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

Complexity-Theoretic Inductive Inference

Shuichi Hirahara* Mikito Nanashima†

Abstract

Inductive inference, introduced by Solomonoff (Information and Control, 1964), is a foundational concept in knowledge acquisition, formulated as the task of extrapolating a sequence of symbols. In his seminal work, Solomonoff established a fundamental theorem for *recursion-theoretic universal* inductive inference, applicable to sequences generated by all Turing machines, based on the (uncomputable) task of computing Kolmogorov complexity.

In this work, we present a *complexity-theoretic counterpart* to Solomonoff's theorem: we construct an *efficient* universal inductive inference algorithm, applicable to sequences *efficiently* generated by all randomized Turing machines, assuming that *time-bounded* Kolmogorov complexity can be efficiently approximated. This assumption is known to follow from the average-csae easiness of NP. As corollaries, we obtain various worst-case learning algorithms, such as distributional learning and learning adaptively changing distributions, under the assumption that NP is easy on average.

Our inductive inference algorithm and its analysis build on techniques developed in meta-complexity theory. At a high level, the algorithm divides a given sequence of symbols into two parts, "advice" and "context", and extrapolates the next symbol of the context according to the time-bounded universal distribution given the advice. We prove that this algorithm avoids computationally deep strings, which are hard instances for every average-case 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 his seminal work, known as Solomonoff's theory of inductive inference, he formulated the problem as the extrapolation of a long sequence of symbols: given a sequence $x_1, x_2, \ldots, x_{i-1}$, the task is to extrapolate the next symbol x_i for each $i \in \mathbb{N}$. This kind 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 to recognize danger in their environment. