sum_{\pi \in X} 2^{-|\pi| - c_0 \log t} \\
&= \frac{1}{p_0(t)} \cdot \sum_{\pi \in X} 2^{-|\pi|} = \frac{1}{p_0(t)} \cdot \mathrm{Q}^{t,z}(x*).
\end{aligned}$$

Rearranging,

$$\frac{\mathrm{Q}^{t^2,z}(x \circ b)}{\mathrm{Q}^{t,z}(x*)} \geq \frac{1}{p_0(t)}.$$

On the other hand, for some universal polynomial p_1,

$$\begin{aligned}
\mathrm{Q}^{t,z}(x \circ b*) &\geq \mathrm{Q}^{t,z}(x \circ b) \\
&= 2^{-\mathrm{q}^t(x \circ b|z)} \\
&= 2^{-\mathrm{cd}^t(x \circ b|z) - \mathrm{K}(x \circ b|z)} \\
&\geq 2^{-\mathrm{cd}^t(x \circ b|z) - \mathrm{q}^{t^2}(x \circ b|z) - \log p_1(t)} \\
&= \frac{\mathrm{Q}^{t^2,z}(x \circ b)}{2^{\mathrm{cd}^t(x \circ b|z)} \cdot p_1(t)},
\end{aligned}$$

where the second inequality follows from Proposition 4.7. Thus,

$$\frac{\mathrm{Q}^{t,z}(x \circ b*)}{\mathrm{Q}^{t^2,z}(x \circ b)} \geq \frac{1}{2^{\mathrm{cd}^t(x \circ b|z)} \cdot p_1(t)}.$$

Combining the inequalities above,

$$\mathrm{Q}^{t,z}(b\|x) \geq \frac{\mathrm{Q}^{t^2,z}(x \circ b)}{\mathrm{Q}^{t,z}(x*)} \cdot \frac{\mathrm{Q}^{t,z}(x \circ b*)}{\mathrm{Q}^{t^2,z}(x \circ b)} \geq \frac{1}{2^{\mathrm{cd}^t(x \circ b|z)} \cdot p_0(t) \cdot p_1(t)}.$$

This proves the lemma. □

We now proceed to the proof of Lemma 8.1.

Proof of Lemma 8.1. From the assumption that there is no auxiliary-input one-way function, we obtain the universal extrapolation algorithm UE in Lemma 4.8. In addition, let p be the polynomial of Lemma 8.2.

The algorithm $\tilde{Q}$, given a prefix string $x \in \{0,1\}^*$, advice $z \in \{0,1\}^*$, and parameters $t, \epsilon^{-1}, d \in \mathbb{N}$ (all given in unary), executes

$$\mathsf{UE}(x; z, 1^{\langle 1, t, \epsilon_{\mathsf{UE}}^{-1}, d \rangle})$$

independently N times, where the accuracy parameter is set to be

$$\epsilon_{\mathsf{UE}}^{-1} := 4\epsilon^{-1} d \cdot p(t),$$

and N is chosen as the smallest power of two such that $N \geq q(t) \cdot (4d\epsilon^{-1}p(t))^2$ for some sufficiently large polynomial q, so that the empirical estimation of probability achieves an accuracy error of at most $\pm\epsilon/(4d \cdot p(t))$ with negligible confidence error. Let $b_1, \ldots, b_N \in \{0, 1, \varepsilon\}$ denote the resulting independent samples from UE.

For each $b \in \{0, 1\}$, let

$$\tilde{p}_b = \frac{|\{i \in [N] : b_i = b\}|}{N}.$$

By construction, we have $\tilde{p}_0 + \tilde{p}_1 \leq 1$ and $\tilde{p}_b \geq 0$ for each $b \in \{0, 1\}$. Then $\tilde{Q}$ computes (p_0, p_1) satisfying the following conditions:

1. $p_b \geq \frac{\epsilon}{8d \cdot p(t)}$ for each $b \in \{0, 1\}$.
2. Each p_b is a rational number whose denominator is a power of two.
3. $p_0 + p_1 \leq 1$.
4. $|\tilde{p}_b - p_b| \leq \frac{\epsilon}{4d \cdot p(t)}$ for each $b \in \{0, 1\}$.

Such a pair (p_0, p_1) can be obtained by setting for each $b \in \{0, 1\}$,

$$p_b = \begin{cases} \tilde{p}_b - \frac{1}{N'} & \text{if } \tilde{p}_b \geq \frac{\epsilon}{2d \cdot p(t)}, \\ \tilde{p}_b + \frac{1}{N'} & \text{if } \tilde{p}_b \leq \frac{\epsilon}{4d \cdot p(t)}, \\ \tilde{p}_b & \text{otherwise,} \end{cases}$$

where N' is the smallest power of two such that $N' \geq 4d\epsilon^{-1} \cdot p(t)$. The properties above are easily verified from the facts that

$$\frac{\epsilon}{8d \cdot p(t)} < \frac{1}{N'} \leq \frac{\epsilon}{4d \cdot p(t)},$$

and that both N and N' are powers of two.

Finally, $\tilde{Q}$ outputs (p_0, p_1) as its answer.

Now, we verify the properties of $\tilde{Q}$. Since UE halts in

$$\mathsf{poly}(|x|, |z|, t, \epsilon_{\mathsf{UE}}^{-1}, d) = \mathsf{poly}(|x|, |z|, t, \epsilon^{-1}, d),$$

the algorithm $\tilde{Q}$ runs in polynomial time. Moreover, the first property of the lemma trivially follows from Items 1 and 2 by taking $\gamma(t, \epsilon^{-1}, d) = 8\epsilon^{-1}d \cdot p(t)$.

Thus, it remains to verify the second property of the lemma. To this end, assume that $t \geq O(|x|)$, $\mathrm{cd}^t(x \mid z) \leq \log d$, and $\mathrm{cd}^t(x \circ b \mid z) \leq \log d$.

For each $b' \in \{0, 1, \varepsilon\}$, let

$$p^*_{b'} = \Pr_{\mathsf{UE}}\left[b' \leftarrow \mathsf{UE}(x; z, 1^{\langle 1, t, \epsilon_{\mathsf{UE}}^{-1}, d\rangle})\right].$$

Since $t \geq O(|x|)$ and $\mathrm{cd}^t(x \mid z) \leq \log d$, it follows from Lemma 4.8 that

$$\Delta := \Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(x; z, 1^{\langle 1, t, \epsilon_{\mathsf{UE}}^{-1}, d\rangle}\right), \mathsf{Next}_1\left(x; \mathrm{Q}^{t,z}\right)\right) \leq \epsilon_{\mathsf{UE}},$$

and hence

$$|p^*_b - \mathrm{Q}^{t,z}(b\|x)| \leq \sum_{b' \in \{0,1,\varepsilon\}} \left|p^*_{b'} - \mathrm{Q}^{t,z}(b'\|x)\right| = 2\Delta \leq 2\epsilon_{\mathsf{UE}} = \frac{\epsilon}{2d \cdot p(t)}.$$

As long as the empirical estimation $\tilde{p}_b$ of p_b^* succeeds (which holds except with negligible probability), we have

$$|\tilde{p}_b - p_b^*| \le \frac{\epsilon}{4d \cdot p(t)}.$$

Thus, we derive from the triangle inequality that for each $b \in \{0,1\}$,

$$\begin{aligned}\left|p_b - \mathrm{Q}^{t,z}(b\|x)\right| &\le |p_b - \tilde{p}_b| + |\tilde{p}_b - p_b^*| + \left|p_b^* - \mathrm{Q}^{t,z}(b\|x)\right| \\ &\le \frac{\epsilon}{4d \cdot p(t)} + \frac{\epsilon}{4d \cdot p(t)} + \frac{\epsilon}{2d \cdot p(t)} \\ &\le \frac{\epsilon}{d \cdot p(t)}.\end{aligned}$$

In addition, since $t \ge O(|x|)$ and $\mathrm{cd}^t(x \circ b \mid z) \le \log d$, Lemma 8.2 implies

$$\mathrm{Q}^{t,z}(b\|x) \ge \frac{1}{2^{\mathrm{cd}^t(x\circ b|z)} \cdot p(t)} \ge \frac{1}{d \cdot p(t)}.$$

By combining the two inequalities above, we obtain

$$\left|p_b - \mathrm{Q}^{t,z}(b\|x)\right| \;\le\; \frac{\epsilon}{d \cdot p(t)} \;\le\; \epsilon \cdot \mathrm{Q}^{t,z}(b\|x)\,,$$

which implies

$$(1-\epsilon) \cdot \mathrm{Q}^{t,z}(b\|x) \;\le\; p_b \;\le\; (1+\epsilon) \cdot \mathrm{Q}^{t,z}(b\|x)\,,$$

as desired. □

Finally, we introduce notation for the information content (i.e., the code length) under $\tilde{Q}$. For $x, z \in \{0,1\}^*$ and $t, d \in \mathbb{N}$, define

$$\tilde{q}^t_{\epsilon,d}(x \mid z) \;:=\; -\log \prod_{i=1}^{|x|} p_{x_i \,\|\, x_{<i}},$$

where $(p_{0\,\|\,x_{<i}}, p_{1\,\|\,x_{<i}}) \leftarrow \tilde{Q}\Big(x_{<i};\, z, 1^t, 1^{\epsilon^{-1}}, 1^d\Big)$.

8.2 Confidence Boosting via Merge-Segmentation

We now prove Theorem 8.1. As a technical device for our confidence boosting, we first formalize a merge segmenter.

Definition 8.2. *A* merge segmenter *is a randomized Turing machine M that, given a sequence of binary strings $\boldsymbol{x} = (x_1, \ldots, x_m)$ and an advice string $\alpha \in \{0,1\}^*$, outputs a sequence of strings $\boldsymbol{y} = (y_1, \ldots, y_n)$ with $n \le m$ such that, for every internal randomness for M, there exist indices $s_i, t_i \in [m]$ with $s_i \le t_i$ for all $i \in [n]$ and:*

1. *$y_i = \langle x_{s_i}, x_{s_i+1}, \ldots, x_{t_i}\rangle$ for all $i \in [n]$;*
2. *$s_1 = 1$, $t_i + 1 = s_{i+1}$ for all $i < n$, and $t_n = m$, i.e.,*

$$(x_{s_1}, \ldots, x_{t_1},\, x_{s_2}, \ldots, x_{t_2},\, \ldots,\, x_{s_n}, \ldots, x_{t_n}) = \boldsymbol{x}.$$

We use the notation $C(\boldsymbol{x};\alpha)$ to denote the output of a merge segmenter C on input sequence $\boldsymbol{x}$ with advice α.

As an intermediate step, we prove the following lemma via our merge-segmentation argument. It shows that, except on a δ-fraction of positions and pasts, the code length assigned by our next-bit generator $\tilde{Q}$ is within an additive $O(\eta^{-1}\delta^{-1}\ell\log t)$ of the optimal (i.e., K), after an appropriate resegmentation of the past by a merge segmenter.

Lemma 8.3. *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exist a polynomial-time merge segmenter* C*, a polynomial* τ*, and a constant* c *such that for every* $b,s,t,\delta^{-1},\eta^{-1}\in\mathbb{N}$ *and every sequence* $x_1,\ldots,x_b\in\{0,1\}^*$ *generated by an* s*-size,* t*-time randomized program, the following holds: if* $b\geq 4\delta^{-1}\eta^{-1}s/\log t$ *and* $t\delta^{-1}\eta^{-1}\leq 2^s$*, then*

$$\Pr_{i\sim[b],\boldsymbol{x}_{<i}}\left[\Pr_{x_i,C,\tilde{Q}}\left[\exists\alpha\in[\ell]\text{ such that }\tilde{q}^{\tau(t)}_{\epsilon,d}(x_i\mid\boldsymbol{y}^\alpha)\leq \mathrm{K}(x_i\mid\boldsymbol{x}_{<i})+c\eta^{-1}\delta^{-1}\ell\log t\right]\geq 1-\eta\right]\geq 1-\delta,$$

where $\boldsymbol{y}^\alpha=C(\boldsymbol{x}_{<i};1^\alpha,1^t)$*,* $\ell=O(\log(\delta^{-1}\eta^{-1}))$*,* $\epsilon^{-1}=O(t)$*, and* $d=\mathsf{poly}(t)$*.*

Proof. Since $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP, we obtain the next-bit generator $\tilde{Q}$ from Lemma 8.1 and the polynomial-time algorithm CR together with a polynomial τ_1 from Lemma 5.1. Without loss of generality, we assume $s\leq t$; otherwise, the universal Turing machine would not be able to read, and thus would not use, the entire description of the input. Let τ_2 denote the polynomial from Proposition 4.6.

We fix the negligible function $\mathsf{negl}(\cdot)$ appearing in the statements of Lemmas 5.1 and 8.1, and may assume without loss of generality that $\mathsf{negl}(t)\leq 2^{-t^2}$, since these functions arise from the empirical estimation used in the proofs of Lemmas 5.1 and 8.1 (see each proof).

For notational simplicity, let us define, for the polynomial-time merge segmenter C (to be constructed later), for $i,x_{<i},x_i$, and randomness ρ used in executing C and $\tilde{Q}$, the event $E(i,x_{<i},x_i,\rho)$ to be that these variables satisfy the condition in the statement of the lemma; i.e.,

$$\exists\alpha\in[\ell]\text{ such that }\tilde{q}^{\tau(t)}_{\epsilon,d}(x_i\mid C(\boldsymbol{x}_{<i};1^\alpha,1^t))\leq \mathrm{K}(x_i\mid\boldsymbol{x}_{<i})+c\eta^{-1}\delta^{-1}\ell\log t,$$

where $\ell=\log(4\delta^{-1}\eta^{-1})$, $\epsilon^{-1}=2t$, $d=p_d(t)$, and c is a sufficiently large constant, and p_d,τ are sufficiently large polynomials, all to be specified later.

Let $\boldsymbol{x}$ denote $(x_1,\ldots,x_m)$. We construct the merge segmenter C and show that

$$\Pr_x\left[\Pr_{i,\rho}\left[E(i,x_{<i},x_i,\rho)\right]\;\geq\;1-\frac{3}{4}\eta\delta\right]\;\geq\;1-2^{-(s+2)}. \tag{15}$$

Indeed, the lemma follows immediately from Equation (15) as follows. By the union bound,

$$\Pr_{x,i,\rho}\left[E(i,x_{<i},x_i,\rho)\right]\;\geq\;1-\tfrac{3}{4}\eta\delta-2^{-(s+2)}\;\geq\;1-\eta\delta,$$

since $2^{-s}\leq\eta\delta/t\leq\eta\delta$. Thus, by Markov's inequality we obtain the lemma:

$$\Pr_{i,\boldsymbol{x}_{<i}}\left[\Pr_{x_i,\rho}[E(i,x_{<i},x_i,\rho)]\;\geq\;1-\eta\right]\;\geq\;1-\delta.$$

Therefore, it suffices to prove Equation (15), whose proof is given below.

From Proposition 4.11, for any sufficiently large polynomial p_1,

$$\Pr_x\left[\mathrm{cd}^{p_1(t)}(x)\leq O(s+\log t)\right]\;\geq\;1-2^{-(s+2)}.$$

Since we assume $t\delta^{-1}\eta^{-1} \le 2^s$, it follows that $\log t \le s$. Hence,

$$\Pr_x\left[\mathrm{cd}^{p_1(t)}(x) \le c_1 s\right] \;\ge\; 1 - 2^{-(s+2)},$$

by choosing c_1 to be a sufficiently large universal constant.

Thus, to prove Equation (15), it suffices to show that for every x with $\mathrm{cd}^{p_1(t)}(x) \le c_1 s$,

$$\Pr_{i,\rho}[E(i, x_{<i}, x_i, \rho)] \;\ge\; 1 - \frac{3}{4}\eta\delta. \tag{16}$$

Fix such an $\boldsymbol{x}$ arbitrarily.

Now, we consider an arbitrary sequence $\boldsymbol{y} = (y_1, \ldots, y_n)$ obtained by merging consecutive strings from $\boldsymbol{x}$ into some blocks; i.e., $n \le m$, and there exist indices $s_i, t_i \in [m]$ with $s_i \le t_i$ for all $i \in [n]$ such that

1. $y_i = \langle x_{s_i}, x_{s_i+1}, \ldots, x_{t_i}\rangle$ for all $i \in [n]$;
2. $s_1 = 1$, $t_i + 1 = s_{i+1}$ for all $i < n$, and $t_n = m$, i.e.,

$$(x_{s_1}, \ldots, x_{t_1},\, x_{s_2}, \ldots, x_{t_2},\, \ldots,\, x_{s_n}, \ldots, x_{t_n}) = \boldsymbol{x}.$$

These are the properties of merge segmenters. We call such a sequence $\boldsymbol{y}$ a *resegmentation* of $\boldsymbol{x}$.

Notice that any such resegmentation $\boldsymbol{y}$ of $\boldsymbol{x}$ is polynomial-time computable from $\boldsymbol{x}$ and the auxiliary information $Y := (|y_1|, \ldots, |y_n|)$. The latter can be described using $n \cdot O(\log t)$ bits, since the total length of the strings in $\boldsymbol{x}$ (and thus in $\boldsymbol{y}$) is at most $O(t)$, as it is generated within time t. Namely, by the slow growth law (Lemma 4.6), there exists a universal polynomial p_2 such that for every $n \in \mathbb{N}$ and every $y = (y_1, \ldots, y_n)$ satisfying the above,

$$\begin{aligned}
\mathrm{cd}^{p_2(t)}(\boldsymbol{y}) &\le \mathrm{cd}^{4p_1(t)}(\boldsymbol{x}, Y) + O(\log t) \\
&= \mathrm{q}^{4p_1(t)}(\boldsymbol{x}, Y) - \mathrm{K}(\boldsymbol{x}, Y) + O(\log t) \\
&\le \mathrm{q}^{p_1(t)}(\boldsymbol{x}) + \mathrm{q}^{p_1(t)}(Y \mid \boldsymbol{x}) - \mathrm{K}(\boldsymbol{x}) + O(\log t) \\
&\le \mathrm{q}^{p_1(t)}(\boldsymbol{x}) + |Y| - \mathrm{K}(\boldsymbol{x}) + O(\log t) \\
&\le \mathrm{cd}^{p_1(t)}(\boldsymbol{x}) + n \cdot O(\log t) \\
&\le c_2 \cdot (s + n \log t),
\end{aligned} \tag{17}$$

where c_2 is a sufficiently large constant.

Now we apply the adaptive merge-segmentation argument. Define a verifier V as follows. For a sequence $(y_1, \ldots, y_n)$ and $i \in [n]$, the verifier V decides whether to accept y_i using only $\boldsymbol{y}_{<i}$ and y_i, as follows: (i) it computes $\tilde{q}^{\tau(t)}_{\epsilon,d}(y_i \mid \boldsymbol{y}_{<i})$ by executing $\tilde{Q}$ given advice $(\boldsymbol{y}_{<i}, 1^{\tau(t)}, 1^{\epsilon^{-1}}, 1^d)$; (ii) it computes k_i by executing $\mathsf{CR}(\boldsymbol{y}_{\le i}, 1^{t'})$, where $t' = \tau_1^{-1}(\tau(t))$; and (iii) it accepts if and only if

$$\tilde{q}^{\tau(t)}_{\epsilon,d}(y_i \mid \boldsymbol{y}_{<i}) \le k_i + 1.$$

The merge-segmentation argument is outlined as follows. As initialization, let $\boldsymbol{x}^1 = \boldsymbol{x}$. At round $\alpha \ge 1$, we apply V to strings in $\boldsymbol{x}^\alpha$ in an online manner and obtain $\boldsymbol{x}^{\alpha+1}$ by merging all consecutive accepted strings, thereby decreasing the number of strings. We will show that at each step V accepts a large constant fraction (e.g., 0.8) of the blocks. This implies that at each step the number of blocks decreases by a factor of two, while the fraction of unsatisfied strings within the

new blocks increases. Consequently, after $\ell = O(\log(\delta^{-1}\eta^{-1}))$ rounds, all but an $O(\delta\eta)$ fraction of strings in $\boldsymbol{x}$ are accepted at some round. Intuitively, C receives $\alpha \in [\ell]$, which specifies the round at which x_i is first accepted, and then outputs a segmentation $\boldsymbol{y}^\alpha$, consisting of the sequence in $\boldsymbol{x}^\alpha$ preceding x_i, obtained by applying the above merging process for α rounds.

We now formalize the idea outlined above. First, we observe that V accepts a large fraction of the strings in any resegmentation of $\boldsymbol{x}$, provided that the number of strings is sufficiently large, as stated below:

Claim 8.1. *For any resegmentation $\boldsymbol{y} = (y_1, \ldots, y_n)$ of $\boldsymbol{x}$, if $n \geq s/\log t$, then*

$$\Pr_{i\sim[n]}\left[\Pr_{V}\left[V(\boldsymbol{y}_{<i}, y_i) = \mathsf{accept}\right] \geq 1 - 2^{-t^{1.9}}\right] \geq 0.8.$$

Proof. We apply Lemma 5.3 with parameter $t := p_2(t)$. Combining this with Equation (17), there exist a universal polynomial p_3 and a universal constant c_3 such that

$$\Pr_{i\sim[n]}\left[\mathrm{cd}^{p_3(t)}(y_i \mid \boldsymbol{y}_{<i}) \leq c_3 \cdot \log t\right] \geq 0.8. \tag{18}$$

because

$$\begin{aligned}(0.8)^{-1} \cdot \left(\frac{\mathrm{cd}^{p_2(t)}(\boldsymbol{y})}{n} + O(\log t)\right) &\leq (0.8)^{-1} \cdot \left(\frac{c_2(s + n\log t)}{n} + O(\log t)\right)\\ &\leq O\Big(\frac{s}{n}\Big) + O(\log t)\\ &\leq O(\log t),\end{aligned}$$

where the last inequality follows from $s/n \leq \log t$.

From Lemma 4.7, there exist universal polynomials τ and p_d such that

$$\Pr_{i\sim[n]}\left[\forall j \in [|y_i|]\, \mathrm{cd}^{\tau(t)}((y_i)_{[j]} \mid \boldsymbol{y}_{<i}) \leq \log p_d(t)\right] \geq 0.8,$$

where recall that $(y_i)_{[j]}$ denotes the first j bits of y_i. Here, from Proposition 4.6, we can choose τ sufficiently large so that $t' = \tau_1^{-1}(\tau(t)) \geq \tau_2(p_2(t))$ for each t (i.e., $\tau(t) \geq \tau_1(\tau_2(p_3(t)))$), which specifies the aforementioned τ and p_d.

Thus, it suffices to show that the claimed event occurs for any such index $i \in [n]$.

Since any prefix of y_i satisfies the condition in the second item of Lemma 8.1, the union bound implies that with probability at least $1 - |y_i| \cdot 2^{-t^2} \geq 1 - t \cdot 2^{-t^2}$ over the randomness of $\tilde{Q}$, for all $j \in [|y_i|]$,

$$p^i_{(y_i)_j \| (y_i)_{[j-1]}} \geq (1-\epsilon) \cdot \mathrm{Q}^{\tau(t), \boldsymbol{y}_{<i}}((y_i)_j \mid (y_i)_{[j-1]}),$$

where $(y_i)_j$ is the j-th bit of y_i, and

$$(p^i_{0\|(y_i)_{[j-1]}}, p^i_{1\|(y_i)_{[j-1]}}) \leftarrow \tilde{Q}((y_i)_{[j-1]}; \boldsymbol{y}_{<i}, 1^{\tau(t)}, 1^{\epsilon^{-1}}, 1^d).$$

Under this event, we have

$$\begin{aligned}\tilde{q}_{\epsilon,d}^{\tau(t)}(y_i \mid \boldsymbol{y}_{<i}) &= -\log \prod_{j=1}^{|y_i|} p^i_{(y_i)_j|(y_i)_{[j-1]}} \\ &\le -\log \prod_{j=1}^{|y_i|} \mathrm{Q}^{\tau(t),\boldsymbol{y}_{<i}}((y_i)_j \mid (y_i)_{[j-1]}) + |y_i| \cdot \log(1-\epsilon)^{-1} \\ &\le \mathrm{q}^{\tau(t)}(y_i \mid \boldsymbol{y}_{<i}) + t \cdot \log(1-(2t)^{-1})^{-1} \\ &\le \mathrm{q}^{\tau(t)}(y_i \mid \boldsymbol{y}_{<i}) + 1,\end{aligned}$$

where the last inequality uses $\log(1-(2t)^{-1})^{-1} \le t^{-1}$ for all $t \ge 1$.

In addition, from Lemma 5.1, with probability at least $1-2^{-t^2}$ over the randomness of CR, $k_i \leftarrow \mathsf{CR}(\boldsymbol{y}_{\le i}, 1^{t'})$ satisfies

$$k_i \ge \mathrm{q}^{\tau_1(t')}(y_i \mid \boldsymbol{y}_{<i}) = \mathrm{q}^{\tau(t)}(y_i \mid \boldsymbol{y}_{<i}).$$

If both events occur, then

$$\tilde{q}_{\epsilon,d}^{\tau(t)}(y_i \mid \boldsymbol{y}_{<i}) \le \mathrm{q}^{\tau(t)}(y_i \mid \boldsymbol{y}_{<i}) + 1 \le k_i + 1,$$

and thus V accepts. By the union bound, this happens with probability at least $1-(t+1)\cdot 2^{-t^2} \ge 1-2^{-t^{1.9}}$ for sufficiently large t. ◇

Now we consider the following adaptive procedure with ℓ rounds. Let $\boldsymbol{x}^1 := \boldsymbol{x}$. In the α-th round, given the current sequence $\boldsymbol{x}^\alpha = (x_1^\alpha, \dots, x_{m_\alpha}^\alpha)$ with $m_\alpha \le m$, we execute $V(\boldsymbol{x}^\alpha_{<I}, x_I^\alpha)$ for each $I \in [m_\alpha]$. If V accepts, we call such an x_I^α an *accepted block*. We then construct a new sequence $\boldsymbol{x}^{\alpha+1} = (x_1^{\alpha+1}, \dots, x_{m_{\alpha+1}}^{\alpha+1})$ where $m_{\alpha+1} \le m_\alpha$, obtained by merging each maximal sequence of consecutive accepted blocks into a single block. Namely, $\boldsymbol{x}^{\alpha+1}$ is a resegmentation of $\boldsymbol{x}$, and each string in $\boldsymbol{x}^{\alpha+1}$ is either

(i) $\flat\langle x_k^\alpha, x_{k+1}^\alpha, \dots, x_{k'}^\alpha\rangle$ for some $k < k'$, where $x_k^\alpha, \dots, x_{k'}^\alpha$ are all accepted but x_{k-1}^α and $x_{k'+1}^\alpha$ are not accepted (if they exist), or

(ii) a singleton corresponding to a non-accepted block.

We recall that $\flat$ denotes the flattening operator.

First, we assume that the negligible error event with probability $2^{-t^{1.9}}$ from the randomness in executing V in Claim 8.1 does not occur.

We observe that as long as $m_i \ge s/\log t$, the length of the sequence is reduced by at least half, i.e., $m_{i+1} \le m_i/2$. Indeed, if $m_{i+1} > m_i/2$, then there must be at least $m_i/4$ non-accepted blocks (otherwise, more than half of the blocks in $\boldsymbol{x}^{i+1}$ would consist of accepted ones, which implies that some consecutive accepted blocks were not merged). This contradicts Claim 8.1, which states that at most $0.2m_i (< m_i/4)$ non-accepted blocks can exist.

Namely, after $\ell = \log(4\delta^{-1}\eta^{-1})$ rounds, the length of the sequence satisfies

$$m_\ell \le \max\{2^{-\ell}\cdot m,\ s/\log t\} \le \tfrac{\delta\eta}{4}\cdot m.$$

Thus, for a $1-\delta\eta/4$ fraction of indices $i \in [m]$, there exists a round $\alpha_i \in [\ell]$ at which V accepts x_i for the first time (i.e., x_i appears as a singleton in $\boldsymbol{y}^{\alpha_i}$) and $m_{\alpha_i} \ge s/\log t$. We call such α_i the *critical round* for the i-th string x_i. For the remaining indices i, we define $\alpha_i := \perp$ for convenience

(note that even if x_i is accepted for the first time, we regard α_i as $\perp$ whenever $m_{\alpha_i} < s/\log t$). With this terminology,

$$\Pr_{i\sim[m]}[\alpha_i \neq \perp] \geq 1 - \frac{\eta\delta}{4}. \tag{19}$$

Next we demonstrate that for most i, at its critical round, the corresponding output from CR well approximates the time-unbounded conditional Kolmogorov complexity within additive error $O(\eta^{-1}\delta^{-1}\ell\log t)$.

For each round $\alpha \in [\ell]$, consider the sequence $\boldsymbol{x}^\alpha$ at this round. Let $k_1^\alpha, \ldots, k_{m_\alpha}^\alpha$ denote the outputs obtained by sequentially executing $\mathsf{CR}(\text{-}; 1^{t'})$ on $\boldsymbol{x}^\alpha$. Since $\boldsymbol{x}^\alpha$ is a resegmentation of $\boldsymbol{x}$, from Equation (17) we have

$$\mathrm{cd}^{p_2(t)}(\boldsymbol{x}^\alpha) \leq c_2 \cdot (s + m_\alpha \log t).$$

Suppose that $m_\alpha \geq s/\log t$. Then, from Lemma 5.1, except with negligible error probability 2^{-t^2} in CR, it holds that

$$\Pr_{i_\alpha\sim[m_\alpha]}\left[k_{i_\alpha} \leq \mathrm{K}(x_{i_\alpha}^\alpha \mid \boldsymbol{x}_{<i_\alpha}^\alpha) + c_4\eta^{-1}\delta^{-1}\ell\log t\right] \geq 1 - \frac{\eta\delta}{4\ell}, \tag{20}$$

for some universal constant c_4, because

$$\begin{aligned}
4\ell\eta^{-1}\delta^{-1} \cdot \left(\frac{\mathrm{cd}^{t'}(\boldsymbol{x}^\alpha)}{m_\alpha} + O(\log t)\right) &\leq O(\ell\eta^{-1}\delta^{-1}) \cdot \left(\frac{\mathrm{cd}^{p_2(t)}(\boldsymbol{x}^\alpha) + O(\log t)}{m_\alpha} + O(\log t)\right) \\
&\leq O(\ell\eta^{-1}\delta^{-1}) \cdot \left(\frac{c_2 \cdot (s + m_\alpha \log t)}{m_\alpha} + O(\log t)\right) \\
&\leq O(\ell\eta^{-1}\delta^{-1}) \cdot \left(\frac{s}{m_\alpha} + \log t\right) \\
&\leq O(\ell\eta^{-1}\delta^{-1}\log t),
\end{aligned}$$

where the first inequality follows from Proposition 4.6, since $t' \geq \tau_2(p_2(t))$, and the last inequality follows from $m_\alpha \geq s/\log t$. Namely, for each α with $m_\alpha \geq s/\log t$, there are at most $(\eta\delta/4\ell)\cdot m_\alpha \leq (\eta\delta/4\ell)\cdot m$ indices that do not satisfy the event in Equation (20). Thus, except with negligible error probability $\ell \cdot 2^{-t^2}$ in CR, there are at most $(\eta\delta/4)\cdot m$ pairs $(\alpha, i_\alpha) \in [\ell] \times [m_\alpha]$ satisfying

$$m_\alpha \geq s/\log t \ \wedge\ k_{i_\alpha}^\alpha > \mathrm{K}(x_{i_\alpha}^\alpha \mid \boldsymbol{x}_{<i_\alpha}^\alpha) + c_4\eta^{-1}\delta^{-1}\ell\log t. \tag{21}$$

For the moment, we assume that this negligible error in executing CR does not occur.

Consider any $i \in [m]$ with $\alpha_i \neq \perp$. Since x_i is accepted for the first time at round α_i, there is a unique pair $(\alpha_i, \mathsf{cidx}(i)) \in [\ell] \times [m_{\alpha_i}]$, where $\mathsf{cidx}(i)$ denotes the index of x_i in $\boldsymbol{x}^{\alpha_i}$, such that $x_i = x_{\mathsf{cidx}(i)}^{\alpha_i}$. Thus, if

$$k_{\mathsf{cidx}(i)}^{\alpha_i} > \mathrm{K}(x_i \mid \boldsymbol{x}_{<\mathsf{cidx}(i)}^{\alpha_i}) + c_4\eta^{-1}\delta^{-1}\ell\log t$$

holds, then $(\alpha_i, \mathsf{cidx}(i))$ belongs to the set of pairs satisfying Equation (21). Since there are at most $(\eta\delta/4)\cdot m$ such pairs, we have

$$\Pr_{i\sim[m]}\left[\alpha_i \neq \perp \Longrightarrow k_{\mathsf{cidx}(i)}^{\alpha_i} \leq \mathrm{K}(x_i \mid \boldsymbol{x}_{<\mathsf{cidx}(i)}^{\alpha_i}) + c_4\eta^{-1}\delta^{-1}\ell\log t\right] \geq 1 - \frac{(\eta\delta/4)\cdot m}{m} = 1 - \frac{\eta\delta}{4}.$$

From the union bound applied together with Equation (19), it follows that

$$\Pr_{i\sim[m]}\left[\alpha_i \neq \perp \ \wedge\ k_{\mathsf{cidx}(i)}^{\alpha_i} \leq \mathrm{K}(x_i \mid \boldsymbol{x}_{<\mathsf{cidx}(i)}^{\alpha_i}) + c_4\eta^{-1}\delta^{-1}\ell\log t\right] \geq 1 - \left(\frac{\eta\delta}{4} + \frac{\eta\delta}{4}\right) = 1 - \frac{\eta\delta}{2}. \tag{22}$$

For such i, since x_i $(= x^{\alpha_i}_{\mathsf{cidx}(i)})$ is accepted by V, we have, for a sufficiently large universal constant c,

$$\begin{aligned}\tilde{q}^{\tau(t)}_{\epsilon,d}(x_i \mid \boldsymbol{x}^{\alpha_i}_{<\mathsf{cidx}(i)}) &\le k^{\alpha_i}_{\mathsf{cidx}(i)} + 1\\ &\le \mathrm{K}(x_i \mid \boldsymbol{x}^{\alpha_i}_{<\mathsf{cidx}(i)}) + c_4\eta^{-1}\delta^{-1}\ell\log t + 1\\ &\le \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + O(1) + O(\eta^{-1}\delta^{-1}\ell\log t)\\ &\le \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + c\eta^{-1}\delta^{-1}\ell\log t,\end{aligned}$$

where the third inequality holds because $\boldsymbol{x}_{<i}$ can be obtained from $\boldsymbol{x}^{\alpha_i}_{<\mathsf{cidx}(i)}$ by flattening.

Now, we specify the merge segmenter C. Given a sequence $\boldsymbol{x}_{<i}$, advice $\alpha \in [\ell]$, and $t \in \mathbb{N}$ (in unary), C follows the merge-segmentation procedure described above for $\alpha - 1$ rounds, with the difference that C does not perform verification or merging of $\boldsymbol{x}_{\ge i} := (x_i, \dots, x_m)$. Notice that as long as x_i is not accepted in any of the rounds from 1 to $\alpha-1$, $\boldsymbol{x}_{\ge i}$ does not affect the resegmentation of $\boldsymbol{x}_{<i}$, since x_i is never concatenated. Namely, C exactly simulates the above procedure up to the round immediately before x_i is accepted. Thus, if α_i is the first round at which x_i is accepted by V, the merge segmenter C outputs $\boldsymbol{x}^{\alpha}_{<\mathsf{cidx}(i)}$, where $\mathsf{cidx}(i) \in [m_\alpha]$ denotes the index of x_i in the resegmentation $\boldsymbol{x}^\alpha$ (provided that the same random tape is used for each execution of V, which can be ensured by taking a sufficiently long random tape and fixing which portion is used depending on the round and the index in the current segment).

Therefore, if $i \in [m]$ satisfies the event in Equation (22), then for $\alpha := \alpha_i$

$$\tilde{q}^{\tau(t)}_{\epsilon,d}(x_i \mid \boldsymbol{y}^\alpha) \le \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + c\eta^{-1}\delta^{-1}\ell\log t.$$

for $\boldsymbol{y}^\alpha = C(\boldsymbol{x}_{<i}; 1^\alpha, 1^t)$.

Since each execution of $\tilde{Q}(\text{-}; 1^{\tau(t)}, 1^{\epsilon^{-1}}, 1^d)$ and $\mathsf{CR}(\text{-}; 1^{t'})$ halts in $\mathsf{poly}(t)$ time, the total running time of C is bounded by $\alpha \cdot \mathsf{poly}(t)$.

Taking into account the error probability in C (arising from $\tilde{Q}$ and CR), the union bound yields the following bound in Equation (16):

$$\begin{aligned}\Pr_{i,\rho}[E(i, x_{<i}, x_i, \rho)] &= \Pr_{i\sim[m],\tilde{Q},C}\left[\exists\alpha \in [\ell] \text{ such that } \tilde{q}^{\tau(t)}_{\epsilon,d}(x_i \mid \boldsymbol{y}^\alpha) \le \mathrm{K}(x_i \mid \boldsymbol{x}_{<i}) + c\eta^{-1}\delta^{-1}\ell\log t\right]\\ &\ge 1 - \frac{\eta\delta}{2} - 2\ell\cdot 2^{-t^{1.9}}\\ &\ge 1 - \frac{3}{4}\eta\delta,\end{aligned}$$

where the last inequality holds because

$$2\ell\cdot 2^{-t^{1.9}} \le O(\log(\delta^{-1}\eta^{-1}))\cdot 2^{-t^{1.9}} \le O(s)\cdot 2^{-t^{1.9}} \le 2^{-t} \le 2^{-s} \le \delta\eta/t \le \delta\eta/4$$

for any sufficiently large t. This completes the proof of Equation (15), and hence of the lemma. □

Intuitively, Lemma 8.3 establishes algorithmic near-optimality for almost all positions with high probability over the data, which in turn yields statistical optimality (e.g., a KL-divergence bound). We now formalize this to complete the proof of Theorem 8.1.

Proof of Theorem 8.1. Since we assume that $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$, we obtain the next-bit generator $\tilde{Q}$ from Lemma 8.1, and the merge segmenter C together with the polynomial τ from Lemma 8.3. For each parameter $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, we define

$$\eta^{-1} = c_\eta\delta^{-1}\epsilon^{-2}n\log t,$$

where c_η is a sufficiently large constant to be specified later. Applying Lemma 8.3 with δ^{-1} and η^{-1} yields

$$\ell = O(\log(\delta^{-1}\eta^{-1})).$$

Next, we define

$$w = \left\lceil c_w \delta^{-2}\epsilon^{-2}\eta^{-1}\ell \log t \right\rceil,$$

where c_w is a sufficiently large constant to be specified later. Let c_1 denote the universal constant from Lemma 8.3; we assume without loss of generality that c_1 is large enough so that $\ell \le c_1 \log(\delta^{-1}\eta^{-1})$. We then set

$$b_0 = c_1 \delta^{-1}\eta^{-1}\frac{s}{\ell \log t}.$$

Finally, define b as the largest integer $b \in \mathbb{N}$ such that $bw \le m$.

Throughout, we assume that $m \ge wb_0$; thus, $b \ge b_0$. Since

$$wb_0 = O\Big(\delta^{-2}\epsilon^{-2}\eta^{-1}\ell \log t \cdot \delta^{-1}\eta^{-1}\tfrac{s}{\ell \log t}\Big) = s \cdot O(\delta^{-3}\epsilon^{-2}\eta^{-2}) = s \cdot \tilde{O}(n^2\epsilon^{-6}\delta^{-5}),$$

this assumption can be expressed as

$$m \ge s \cdot p(n, \epsilon^{-1}, \delta^{-1}),$$

for some universal polynomial $p(n, \epsilon^{-1}, \delta^{-1}) = \tilde{O}(n^2\epsilon^{-6}\delta^{-5})$, as stated in the theorem.

Under these parameters, we observe that a random selection from $[m]$ rarely falls outside $[bw]$, as stated below:

Claim 8.2. *If $2^s \ge \delta^{-1}\eta^{-1}$, then with probability at least $1-\delta$ over $i \sim [m]$, we have $i \le bw$.*

We defer the proof for now. Based on this fact, we mainly focus on the case in which the prediction stage i lies within $[bw]$ in what follows.

For a sequence of binary strings $x_1, \dots, x_m \in \{0,1\}^n$, we use the notation $x^i_j \in \{0,1\}^n$ to denote the $((i-1)w+j)$-th string, and let $\bar{x}^i = x^i_1 \circ \cdots \circ x^i_w$, that is, x^i_j is the j-th string in the i-th block $\bar{x}^i$. Note that selecting an index uniformly at random from $[m]$, conditioned on it being at most $w \cdot b$, is equivalent to selecting a pair $(i,j) \in [b] \times [w]$ uniformly at random.

We consider the case in which $t\delta^{-1}\eta^{-1} \le 2^s$ is satisfied. From Lemma 8.3, with probability at least $1-\delta$ over i and $\boldsymbol{x}^{<i} = (\bar{x}^1, \dots, \bar{x}^{i-1})$, it holds that

$$\Pr_{\bar{x}^i, C, \tilde{Q}}\Big[\exists \alpha \in [\ell] \text{ such that } \tilde{q}^{\tau(t)}_{O(t),\mathsf{poly}(t)}(\bar{x}^i \mid \boldsymbol{y}^\alpha) \le \mathrm{K}(\bar{x}^i \mid \boldsymbol{x}^{<i}) + c_1\eta^{-1}\delta^{-1}\ell \log t\Big] \ge 1-\eta, \tag{23}$$

where $\boldsymbol{y}^\alpha = C(\boldsymbol{x}^{<i}; 1^\alpha, 1^t)$. Below we fix such a pair $(i, \boldsymbol{x}^{<i})$ arbitrarily.

For each $\alpha \in [\ell]$, let D^i_α be the distribution over $\{0,1\}^{\le wn}$ defined as follows: for each $x \in \{0,1\}^{<wn}$ and $y \in \{0,1\}$,

$$D^i_\alpha(y \,\|\, x) = p^\alpha_{y \,\|\, x},$$

where $(p^\alpha_{0\,\|\,x}, p^\alpha_{1\,\|\,x}) \leftarrow \tilde{Q}(x; C(\boldsymbol{x}^{<i}; 1^\alpha, 1^t), 1^{\tau(t)}, 1^{O(t)}, 1^{\mathsf{poly}(t)})$. Furthermore, we define the distribution $\bar{D}$ over $\{0,1\}^{\le wn}$ by

$$\bar{D}^i(x) = \mathop{\mathbb{E}}_{\alpha \sim [\ell]}\big[D^i_\alpha(x)\big] \quad \text{for each } x \in \{0,1\}^{\le wn}.$$

First, we show that

$$\mathop{\mathbb{E}}_{j \sim [w], \bar{x}^i_{<j}, x^i_j, C, \tilde{Q}}\Big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) + \log \mathrm{Q}^{\tau'(t)}\Big(x^i_j \,\Big\|\, \bar{x}^{\le i}_{<j}\Big)\Big] \;\le\; \epsilon^2\delta. \tag{24}$$

where $\bar{x}^{<i} = \bar{x}^1 \circ \cdots \circ \bar{x}^{i-1}$, $\bar{x}^i_{<j} = x^i_1 \circ \cdots \circ x^i_{j-1}$, $\bar{x}^{\le i}_{<j} = \bar{x}^{<i} \circ \bar{x}^i_{<j}$, and τ' is a sufficiently large polynomial to be specified later.

From the definition of $\bar{D}^i$, for each $\alpha \in [\ell]$ and each x, we have

$$-\log \bar{D}^i(x) \le -\log D^i_\alpha(x) + \log \ell;$$

thus,

$$-\log \bar{D}^i(x) \le \min_{\alpha\in[\ell]} -\log D^i_\alpha(x) + \log \ell.$$

Let E be the event in Equation (23) over $\bar{x}^i, C, \tilde{Q}$. Then,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}[-\log \bar{D}^i(\bar{x}^i) \mid E] &\le \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}\left[\min_{\alpha\in[\ell]} -\log D^i_\alpha(\bar{x}^i) \,\middle|\, E\right] + \log \ell \\
&\le \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}\left[\min_{\alpha\in[\ell]} \tilde{q}^{\tau}_{O(t),\mathsf{poly}(t)}(\bar{x}^i \mid C(\bar{x}^{<i}; 1^\alpha, 1^t)) \,\middle|\, E\right] + \log \ell \\
&\le \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}[\mathrm{K}(\bar{x}^i \mid \boldsymbol{x}^{<i}) \mid E] + c_1\eta^{-1}\delta^{-1}\ell\log t + \log \ell \\
&\le \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}[\mathrm{K}(\bar{x}^i \mid \bar{x}^{<i}) \mid E] + c_1\eta^{-1}\delta^{-1}\ell\log t + \log \ell + O(1) \\
&\le \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}[-\log \mathrm{Q}^{\tau'(t)}(\bar{x}^i \,\|\, \bar{x}^{<i}) \mid E] + c_1\eta^{-1}\delta^{-1}\ell\log t + O(\log \ell t),
\end{aligned}$$

where the third inequality follows from the definition of E, the fourth inequality holds since $\bar{x}^{<i}$ $(= \bar{x}^1 \circ \cdots \circ \bar{x}^{i-1})$ is computable from $\boldsymbol{x}^{<i}$ $(= (\bar{x}^1, \ldots, \bar{x}^{i-1}))$ by concatenation, and the last inequality follows from Proposition 4.2.

By rearranging the above, we obtain

$$\begin{aligned}
&\mathop{\mathbb{E}}_{j,\bar{x}^i,C,\tilde{Q}}\left[-\log \bar{D}^i(x^i_j \,\|\, \bar{x}^i_{<j}) + \log \mathrm{Q}^{\tau'(t)}(x^i_j \,\|\, \bar{x}^{\le i}_{<j}) \,\middle|\, E\right] \\
&\quad= \frac{1}{w}\sum_{j=1}^{w} \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}\left[-\log \bar{D}^i(x^i_j \,\|\, \bar{x}^i_{<j}) + \log \mathrm{Q}^{\tau'(t)}(x^i_j \,\|\, \bar{x}^{\le i}_{<j}) \,\middle|\, E\right] \\
&\quad= \frac{1}{w} \mathop{\mathbb{E}}_{\bar{x}^i,C,\tilde{Q}}\left[-\log \bar{D}^i(\bar{x}^i) + \log \mathrm{Q}^{\tau'(t)}(\bar{x}^i \,\|\, \bar{x}^{<i}) \,\middle|\, E\right] \\
&\quad\le \frac{1}{w}\left(c_1\eta^{-1}\delta^{-1}\ell\log t + O(\log \ell t)\right) \\
&\quad\le \delta\epsilon^2 \cdot \frac{c_1\eta^{-1}\delta^{-1}\ell\log t + O(\log \ell t)}{c_w\eta^{-1}\delta^{-1}\ell\log t}.
\end{aligned}$$

Recall that $\eta^{-1} \ge \delta^{-1}\epsilon^{-1}n\log t$. Thus, by choosing c_w sufficiently large with respect to c_1 and the universal constant hidden in $O(\log \ell t)$, we obtain

$$\mathop{\mathbb{E}}_{j,\bar{x}^i_{<j},x^i_j,C,\tilde{Q}}\left[-\log \bar{D}^i(x^i_j \,\|\, \bar{x}^i_{<j}) + \log \mathrm{Q}^{\tau'(t)}(x^i_j \,\|\, \bar{x}^{\le i}_{<j}) \,\middle|\, E\right] \le \frac{\epsilon^2\delta}{2}.$$

We also consider the case where E does not occur. From Lemma 8.1, there exists a universal polynomial γ such that for every $x, y \in \{0,1\}^*$ and every $\alpha \in [\ell]$,

$$-\log D^i_\alpha(y \,\|\, x) \;\le\; -\sum_{i=1}^{|y|} \log p^\alpha_{y_i \,\|\, x\circ y_{<i}} \;\le\; |y| \cdot \log \gamma(t).$$

We can easily verify that

$$\bar{D}^i(y \,\|\, x) = \sum_{\alpha \in [\ell]} \frac{p_\alpha(x)}{\sum_{\alpha' \in [\ell]} p_{\alpha'}(x)} D^i_\alpha(y \,\|\, x), \tag{25}$$

where $p_\alpha(x)$ is the probability that a prefix of a sample from D^i_α corresponds to x. Thus, we obtain

$$\begin{aligned}
-\log \bar{D}^i(y \,\|\, x) &= -\log \left(\sum_{\alpha \in [\ell]} \frac{p_\alpha(x)}{\sum_{\alpha' \in [\ell]} p_{\alpha'}(x)} D^i_\alpha(y \,\|\, x) \right) \\
&\le \sum_{\alpha \in [\ell]} \frac{p_\alpha(x)}{\sum_{\alpha' \in [\ell]} p_{\alpha'}(x)} \big(-\log D^i_\alpha(y \,\|\, x) \big) \\
&\le |y| \cdot \log \gamma(t),
\end{aligned}$$

where the first inequality follows from Jensen's inequality.

We use this evaluation when E does not occur as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{j \sim [w],\, x^i_{<j},\, C,\, \tilde{Q}} \Big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) + \log \mathrm{Q}^{\tau'(t)}\Big(x^i_j \,\Big\|\, \bar{x}^{\le i}_{<j}\Big) \Big] \\
&\le \Pr[E] \cdot \mathop{\mathbb{E}}_{j \sim [w],\, x^i_{<j},\, C,\, \tilde{Q}} \Big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) + \log \mathrm{Q}^{\tau'(t)}\Big(x^i_j \,\Big\|\, \bar{x}^{\le i}_{<j}\Big) \,\Big|\, E \Big] \\
&\quad + \Pr[E^c] \cdot \mathop{\mathbb{E}}_{j \sim [w],\, x^i_{<j},\, C,\, \tilde{Q}} \Big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) + \log \mathrm{Q}^{\tau'(t)}\Big(x^i_j \,\Big\|\, \bar{x}^{\le i}_{<j}\Big) \,\Big|\, E^c \Big] \\
&\le 1 \cdot \frac{\epsilon^2 \delta}{2} \;+\; \Pr[E^c] \cdot \mathop{\mathbb{E}}_{j \sim [w],\, x^i_{<j},\, C,\, \tilde{Q}} \big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) \,\big|\, E^c \big] \\
&\le \frac{\epsilon^2 \delta}{2} \;+\; \eta \cdot n \log \gamma(t) \\
&= \frac{\epsilon^2 \delta}{2} \;+\; \epsilon^2 \delta \cdot \frac{n \log \gamma(t)}{c_\eta n \log t}.
\end{aligned}$$

Thus, by taking c_η sufficiently large with respect to the exponent of γ, we obtain Equation (24):

$$\mathop{\mathbb{E}}_{j \sim [w],\, \bar{x}^i_{<j},\, x^i_j,\, C,\, \tilde{Q}} \Big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) + \log \mathrm{Q}^{\tau'(t)}\Big(x^i_j \,\Big\|\, \bar{x}^{\le i}_{<j}\Big) \Big] \le \frac{\epsilon^2 \delta}{2} + \frac{\epsilon^2 \delta}{2} = \epsilon^2 \delta.$$

Let $X^i_j \mid \boldsymbol{x}^{\le i}_{<j}$ denote the conditional distribution of x^i_j given

$$\boldsymbol{x}^{\le i}_{<j} = (x^1_1, \ldots, x^1_w, x^2_1, \ldots, x^2_w, \ldots, x^i_1, \ldots, x^i_{j-1}) \; \big(= \boldsymbol{x}_{<((i-1)w+j)}\big).$$

Then we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{j,\, \boldsymbol{x}^{\le i}_{<j},\, C,\, \tilde{Q}} \Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\le i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^i_{<j}; \bar{D}^i)\big) - \mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\le i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\le i}_{<j}; \mathrm{Q}^{\tau'(t)})\big) \Big] \\
&= \mathop{\mathbb{E}}_{j,\, \boldsymbol{x}^{\le i}_{<j},\, C,\, \tilde{Q}} \left[\mathop{\mathbb{E}}_{x^i_j \sim X^i_j \mid \boldsymbol{x}^{\le i}_{<j}} \Big[-\log \bar{D}^i\big(x^i_j \,\big\|\, \bar{x}^i_{<j}\big) + \log \mathrm{Q}^{\tau'(t)}\Big(x^i_j \,\Big\|\, \bar{x}^{\le i}_{<j}\Big) \Big] \right] \\
&\le \epsilon^2 \delta.
\end{aligned}$$

Thus,

$$\mathop{\mathbb{E}}_{j,\boldsymbol{x}^{\leq i}_{<j},C,\tilde{Q}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^i_{<j};\bar{D}^i)\big)\Big] \leq \mathop{\mathbb{E}}_{j,\boldsymbol{x}^{\leq i}_{<j}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\leq i}_{<j};\mathrm{Q}^{\tau'(t)})\big)\Big] + \epsilon^2\delta.$$

Recall that this bound holds with probability at least $1-\delta$ over the choices of i and $\boldsymbol{x}^{<i}$.

We now use the following claim.

Claim 8.3. *Let τ' be a sufficiently large polynomial. With probability at least $1-\delta$ over the choices of i and $\boldsymbol{x}^{<i}$,*

$$\mathop{\mathbb{E}}_{j,\boldsymbol{x}^{i}_{<j}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\leq i}_{<j};\mathrm{Q}^{\tau'(t)})\big)\Big] \leq \epsilon^2\delta.$$

The proof is based on a standard probabilistic argument using the chain rule for KL divergence and is deferred. We now fix an arbitrary τ' that satisfies this condition.

Combining this claim with the previous bound and applying the union bound, we conclude that with probability at least $1-2\delta$ over the choices of i and $\boldsymbol{x}^{<i}$,

$$\begin{aligned}\mathop{\mathbb{E}}_{j,\boldsymbol{x}^{\leq i}_{<j},C,\tilde{Q}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^i_{<j};\bar{D}^i)\big)\Big] &\leq \mathop{\mathbb{E}}_{j,\boldsymbol{x}^{\leq i}_{<j}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\leq i}_{<j};\mathrm{Q}^{\tau'(t)})\big)\Big] + \epsilon^2\delta \\ &\leq \epsilon^2\delta + \epsilon^2\delta = 2\epsilon^2\delta.\end{aligned}$$

From the non-negativity of the KL divergence and Markov's inequality, we have

$$\Pr_{j,\boldsymbol{x}^{\leq i}_{<j}}\left[\mathop{\mathbb{E}}_{C,\tilde{Q}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^i_{<j};\bar{D}^i)\big)\Big] \leq 2\epsilon^2\right] \geq 1-\delta.$$

We fix arbitrarily $j, \boldsymbol{x}^{\leq i}_{<j}$ satisfying the event above. Let $\overline{\mathsf{Next}}_n(\bar{x}^i_{<j};\bar{D}^i)$ denote the distribution obtained from $\mathsf{Next}_n(\bar{x}^i_{<j};\bar{D}^i)$ using a uniformly random seed for C and $\tilde{Q}$ (which determines the distribution $\bar{D}^i$). Then we obtain

$$\begin{aligned}\Delta_{\mathrm{tv}}\Big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j},\, \overline{\mathsf{Next}}_n(\bar{x}^i_{<j};\bar{D}^i)\Big) &\leq \sqrt{\tfrac{1}{2}\cdot \mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \overline{\mathsf{Next}}_n(\bar{x}^i_{<j};\bar{D}^i)\big)} \\ &\leq \sqrt{\tfrac{1}{2}\cdot \mathop{\mathbb{E}}_{C,\tilde{Q}}\Big[\mathrm{KL}\big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^i_{<j};\bar{D}^i)\big)\Big]} \\ &\leq \sqrt{\tfrac{1}{2}\cdot 2\epsilon^2} \\ &= \epsilon,\end{aligned}$$

where the first inequality follows from Pinsker's inequality, and the second inequality follows from Jensen's inequality.

By the union bound, we conclude that

$$\Pr_{i,j,\boldsymbol{x}^{\leq i}_{<j}}\Big[\Delta_{\mathrm{tv}}\Big(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j},\, \overline{\mathsf{Next}}_n(\bar{x}^i_{<j};\bar{D}^i)\Big) \leq \epsilon\Big] \geq 1-3\delta. \tag{26}$$

This yields the construction of our inference algorithm L, which performs sampling according to $\overline{\mathsf{Next}}_n(\bar{x}^i_{<j};\bar{D}^i)$.

We now specify L in a more formal manner. Given parameters $n, s, t, \epsilon^{-1}, \delta^{-1}, \eta^{-1}$, the algorithm L first computes

$$\eta^{-1} = c_\eta \delta^{-1}\epsilon^{-2} n \log t \quad \text{and} \quad w = \left\lceil c_w \delta^{-2}\epsilon^{-2}\eta^{-1}\ell \log t \right\rceil,$$

and then verifies whether $t\delta^{-1}\eta^{-1} \leq 2^s$. If this condition does not hold (i.e., $t\delta^{-1}\eta^{-1} > 2^s$), L executes the inference algorithm from Proposition 4.12, which works for $m \geq O(\epsilon^{-2}\delta^{-1}s)$ and runs in time $\mathsf{poly}(2^s, t, \epsilon^{-1}, \delta^{-1}) \leq \mathsf{poly}(t, \epsilon^{-1}, \delta^{-1})$. Thus, it suffices to consider the case in which $t\delta^{-1}\eta^{-1} \leq 2^s$.

Given the past stream $\boldsymbol{x}_{<\iota}$, L partitions them into blocks, each consisting of w strings, and identifies a pair of indices $(i, j) \in \mathbb{N} \times [w]$ corresponding to the next position ι. Let b be the largest integer $b \in \mathbb{N}$ such that $bw \leq m$. This b is introduced only for the analysis, and L does not need to know it (since computing b would require the total bound m of messages, which is not given). We now assume that $i \leq b$ and ignore the case $i > b$ (which will be regarded as a confidence error). Thus, we can parse the past stream as $(\boldsymbol{x}^{<i}, \boldsymbol{x}^i_{<j})$ and compute $\bar{x}^i_{<j}$ by concatenation.

Then, L makes its prediction by sampling $\tilde{x}$ from $\overline{\mathsf{Next}}_n(\bar{x}^i_{<j}; \bar{D}^i)$. This procedure is carried out as follows. First, L selects the randomness required for executing C and $\tilde{Q}$, which in turn determines $\bar{D}^i$. Then L obtains a sample from $\mathsf{Next}_n(\bar{x}^i_{<j}; \bar{D}^i)$ by (i) selecting $\alpha \in [\ell]$ with probability $\frac{p_\alpha(\bar{x}^i_{<j})}{\sum_{\alpha'} p_{\alpha'}(\bar{x}^i_{<j})}$, where $p_\alpha(\bar{x}^i_{<j})$ is the probability that a prefix of a sample from D^i_α corresponds to x and (ii) sampling $\tilde{x}$ with probability $D^i_\alpha(\tilde{x} \,\|\, \bar{x}^i_{<j})$. Notice that, by Equation (25), this procedure is equivalent to sampling $\tilde{x}$ directly from $\overline{\mathsf{Next}}_n(\bar{x}^i_{<j}; \bar{D}^i)$.

The first step is performed by computing $p_\alpha(\bar{x}^i_{<j})$ for all $\alpha \in [\ell]$ via executing

$$\tilde{Q}(\text{-}; C(\boldsymbol{x}^{<i}; 1^\alpha, 1^t),\, 1^{\tau(t)},\, 1^{O(t)},\, 1^{\mathsf{poly}(t)}).$$

The second step is carried out by executing the same next-bit generator $\tilde{Q}$ with the same advice $(C(\boldsymbol{x}^{<i}; 1^\alpha, 1^t),\, 1^{\tau(t)},\, 1^{O(t)},\, 1^{\mathsf{poly}(t)})$ and constructing $\tilde{x}$ by sequentially generating each bit.

The total running time is at most $\mathsf{poly}(n, m, t, \ell) \leq \mathsf{poly}(t)$, where we used $n \leq t$ and $b_0 w \leq m \leq t$, which holds since the entire stream is generated within t time.

Since L can exactly sample from $\overline{\mathsf{Next}}_n(\bar{x}^i_{<j}; \bar{D}^i)$ without any approximation, the guarantee follows from Equation (26). By combining this with Claim 8.2 and the union bound, we obtain

$$\begin{aligned}
&\Pr_{\iota, \boldsymbol{x}_{<\iota}}\left[\Delta_{\mathrm{tv}}\big(X_\iota \mid \boldsymbol{x}_{<\iota},\, L(\boldsymbol{x}_{<\iota}; n, s, t, \epsilon^{-1}, \delta^{-1})\big) \leq \epsilon\right] \\
&\quad \geq \Pr_{\iota \sim [m]}[\iota \leq bw] \cdot \Pr_{i,j,\boldsymbol{x}^{\leq i}_{<j}}\left[\Delta_{\mathrm{tv}}\left(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j},\, \overline{\mathsf{Next}}_n(\bar{x}^i_{<j}; \bar{D}^i)\right) \leq \epsilon\right] \\
&\quad \geq 1 - 4\delta.
\end{aligned}$$

Hence, the statement follows after replacing δ^{-1} with $4\delta^{-1}$ throughout the argument.

The deferred proofs are provided below, thereby completing the proof.

Proof of Claim 8.2. Since b is the largest integer such that $bw \leq m$, we have $(b+1)w > m$, and hence $bw > m - w$. The claim is verified as follows:

$$\Pr_{i \sim [m]}[i \leq bw] = \frac{bw}{m} \geq \frac{m-w}{m} = 1 - \frac{w}{m} \geq 1 - \frac{1}{b_0} \geq 1 - \delta,$$

where the second inequality follows from $m \geq b_0 w$, and the last inequality holds since

$$\begin{aligned}
b_0 &= c_1 \delta^{-1} \eta^{-1} \frac{s}{\ell \log t} \\
&\geq c_1 \delta^{-1} \cdot c_\eta \delta^{-1} \epsilon^{-2} n \log t \cdot \frac{s}{c_1 \log(\delta^{-1}\eta^{-1}) \cdot \log t} \\
&\geq \delta^{-1} \cdot \frac{s}{\log(\delta^{-1}\eta^{-1})} \geq \delta^{-1}.
\end{aligned}$$

◇

Proof of Claim 8.3. For any sufficiently large polynomial τ', by Proposition 4.4 (domination property), we have that for any $\boldsymbol{x} = (x_1, \ldots, x_m)$ in the domain,

$$\begin{aligned}\mathrm{Q}^{\tau'(t)}(\bar{x}) &\geq 2^{-O(s+\log(st\delta^{-1}\epsilon^{-1}))} \cdot \Pr[\bar{X}_{\leq bw} = \bar{x}] \\ &\geq 2^{-O(s)} \cdot \Pr[\bar{X}_{\leq bw} = \bar{x}],\end{aligned}$$

where $\bar{x} = x_1 \circ \cdots \circ x_{bw}$ and $\bar{X}_{\leq bw} = X_1 \circ \cdots \circ X_{bw}$, with X_i denoting the random variable of the i-th string. Note that the additive term $O(\log st\delta^{-1}\epsilon^{-1})$ accounts for encoding b and w.

Therefore, we have

$$\mathrm{KL}(\bar{X}_{\leq bw} \| \mathrm{Q}^{\tau'(t)}) = \mathop{\mathbb{E}}_{\bar{x}}\left[\log \frac{\Pr[\bar{X}_{\leq bw} = \bar{x}]}{\mathrm{Q}^{\tau'(t)}(\bar{x})}\right] \leq O(s).$$

Applying the chain rule for KL divergence, for any sufficiently large $t \in \mathbb{N}$,

$$\begin{aligned}&\mathop{\mathbb{E}}_{(i,j)\sim[b]\times[w],\, \boldsymbol{x}^{<i},\, \boldsymbol{x}^{i}_{<j}} \left[\mathrm{KL}(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\leq i}_{<j}; \mathrm{Q}^{\tau'(t)}))\right] \\ &= \frac{1}{bw} \sum_{(i,j)\sim[b]\times[w]} \mathop{\mathbb{E}}_{\boldsymbol{x}^{\leq i}_{<j}} \left[\mathrm{KL}(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\leq i}_{<j}; \mathrm{Q}^{\tau'(t)}))\right] \\ &= \frac{1}{bw} \cdot \mathrm{KL}(\bar{X}_{\leq bw} \| \mathrm{Q}^{\tau'(t)}) \\ &\leq \frac{O(s)}{bw} \leq \frac{O(s)}{s \cdot \Omega(\epsilon^{-2}\delta^{-3}\eta^{-2})} \leq \epsilon^2\delta^2.\end{aligned}$$

From the nonnegativity of KL divergence and Markov's inequality, we obtain

$$\Pr_{i,\, \boldsymbol{x}^{<i}}\left[\mathop{\mathbb{E}}_{j,\, \boldsymbol{x}^{i}_{<j}} \left[\mathrm{KL}(X^i_j \mid \boldsymbol{x}^{\leq i}_{<j} \,\big\|\, \mathsf{Next}_n(\bar{x}^{\leq i}_{<j}; \mathrm{Q}^{\tau'(t)}))\right] \leq \epsilon^2\delta\right] \geq 1 - \delta.$$

◇

□

9 Improving Time and Sample Complexity in IID Cases

In this section we present a fully polynomial-time inference algorithm for i.i.d. samples from an unknown target distribution that achieves improved sample complexity.

This task falls under the *distributional learning model* introduced by Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie [KMRRSS94]. For completeness we recall the definition.

Definition 9.1 (Distributional Learning [KMRRSS94])**.** *Let $\mathscr{D} = \{\mathscr{D}_n\}_{n\in\mathbb{N}}$ be a class of distributions, where each $\mathscr{D}_n$ is a set of distributions over $\{0,1\}^n$, and let $m\colon \mathbb{N} \times (0,1] \times (0,1] \to \mathbb{N}$.*

We say that $\mathscr{D}$ is distributionally learnable in polynomial time with sample complexity $m(n,\epsilon,\delta)$ *if there exists a randomized polynomial-time oracle machine L such that for every $n, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, and every distribution $\mathcal{D} \in \mathscr{D}_n$, the algorithm $L^{\mathcal{D}}(1^n, 1^{\epsilon^{-1}}, 1^{\delta^{-1}})$ draws at most $m(n,\epsilon,\delta)$ i.i.d. samples from its sampling oracle for $\mathcal{D}$ and outputs a description of a sampling circuit $h : \{0,1\}^\ell \to \{0,1\}^n$ satisfying*

$$\Delta_{\mathrm{tv}}\big(h(U_\ell),\, \mathcal{D}\big) \leq \epsilon,$$

with probability at least $1-\delta$ over the samples and the internal randomness of L. Here U_ℓ denotes the uniform distribution over $\{0,1\}^\ell$.

The main result of this section (restated from Corollary 2.1 in Section 2) is as follows.

Theorem 9.1. *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP, *then for every pair of polynomials* $s(\cdot)$ *and* $t(\cdot)$, *the class of all distributions over* $\{0,1\}^n$ *that are samplable in time* $t(n)$ *by a Turing machine whose description length is at most* $s(n)$ *is distributionally learnable in polynomial time with sample complexity*

$$O\big(s(n)\,\epsilon^{-2}\log\delta^{-1}\big).$$

We will use the following auxiliary lemma as a technical tool in the proof.

Lemma 9.1. *Assuming auxiliary-input one-way functions do not exist, there is a randomized polynomial-time algorithm* $\tilde{\Delta}$ *such that for every pair of circuit descriptions* D_0, D_1 *and every* $\epsilon, \delta \in (0,1]$,

$$\Pr_{\tilde{\Delta}}\Big[\,\tilde{\Delta}\Big(D_0, D_1; 1^{\epsilon^{-1}}, 1^{\delta^{-1}}\Big) \in \big[\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)-\epsilon,\ \Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)+\epsilon\big]\Big] \geq 1-\delta,$$

where $\mathcal{D}_b$ *denotes the distribution of* $D_b(U_\ell)$ *for a uniformly random seed* U_ℓ *(and* $b \in \{0,1\}$*).*

We refer to ϵ and δ as the *accuracy* and *confidence* parameters, respectively.

The proof of the lemma follows the technique used in [NR06, Theorem 4.1]. At a high level, use an inverter for auxiliary-input functions to (distributionally) invert $f_{D_0,D_1}(b,r) = D_b(r)$ with $b \leftarrow \{0,1\}$ and (D_0, D_1) as auxiliary input. We then empirically estimate the predictability of b from $x \leftarrow D_b(r)$ via the inverter, and recover the statistical distance from this success probability. See Appendix D for the formal proof.

We now prove Theorem 9.1 using Lemma 9.1.

Proof of Theorem 9.1. Let L be the inference algorithm from Theorem 7.1. Fix a sampler D of description length $s = s(n)$ that runs in time $t = t(n)$ and samples $\mathcal{D}$. Consider a program Π of size $s' = O(s + \log n)$ that hard-wires the description of D and the parameter n (in binary) and outputs a fresh sample from $\mathcal{D}$ on fresh random coins at each round. We treat Π as the target program to be inferred by L.

Base learner. Set $\epsilon_L := \epsilon/5$ and $\delta_L := 1/4$, and let

$$m_L := c\,s'\,\epsilon_L^{-2}\delta_L^{-1}$$

for a sufficiently large universal constant $c > 0$. The base learner L_{base} chooses $i \sim [m_L]$ uniformly at random, draws $x_1, \ldots, x_{i-1} \sim \mathcal{D}$ i.i.d. from its oracle, and runs L on the prefix $x_{<i}$ with parameters $n, s', t, \epsilon_L^{-1}, \delta_L^{-1}$, outputting the sampler h produced by L (which hard-wires $x_{<i}$).

By Theorem 7.1, with probability at least $1 - \delta_L = 3/4$ over the choice of i and $x_{<i}$, the output h satisfies

$$\Delta_{\mathrm{tv}}\big(\mathcal{H}, \mathcal{D}\big) \leq \epsilon_L = \epsilon/5,$$

where $\mathcal{H}$ is the distribution induced by $h(U_\ell)$; we used that the conditional distribution of the ith sample given $x_{<i}$ is exactly $\mathcal{D}$ in the i.i.d. setting. The running time is $\mathsf{poly}(n, \epsilon^{-1})$, and the sample complexity is $m_L = O(s\,\epsilon^{-2})$ (assuming $\log n \leq s$; otherwise, since $2^s \leq n$, one can instead invoke the exponential-time inference algorithm from Proposition 4.12 (as in prior work [HN23]), which runs in time $2^{O(s)} \cdot \mathsf{poly}(\epsilon^{-1}) \leq \mathsf{poly}(n, \epsilon^{-1})$ and uses $O(s\,\epsilon^{-2})$ samples.

Confidence boosting. Repeat L_{base} independently

$$N := \left\lceil 32 \ln \frac{2}{\delta} \right\rceil$$

times to obtain hypotheses $h_1, \ldots, h_N$. By Hoeffding's inequality, with probability at least $1 - \delta/2$, at least a $5/8$ fraction of them satisfy $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \le \epsilon/5$, where $\mathcal{H}_i$ is the distribution of $h_i(U_\ell)$.

To identify a good hypothesis without the description of the sampler D, apply $\tilde{\Delta}$ to every pair (h_i, h_j) with accuracy $\epsilon/5$ and confidence

$$\delta' := \frac{\delta}{2\binom{N}{2}},$$

obtaining estimates $\widehat{\Delta}_{i,j}$ of $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j)$. Let $C \subseteq [N]$ be a maximum-size subset such that

$$i, j \in C \text{ and } i < j \implies \widehat{\Delta}_{i,j} \le 3\epsilon/5.$$

This can be found by solving a maximum-clique instance in time $O(2^N) = \mathsf{poly}(\delta^{-1})$, where we place an (undirected) edge (i, j) if $\widehat{\Delta}_{i,j} \le 3\epsilon/5$ (with $i < j$). Output any h_i with $i \in C$.

By a union bound over the $\binom{N}{2}$ estimator calls, with probability at least $1 - \delta/2$ we have

$$\left|\widehat{\Delta}_{i,j} - \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j)\right| \le \epsilon/5 \quad \text{for all } i < j.$$

Together with the previous event

$$\left|\{i \in [N] : \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \le \epsilon/5\}\right| \ge 5N/8$$

(which holds with probability at least $1 - \delta/2$), a union bound implies that both events hold with probability at least $1 - \delta$.

Let $G := \{i \in [N] : \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \le \epsilon/5\}$. Conditioning on these events, for distinct $i, j \in G$,

$$\widehat{\Delta}_{i,j} \le \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j) + \epsilon/5 \le \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) + \Delta_{\mathrm{tv}}(\mathcal{H}_j, \mathcal{D}) + \epsilon/5 \le 3\epsilon/5.$$

Thus, G forms a clique. Furthermore, since $|G| \ge 5N/8$, it must be contained in the maximal clique; hence $G \subseteq C$.

Fix any $i \in C$ and $j \in G$; then

$$\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \le \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j) + \Delta_{\mathrm{tv}}(\mathcal{H}_j, \mathcal{D}) \le \widehat{\Delta}_{i,j} + \epsilon/5 + \epsilon/5 \le \epsilon.$$

Thus the output hypothesis has total variation distance at most ϵ from $\mathcal{D}$ with probability at least $1 - \delta$. Since $N = O(\log \delta^{-1})$, the overall sample complexity is $N \cdot m_L = O\big(s\epsilon^{-2} \log \delta^{-1}\big)$, as claimed. □

Acknowledgment

Shuichi Hirahara was supported by JST, FOREST Grant Number JPMJFR226Y. Mikito Nanashima was supported by JST, ACT-X Grant Number JPMJAX24CJ. Part of this work was carried out while the authors were visiting the EnCORE Institute.

References

[ABX08] Benny Applebaum, Boaz Barak, and David Xiao. "On Basing Lower-Bounds for Learning on Worst-Case Assumptions". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2008, pp. 211–220. DOI: 10.1109/FOCS.2008.35.

[AF09] Luis Filipe Coelho Antunes and Lance Fortnow. "Worst-Case Running Times for Average-Case Algorithms". In: *Proceedings of the Conference on Computational Complexity (CCC)*. 2009, pp. 298–303. DOI: 10.1109/CCC.2009.12.

[AFMV06] Luis Antunes, Lance Fortnow, Dieter van Melkebeek, and N. V. Vinodchandran. "Computational depth: Concept and applications". In: *Theor. Comput. Sci.* 354.3 (2006), pp. 391–404. DOI: 10.1016/j.tcs.2005.11.033.

[AFPS12] Luis Filipe Coelho Antunes, Lance Fortnow, Alexandre Pinto, and Andre Souto. "Low-Depth Witnesses are Easy to Find". In: *Comput. Complex.* 21.3 (2012), pp. 479–497. DOI: 10.1007/s00037-011-0025-1.

[Ben88] C. H. Bennett. "Logical Depth and Physical Complexity". In: *The universal Turing machine, a half century survey* (1988), pp. 227–257.

[BKST25] Guy Blanc, Caleb Koch, Carmen Strassle, and Li-Yang Tan. "Computational-Statistical Tradeoffs from NP-hardness". In: *CoRR* abs/2507.13222 (2025). DOI: 10.48550/ARXIV.2507.13222. arXiv: 2507.13222.

[BLH23] Jörg Bornschein, Yazhe Li, and Marcus Hutter. "Sequential Learning of Neural Networks for Prequential MDL". In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/forum?id=dMMPUvNSYJr.

[BO18] Léonard Blier and Yann Ollivier. "The Description Length of Deep Learning models". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 2018, pp. 2220–2230.

[BT06] Andrej Bogdanov and Luca Trevisan. "On Worst-Case to Average-Case Reductions for NP Problems". In: *SIAM J. Comput.* 36.4 (2006), pp. 1119–1159. DOI: 10.1137/S0097539705446974.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)* Wiley, 2006. ISBN: 978-0-471-24195-9.

[Daw84] A Philip Dawid. "Present position and potential developments: Some personal views statistical theory the prequential approach". In: *Journal of the Royal Statistical Society: Series A (General)* 147.2 (1984), pp. 278–290.

[DRDCGMGWAOHV24] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. "Language Modeling Is Compression". In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=jznbgiynus.

[DV99] A. Philip Dawid and Vladimir Vovk. "Prequential Probability: Principles and Properties". In: *Bernoulli* 5.1 (Feb. 1999), pp. 125–162. DOI: 10.2307/3318616.

[GK21] Halley Goldberg and Valentine Kabanets. “A Simplified Proof of Hirahara’s Theorem”. manuscript. 2021.

[GK22] Halley Goldberg and Valentine Kabanets. “A Simpler Proof of the Worst-Case to Average-Case Reduction for Polynomial Hierarchy via Symmetry of Information”. In: *Electronic Colloquium on Computational Complexity (ECCC)* 007 (2022).

[GK23] Halley Goldberg and Valentine Kabanets. “Improved Learning from Kolmogorov Complexity”. In: *Proceedings of the Computational Complexity Conference (CCC)*. 2023, 12:1–12:29. DOI: 10.4230/LIPICS.CCC.2023.12.

[GKKZ22] Surbhi Goel, Sham M. Kakade, Adam Kalai, and Cyril Zhang. “Recurrent Convolutional Neural Networks Learn Succinct Learning Algorithms”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. 2022.

[GKLO22] Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. “Probabilistic Kolmogorov Complexity with Applications to Average-Case Complexity”. In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 16:1–16:60. DOI: 10.4230/LIPICS.CCC.2022.16. URL: https://doi.org/10.4230/LIPIcs.CCC.2022.16.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. ISBN: 0-521-79172-3. DOI: 10.1017/CBO9780511546891.

[Grü07] Peter D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, Mar. 2007. ISBN: 9780262256292. DOI: 10.7551/mitpress/4643.001.0001.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. “A Pseudorandom Generator from any One-way Function”. In: *SIAM J. Comput.* 28.4 (1999), pp. 1364–1396. DOI: 10.1137/S0097539793244708.

[HILNO23] 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: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2023, pp. 1039–1050. DOI: 10.1145/3564246.3585138.

[Hir18] Shuichi Hirahara. “Non-black-box Worst-case to Average-case Reductions within NP”. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2018, pp. 247–258.

[Hir20a] Shuichi Hirahara. “Characterizing Average-Case Complexity of PH by Worst-Case Meta-Complexity”. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2020, pp. 50–60.

[Hir20b] Shuichi Hirahara. "Non-Disjoint Promise Problems from Meta-Computational View of Pseudorandom Generator Constructions". In: *Proceedings of the Computational Complexity Conference (CCC)*. 2020, 20:1–20:47. DOI: 10.4230/LIPIcs.CCC.2020.20.

[Hir21] Shuichi Hirahara. "Average-case hardness of NP from exponential worst-case hardness assumptions". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2021, pp. 292–302. DOI: 10.1145/3406325.3451065.

[Hir22a] Shuichi Hirahara. "NP-Hardness of Learning Programs and Partial MCSP". In: *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 2022, pp. 968–979. DOI: 10.1109/FOCS54457.2022.00095. URL: https://doi.org/10.1109/FOCS54457.2022.00095.

[Hir22b] Shuichi Hirahara. "Symmetry of Information from Meta-Complexity". In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 26:1–26:41. DOI: 10.4230/LIPICS.CCC.2022.26. URL: https://doi.org/10.4230/LIPIcs.CCC.2022.26.

[Hir23] Shuichi Hirahara. "Capturing One-Way Functions via NP-Hardness of Meta-Complexity". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*. Ed. by Barna Saha and Rocco A. Servedio. ACM, 2023, pp. 1027–1038. DOI: 10.1145/3564246.3585130. URL: https://doi.org/10.1145/3564246.3585130.

[HJW24] Yanjun Han, Tianze Jiang, and Yihong Wu. "Prediction from compression for models with infinite memory, with applications to hidden Markov and renewal processes". In: *The Thirty Seventh Annual Conference on Learning Theory, June 30 - July 3, 2023, Edmonton, Canada*. Ed. by Shipra Agrawal and Aaron Roth. Proceedings of Machine Learning Research. PMLR, 2024, pp. 2270–2307. URL: https://proceedings.mlr.press/v247/han24a.html.

[HLN24] Shuichi Hirahara, Zhenjian Lu, and Mikito Nanashima. "Optimal Coding for Randomized Kolmogorov Complexity and Its Applications". In: *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024, pp. 369–378. DOI: 10.1109/FOCS61266.2024.00030. URL: https://doi.org/10.1109/FOCS61266.2024.00030.

[HLO24] Shuichi Hirahara, Zhenjian Lu, and Igor C. Oliveira. "One-Way Functions and pKt Complexity". In: *Proceedings of the Theory of Cryptography Conference (TCC)*. 2024, pp. 253–286. DOI: 10.1007/978-3-031-78011-0_9.

[HN21] Shuichi Hirahara and Mikito Nanashima. "On Worst-Case Learning in Relativized Heuristica". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2021, pp. 751–758. DOI: 10.1109/FOCS52979.2021.00078.

[HN22] Shuichi Hirahara and Mikito Nanashima. “Finding Errorless Pessiland in Error-Prone Heuristica”. In: *Proceedings of the Computational Complexity Conference (CCC)*. 2022, 25:1–25:28. DOI: 10.4230/LIPIcs.CCC.2022.25.

[HN23] Shuichi Hirahara and Mikito Nanashima. “Learning in Pessiland via Inductive Inference”. In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 447–457. DOI: 10.1109/FOCS57990.2023.00033. URL: https://doi.org/10.1109/FOCS57990.2023.00033.

[HQC24] Marcus Hutter, David Quarel, and Elliot Catt. *An Introduction to Universal Artificial Intelligence*. 2024. URL: http://www.hutter1.net/ai/uaibook2.htm.

[HS17] Shuichi Hirahara and Rahul Santhanam. “On the Average-Case Complexity of MCSP and Its Variants”. In: *Proceedings of the Computational Complexity Conference (CCC)*. 2017, 7:1–7:20. DOI: 10.4230/LIPIcs.CCC.2017.7.

[HS22] Shuichi Hirahara and Rahul Santhanam. “Errorless versus Error-prone Average-case Complexity”. In: *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*. 2022, 38:1–38:23.

[Hut05] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Berlin: Springer, 2005. ISBN: 3-540-22139-5. DOI: 10.1007/b138233.

[IL89] Russell Impagliazzo and Michael Luby. “One-way Functions are Essential for Complexity Based Cryptography (Extended Abstract)”. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 1989, pp. 230–235. DOI: 10.1109/SFCS.1989.63483.

[IL90] Russell Impagliazzo and Leonid A. Levin. “No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random”. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 1990, pp. 812–821. DOI: 10.1109/FSCS.1990.89604.

[Ila23] Rahul Ilango. “SAT Reduces to the Minimum Circuit Size Problem with a Random Oracle”. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 733–742. DOI: 10.1109/FOCS57990.2023.00048.

[Imp95] Russell Impagliazzo. “A Personal View of Average-Case Complexity”. In: *Proceedings of the Structure in Complexity Theory Conference*. 1995, pp. 134–147. DOI: 10.1109/SCT.1995.514853.

[KK25] Valentine Kabanets and Antonina Kolokolova. “Chain Rules for Time-Bounded Kolmogorov Complexity”. In: *Electron. Colloquium Comput. Complex.* TR25-089 (2025). ECCC: TR25-089. URL: https://eccc.weizmann.ac.il/report/2025/089/.

[KMRRSS94] Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. "On the learnability of discrete distributions". In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*. Ed. by Frank Thomson Leighton and Michael T. Goodrich. ACM, 1994, pp. 273–282. DOI: 10.1145/195058.195155. URL: https://doi.org/10.1145/195058.195155.

[LM93] Luc Longpré and Sarah Mocas. "Symmetry of Information and One-Way Functions". In: *Inf. Process. Lett.* 46.2 (1993), pp. 95–100. DOI: 10.1016/0020-0190(93)90204-M.

[LOZ22] Zhenjian Lu, Igor C. Oliveira, and Marius Zimand. "Optimal Coding Theorems in Time-Bounded Kolmogorov Complexity". In: *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*. Ed. by Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff. Vol. 229. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 92:1–92:14. DOI: 10.4230/LIPICS.ICALP.2022.92. URL: https://doi.org/10.4230/LIPIcs.ICALP.2022.92.

[LP20] Yanyi Liu and Rafael Pass. "On One-way Functions and Kolmogorov Complexity". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2020, pp. 1243–1254.

[LV19] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition*. Texts in Computer Science. Springer, 2019. ISBN: 978-3-030-11297-4. DOI: 10.1007/978-3-030-11298-1.

[LW95] Luc Longpré and Osamu Watanabe. "On Symmetry of Information and Polynomial Time Invertibility". In: *Inf. Comput.* 121.1 (1995), pp. 14–22. DOI: 10.1006/inco.1995.1120.

[MF98] Neri Merhav and Meir Feder. "Universal Prediction". In: *IEEE Trans. Inf. Theory* 44.6 (1998), pp. 2124–2147. DOI: 10.1109/18.720534. URL: https://doi.org/10.1109/18.720534.

[NR06] Moni Naor and Guy N. Rothblum. "Learning to impersonate". In: *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*. Ed. by William W. Cohen and Andrew W. Moore. Vol. 148. ACM International Conference Proceeding Series. ACM, 2006, pp. 649–656. DOI: 10.1145/1143844.1143926. URL: https://doi.org/10.1145/1143844.1143926.

[OW93] Rafail Ostrovsky and Avi Wigderson. "One-Way Fuctions are Essential for Non-Trivial Zero-Knowledge". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 1993, pp. 3–17. DOI: 10.1109/ISTCS.1993.253489.

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

[Sol64a] Ray J. Solomonoff. "A Formal Theory of Inductive Inference. Part I". In: *Inf. Control.* 7.1 (1964), pp. 1–22. DOI: 10.1016/S0019-9958(64)90223-2.

[Sol64b] Ray J. Solomonoff. "A Formal Theory of Inductive Inference. Part II". In: *Inf. Control.* 7.2 (1964), pp. 224–254. DOI: 10.1016/S0019-9958(64)90131-7.

[SV15] Igal Sason and Sergio Verdú. "Upper bounds on the relative entropy and Rényi divergence as a function of total variation distance for finite alphabets". In: *2015 IEEE Information Theory Workshop - Fall (ITW), Jeju Island, South Korea, October 11-15, 2015*. 2015, pp. 214–218. DOI: 10.1109/ITWF.2015.7360766.

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

[Xia10] David Xiao. "Learning to Create is as Hard as Learning to Appreciate". In: *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*. Ed. by Adam Tauman Kalai and Mehryar Mohri. Omnipress, 2010, pp. 516–528. URL: http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf%5C#page=524.

[ZL70] Alexander K Zvonkin and Leonid A Levin. "The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms". In: *Russian Mathematical Surveys* 25.6 (1970), pp. 83–124.

A On Extending the Proof of the Symmetry of Information

In this section, we briefly explain why a direct extension of the earlier proofs by Hirahara [Hir22b] and Goldberg and Kabanets [GK22] for the symmetry of information (i.e., the case $m = 2$) yields an additive term of $m^2 \cdot O(\log t)$ in the proof of the chain rule.

Indeed, the direct extension is applicable only in the following weaker form (for convenience we work with the pK-complexity; see Section 4.1 for the definition):

$$\sum_{i=1}^{m} \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}) \leq \mathsf{pK}^{t}(x_1, \ldots, x_m) + m \cdot O(\log t),$$

where $k_i := \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}) - O(\log t)$ for each $i \in [m-1]$. That is, we allow each round to receive, as additional advice, the previous complexity estimates $k_1, \ldots, k_{i-1}$.

Below we outline the proof, in which these complexity estimates are used in a hybrid argument, specifically to construct hybrid distributions. For simplicity, we use poly to denote polynomial time bounds, so that we do not have to explicitly track polynomial overheads.

Let p be a large enough polynomial determined by p_{DP} in Lemma 4.5. We define the pseudorandom string w_{DP} by applying the direct-product generator sequentially as follows:

$$w_{\mathsf{DP}} \sim \mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m),$$

where $z_1, \ldots, z_m$ are independent uniformly random seeds. The values $k_1, \ldots, k_m$ are defined in-

ductively as:

$$k_i := \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}) - c \log t \quad \text{for each } i \in [m-1], \text{ and}$$
$$k_m := \mathsf{pK}^{t}(x_1, \ldots, x_m) - \sum_{i=1}^{m-1} k_i + c \log t.$$

for a large enough constant $c > 0$.

Now consider a distinguisher D that, given a string w, approximates $\mathrm{K}^{\mathsf{poly}(t)}(w)$ using the algorithm $\tilde{K}$ from Proposition 4.3, and outputs 1 (i.e., interprets w as pseudorandom) if the approximated complexity is smaller than a threshold $\tau := \mathsf{pK}^t(x_1, \ldots, x_m) + \sum_i |z_i| + O(\log t)$.

If the input string is indeed the pseudorandom string w_{DP}, then intuitively, since w_{DP} is generated from the sequence $(x_1, \ldots, x_m)$ and random seeds $(z_1, \ldots, z_m)$, we expect:

$$\mathrm{K}^{\mathsf{poly}}(w_{\mathsf{DP}}) \leq \mathsf{pK}^t(x_1, \ldots, x_m) + \sum_{i=1}^{m} |z_i| + O(\log t) = \tau.$$

More precisely, one would also need to include some additional randomness to simulate the probabilistic algorithm deterministically. However, this technical detail is not essential for the current argument, so we omit it for simplicity.

By contrast, for a uniformly random string w of the same length $|w_{\mathsf{DP}}| = \sum_{i=1}^{m}(|z_i| + k_i)$, a standard counting argument implies that, with high probability,

$$\mathrm{K}(w) \geq \sum_{i=1}^{m}(|z_i| + k_i) - O(1) = \mathsf{pK}^t(x_1, \ldots, x_m) + \sum_{i=1}^{m} |z_i| + c \log t - O(1) > \tau$$

by choosing c sufficiently large. Thus, the distinguisher D can distinguish the pseudorandom string w_{DP} from a uniformly random string w with constant advantage.

Now, for each $i \in [m] \cup \{0\}$, we define the hybrid string hyb_i sampled as

$$\mathsf{hyb}_i \sim \mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_i}(x_i; z_i) \circ w_{i+1} \circ \cdots \circ w_m,$$

where each w_j for $j > i$ is an independent uniformly random string of the same length as $\mathsf{DP}_{k_j}(x_j; z_j)$.

Notice that hyb_0 is a uniformly random string, while and hyb_m is distributed identically to the pseudorandom string w_{DP}. Therefore, D distinguishes hyb_m from hyb_0.

Suppose that D does not distinguish hyb_{i-1} from hyb_i for every $i \in [m-1]$. Then, D must distinguish hyb_{m-1} from hyb_m (otherwise, D cannot distinguish hyb_m from hyb_0).

In this case, we can construct a distinguisher D_m for $\mathsf{DP}_{k_m}(x_m; z_m)$ as follows. Given an input w_m and advice $x_1, \ldots, x_{m-1}, k_1, \ldots, k_{m-1}$, the distinguisher D_m executes D on the concatenated string:

$$\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{m-1}}(x_{m-1}; z_{m-1}) \circ w_m$$

Note that each $\mathsf{DP}_{k_i}(x_i; z_i)$ is efficiently samplable given x_i and k_i. The above string behaves as hyb_m when w_m is sampled from $\mathsf{DP}_{k_m}(x_m; z_m)$, and as hyb_{m-1} when w_m is sampled uniformly at random.

Since D_m is specified by D along with the external advice $x_1, \ldots, x_{m-1}, k_1, \ldots, k_{m-1}$, the pK

reconstruction lemma (Lemma 4.5) yields:

$$\begin{aligned}
&\mathsf{pK}^{p(t)}\left(x_m \mid x_1, \ldots, x_{m-1}, k_1, \ldots, k_{i-1}\right) \\
&\leq k_m + O(\log t) \\
&\leq \mathsf{pK}^t(x_1, \ldots, x_m) - \sum_{i=1}^{m-1} k_i + O(\log t) \\
&= \mathsf{pK}^t(x_1, \ldots, x_m) - \sum_{i=1}^{m-1} \mathsf{pK}^{p(t)}\left(x_i \mid x_{<i}, k_1, \cdots, k_{i-1}\right) + m \cdot O(\log t).
\end{aligned}$$

Thus, the chain rule follows as long as hyb_{i-1} and hyb_i are indistinguishable for D for all $i \in [m-1]$.

Indeed, we can observe these indistinguishabilities, and this is precisely where the additional advice strings $k_1, \ldots, k_{i-1}$ become necessary.

Fix $i \in [m-1]$. Suppose that hyb_{i-1} and hyb_i are distinguishable by D. Then, we can construct a distinguisher D_i for $\mathsf{DP}_{k_i}(x_i)$ in the same way as we did for D_m. Applying the pK reconstruction lemma (Lemma 4.5) gives

$$\begin{aligned}
\mathsf{pK}^{p(t)}\left(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}\right) &\leq k_i + O(\log t m \gamma^{-1}) \\
&\leq \mathsf{pK}^{p(t)}\left(x_i \mid x_{<i}, k_1, \cdots, k_{i-1}\right) - c \log t + O(\log t),
\end{aligned}$$

which yields a contradiction for large enough constant c.

Here, the additional advice $k_1, \ldots, k_{i-1}$ is necessary to sample a hybrid string

$$\mathsf{DP}_{k_1}\left(x_1; z_1\right) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}\left(x_{m-1}; z_{m-1}\right) \circ w_i \circ w_{i+1} \circ \cdots \circ w_m,$$

given w_i as input. Without this advice, the distinguisher D_i cannot generate the appropriate prefix of the hybrid string needed to simulate D.

On Removing Advice Strings The issue is that it is unclear whether each k_i defined above can be computed in polynomial time, particularly because we do not yet know whether the conditional version of pK^t is efficiently computable under the assumption $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$. A natural way to handle this is to treat these quantities as external advice. This introduces an additive overhead of

$$\sum_{i=1}^{m} O(\log k_1 + \ldots + \log k_{i-1}) = m^2 \cdot O(\log t).$$

However, this is not sufficient for our purposes, since the additive term diverges in the amortized sense as m grows.

B Lower Bound on Round Complexity

In this section we present a lower bound for complexity theoretic universal inductive inference.

Proposition B.1. *For size parameter s and accuracy and confidence parameters $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, any algorithm, possibly time unbounded, that solves complexity-theoretic universal inductive inference must have round complexity at least*

$$\Omega\big(s\,\epsilon^{-2} + s\,\delta^{-1}\big).$$

Proof sketch. The $\Omega(s\,\epsilon^{-2})$ term. This follows from the reduction in [HN23], which shows that an inference algorithm with round complexity $\sigma(s,\epsilon^{-1})$ at a fixed constant confidence yields a binary classification learner with sample complexity $O(\sigma(s,\epsilon^{-1}))$ for a specific task that requires $\Omega(s\,\epsilon^{-2})$ samples; see [HN23, Section 10 and Appendix A]. Hence $\sigma(s,\epsilon^{-1}) = \Omega(s\,\epsilon^{-2})$.

The $\Omega(s\,\delta^{-1})$ term. Fix $n = 1$ for concreteness. Let s be large and consider a program Π_α indexed by a secret string $\alpha \in \{0,1\}^{s/2}$. On round $i \in [s/2]$ it outputs the bit α_i; after that it outputs any fixed bit. The description length of Π_α is $|\alpha| + O(\log s) \le s$ for large s.

Let L be any inference algorithm. For each prefix $\alpha_{<i}$, L yields a predictive distribution $p_i \in [0,1]$ for the next bit being 1. Define α adversarially by

$$\alpha_i \;=\; \begin{cases} 0 & \text{if } p_i \ge \frac{1}{2}, \\ 1 & \text{if } p_i < \frac{1}{2}. \end{cases}$$

Then the true conditional distribution at round $i \le s/2$ is the point mass at α_i, while L's prediction assigns probability p_i to 1. The total variation distance between these two distributions equals $|p_i - \alpha_i| \ge \frac{1}{2}$. Thus for any $\epsilon < \frac{1}{2}$ the prediction fails at every one of the first $s/2$ rounds.

If the total number of rounds is m, achieving success on a $(1-\delta)$ fraction of rounds requires

$$\frac{s/2}{m} \;\le\; \delta \qquad\Longrightarrow\qquad m \;\ge\; \frac{s}{2}\,\delta^{-1} \;=\; \Omega(s\,\delta^{-1}).$$

Combining the two parts yields the stated lower bound. □

C Inverting AIOWF from Chain Rule

In this section, we observe that the chain rule for q^t implies the nonexistence of auxiliary-input one-way functions. In the main body, we used the assumption GapMINKT $\in$ pr-BPP for two purposes: first, to obtain an algorithmic proof of the chain rule, and second, to deduce the nonexistence of auxiliary-input one-way functions. Hence, the first statement already yields the second, and the role of the assumption GapMINKT $\in$ pr-BPP is only to establish the chain rule.

Proposition C.1. *Suppose the chain rule for q^t holds. That is, there exist an absolute constant $c > 0$ and a polynomial τ such that for every m and every m-tuple $\boldsymbol{x} = (x_1, \ldots, x_m)$ of binary strings, for all $t \ge c \cdot |\boldsymbol{x}|$,*

$$\sum_{i=1}^{m} \mathrm{q}^{\tau(t)}\big(x_i \mid \boldsymbol{x}_{<i}\big) \;\le\; \mathrm{q}^t(\boldsymbol{x}) + m \cdot c \log t.$$

Then, no auxiliary-input one-way function exists.

Proof sketch. From the standard construction of pseudorandom generators from one-way functions [HILL99], an auxiliary-input one-way function yields an auxiliary-input variant of pseudorandom generator. It therefore suffices to show that the chain rule for q^t is sufficient to break any auxiliary-input pseudorandom generator.

By the usual hybrid argument, the stretch of any auxiliary-input pseudorandom generator can be increased to any fixed polynomial [Gol01, Section 3.3.2]. Hence fix $n \in \mathbb{N}$ and let

$$G : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{n^2}$$

be an auxiliary-input pseudorandom generator. The security requirement is that for every polynomial time randomized algorithm A and every $z \in \{0,1\}^n$, the distinguisher A given z cannot distinguish $G(z,x)$ for $x \sim \{0,1\}^n$ from a uniform string in $\{0,1\}^{n^2}$.

We describe a distinguisher A for G. On input (z,y), where y is either $G(z,x)$ for uniform x or a uniform string in $\{0,1\}^{n^2}$, parse y into n consecutive blocks $y^1,\dots,y^n \in \{0,1\}^n$. For each $i \in [n]$, the adversary draws polynomially many independent samples from $\mathrm{Q}^{\tau(t),\boldsymbol{y}^{<i}}$, where $\boldsymbol{y}^{<i} = (y^1,\dots,y^{i-1})$, and tests whether

$$\mathrm{Q}^{\tau(t),\boldsymbol{y}^{<i}}(y^i) \;\geq\; 1/\mathsf{poly}(t),$$

where $t = \mathsf{poly}(n)$ is chosen sufficiently large compared to the running time of G. If this event occurs for some i, A accepts; otherwise A rejects. Recall that sampling from $\mathrm{Q}^{\tau(t),\boldsymbol{y}^{<i}}$ is simply performed by executing the universal Turing machine U with auxiliary advice $\boldsymbol{y}^{<i}$.

We analyze A. If $y = G(z,x)$, then y is generated in polynomial time from a program with description length $O(|z|+|x|) = O(n)$. Hence $\mathrm{q}^t(y) \leq O(n)$ for all large enough t. By the chain rule,

$$\sum_{i=1}^{n} \mathrm{q}^{\tau(t)}(y^i \mid \boldsymbol{y}^{<i}) \;\leq\; \mathrm{q}^t(y) + n \cdot c\log t \;\leq\; n \cdot O(\log t).$$

Therefore, there exists an index $i \in [n]$ with $\mathrm{q}^{\tau(t)}(y^i \mid \boldsymbol{y}^{<i}) \leq O(\log t)$, equivalently $\mathrm{Q}^{\tau(t),\boldsymbol{y}^{<i}}(y^i) \geq 1/\mathsf{poly}(t)$. With polynomially many samples from $\mathrm{Q}^{\tau(t),\boldsymbol{y}^{<i}}$, the value y^i appears with noticeable probability, so A accepts with noticeable probability.

If y is uniform in $\{0,1\}^{n^2}$, then each block y^i is uniform in $\{0,1\}^n$ independently of $\boldsymbol{y}^{<i}$. For any fixed i and any fixed prefix $\boldsymbol{y}^{<i}$, the probability that a uniform y^i matches one of polynomially many samples drawn from $\mathrm{Q}^{\tau(t),\boldsymbol{y}^{<i}}$ is at most $\mathsf{poly}(n)/2^n$, which is negligible. A union bound over $i \in [n]$ shows that A accepts a truly random y only with negligible probability.

Thus A distinguishes $G(z,x)$ from uniform for every auxiliary input z, which contradicts the security of G. Hence no auxiliary-input one-way function exists. □

D Estimating Statistical Distance

We prove the following technical lemma, building on the ideas from [NR06].

Lemma (Restatement of Lemma 9.1)**.** *If there is no auxiliary-input one-way function, then there exists a polynomial-time randomized algorithm $\tilde{\Delta}$ such that for every description of circuits D_0 and D_1 and every $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$,*

$$\Pr_{\tilde{\Delta}}\left[\tilde{\Delta}(D_0,D_1;1^{\epsilon^{-1}},1^{\delta^{-1}}) \in [\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)-\epsilon, \Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)+\epsilon]\right] \geq 1-\delta,$$

where $\mathcal{D}_0$ (resp. $\mathcal{D}_1$) is a distribution of $D_0(r)$ (resp. D_1) for a random seed r.

Proof. We define the following auxiliary-input function f_{D_0,D_1} that takes a description of a pair of circuits D_0 and D_1 as auxiliary input:

$$f_d(b,r) = D_b(r),$$

where $b \in \{0,1\}$, and $r \in \{0,1\}^*$ is the input to D_0 and D_1 (without loss of generality, we assume that the lengths of input to D_0 and D_1 are the same with a proper truncation).

Since we assume that there is no auxiliary-input one-way function, $\{f_{D_0,D_1}\}_{D_0,D_1}$ is not one-way. Furthermore, based on the hashing technique developed in [IL89], we can also simulate uniform

sampling from $f^{-1}_{D_0,D_1}(y)$ on average over $y = f_{D_0,D_1}(b,r)$ with small statistical error, where $b \sim \{0,1\}$ and r is a randomly selected input. Notice that the distribution of y is statistically equivalent to $y \sim \mathcal{D}_b$ for $b \sim \{0,1\}$. Thus, for a given $y \sim \mathcal{D}_b$, where $b \sim \{0,1\}$ is a secret bit chosen at random, we can construct a natural predictor for b that empirically examines which label tends to be occur from the (approximated) uniform sampling from $f^{-1}_{D_0,D_1}(y)$. Based on this idea, Naor and Rothblum [NR06] proved that the predictor can correctly predict the secret b with probability roughly $\frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)}{2}$, resulting in the following claim.

Claim D.1 ([NR06, Lemma 4.2])**.** *If there is no auxiliary-input one-way function, then there exists a polynomial-time algorithm A such that for every pair of circuits D_0 and D_1 and every $\epsilon^{-1} \in \mathbb{N}$,*

$$\Pr_{b\sim\{0,1\},y\sim\mathcal{D}_b,A}[A(y;D_0,D_1,1^{\epsilon^{-1}}) = b] \geq \frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)}{2} - \frac{\epsilon}{2}.$$

By contrast, we can easily observe that $\frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)}{2}$ is the best possible success probability even in the statistical case.

Claim D.2. *For all distributions $\mathcal{D}_0, \mathcal{D}_1$ and all boolean-valued randomized functions f,*

$$\Pr_{b\sim\{0,1\},y\sim\mathcal{D}_b,f}[f(y) = b] \leq \frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)}{2}.$$

Proof. The claim is verified as the following simple calculation:

$$\begin{aligned}
\Pr_{b,y,f}[f(x) = b] &= \sum_y \left(\frac{1}{2}\mathcal{D}_1(y)\Pr_f[f(x)=1] + \frac{1}{2}\mathcal{D}_0(y)\Pr_f[f(x)=0]\right) \\
&= \sum_y \left(\frac{1}{2}\mathcal{D}_0(y) + \frac{1}{2}\left(\mathcal{D}_1(y) - \mathcal{D}_0(y)\right)\Pr_f[f(x)=1]\right) \\
&= \frac{1}{2} + \frac{1}{2}\left(\Pr_{y\sim\mathcal{D}_1,f}[f(y)=1] - \Pr_{y\sim\mathcal{D}_0,f}[f(y)=1]\right) \\
&\leq \frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)}{2}.
\end{aligned}$$

◇

Now, we present the construction of $\tilde{\Delta}$. Let A be the algorithm in Claim D.1. Given descriptions of D_0 and D_1 and parameters $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, the algorithm $\tilde{\Delta}$ empirically estimates the probability that $A(y;D_0,D_1,1^{2\epsilon^{-1}}) = b$ for $b \sim \{0,1\}, y \sim \mathcal{D}_b$ within accuracy error $\epsilon/4$ and confidence error δ. Let $\tilde{p} \in [0,1]$ be the estimated value. Then $\tilde{\Delta}$ outputs $2\tilde{p} - 1$ as the estimation of $\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)$.

By Hoeffiding's inequality, the empirical estimation above is accomplished by examining how many times the event $A(D_b(r);D_0,D_1,1^{2\epsilon^{-1}}) = b$ occurs in $M = O(\epsilon^{-2}\log\delta^{-1})$ trials for fresh seeds b and r. Thus, $\tilde{\Delta}$ halts in polynomial time in ϵ^{-1}, δ^{-1}, and the description size of D_0 and D_1 (notice that the length of each sample is bounded by the description size of D_0 and D_1).

We verify the correctness. Let $p = \Pr_{b\sim\{0,1\},y\sim\mathcal{D}_b,A}[A(y;D_0,D_1,1^{\epsilon^{-1}}) = b]$. With probability at least $1-\delta$, the empirical estimation is successfully performed, i.e., $|\tilde{p} - p| \leq \epsilon/4$ holds. We observe that $2\tilde{p} - 1 \in [\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1) \pm \epsilon]$ in this case.

From Claim D.2,

$$\tilde{p} \leq p + \frac{\epsilon}{4} \leq \frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1)}{2} + \frac{\epsilon}{4};$$

thus, $2\tilde{p} - 1 \leq \Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1) + \epsilon/2 < \Delta_{\mathrm{tv}}(\mathcal{D}_0,\mathcal{D}_1) + \epsilon$.

By contrast, from Claim D.1,

$$\tilde{p} \geq p - \frac{\epsilon}{4} \geq \frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0, \mathcal{D}_1)}{2} - \frac{\epsilon}{4} - \frac{\epsilon}{4} = \frac{1}{2} + \frac{\Delta_{\mathrm{tv}}(\mathcal{D}_0, \mathcal{D}_1)}{2} - \frac{\epsilon}{2};$$

thus, $2\tilde{p} - 1 \geq \Delta_{\mathrm{tv}}(\mathcal{D}_0, \mathcal{D}_1) - \epsilon$, as desired. □

E Robustness of Our Assumption

In this section, we show the robustness of our assumption $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP.

First, we introduce some auxiliary preliminaries.

Definition E.1 (MINKT)**.** *For a function $\sigma\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, the problem* $\mathrm{MINKT}[\sigma(n,t)]$ *is defined as*

$$\mathrm{MINKT}[\sigma(n,t)] = \{(x, 1^t) : \mathrm{K}^t(x) \leq \sigma(|x|, t)\}.$$

Let $\mathcal{U}^* = \{\mathcal{U}_{\langle n,t\rangle}\}_{n,t\in\mathbb{N}}$ be a family of distributions, where each $\mathcal{U}_{\langle n,t\rangle}$ is a distribution over $(x, 1^t)$ for $x \sim \{0,1\}^n$.

Definition E.2 ($\mathrm{GapK^t}$-vs-$\mathrm{pK^{poly}}$)**.** *For $c \geq 0$ and a polynomial τ, the problem* $\mathrm{Gap}_c\mathrm{K^t}$-vs-$\mathrm{pK}^\tau = (\Pi_{\mathrm{yes}}, \Pi_{\mathrm{no}})$ *is defined as follows:*

$$\Pi_{\mathrm{yes}} := \{(x, 1^s, 1^t) : \mathrm{K}^t(x) \leq s\},$$
$$\Pi_{\mathrm{no}} := \{(x, 1^s, 1^t) : \mathsf{pK}^{\tau(t)}(x) > s + c\log(t|x|)\}.$$

We write $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP *to denote that* $\mathrm{Gap}_c\mathrm{K^t}$-vs-$\mathrm{pK}^\tau \in$ pr-BPP *for some constant $c \geq 0$ and polynomial τ.*

The main assumption $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP is robust in the following sense.

Theorem E.1. *The following are equivalent:*

1. $(\mathrm{MINKT}[n - c\log(nt)], \mathcal{U}^*) \in \mathsf{Avg}_{1/p}\mathsf{BPP}$ *for some constant $c > 0$ and some polynomial p.*
2. $\mathrm{GapK^t}$-vs-$\mathrm{pK^{poly}} \in$ pr-BPP.
3. $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP.

Proof sketch. The implication from Item 1 to Item 2 follows from Hirahara's worst-case-to-average-case reduction; we refer the reader to [Hir20b]. The implication from Item 2 to Item 3 follows from the trivial inclusion of NO instances, which holds since $\mathrm{K}(x) \leq \mathsf{pK}^{\tau(t)}(x) + O(\log t)$ for any polynomial τ (see [GKLO22, Lemma 18]).

Thus, it remains to show the implication from Item 3 to Item 1. Assume Item 3. By Proposition 4.3, there is a (randomized) algorithm $\tilde{K}$ such that, with high probability,

$$\mathrm{K}(x) \;\leq\; \tilde{K}(x, 1^t) \;\leq\; \mathrm{K}^t(x) + c\log(|x|t)$$

for some constant c.

We construct an algorithm A that places $(\mathrm{MINKT}[n - 2c\log(nt)], \mathcal{U}^*)$ in $\mathsf{Avg}_{1/\mathsf{poly}}\mathsf{BPP}$ as follows. Given an instance $(x, 1^t)$, the algorithm A outputs 0 if $\tilde{K}(x, 1^t) > |x| - c\log(|x|t)$, and outputs $\perp$ otherwise.

We check that A is errorless and succeeds with inverse-polynomial probability over $(x, 1^t) \sim \mathcal{U}^*$. First, A never errs on NO instances since it never outputs 1. Hence it suffices to show: (i) on YES instances, A outputs $\perp$ with high probability; and (ii) under $x \sim \{0,1\}^n$, the probability that A outputs $\perp$ is at most $1/\mathsf{poly}(nt)$.

For (i), let $(x, 1^t)$ be a YES instance. Then, with high probability over the randomness of $\tilde{K}$,

$$\tilde{K}(x, 1^t) \ \leq \ \mathrm{K}^t(x) + c\log(|x|t) \ \leq \ |x| - c\log(|x|t),$$

so A outputs $\perp$.

For (ii), note that if $\mathrm{K}(x) > |x| - c\log(|x|t)$, then

$$\tilde{K}(x, 1^t) \ \geq \ \mathrm{K}(x) \ > \ |x| - c\log(|x|t),$$

and hence A outputs 0 with high probability. Therefore, A outputs $\perp$ with high probability only when $\mathrm{K}(x) \leq |x| - c\log(|x|t)$. By the standard counting argument, the fraction of such $x \in \{0,1\}^n$ is at most $1/\mathsf{poly}(nt)$, which proves (ii). □

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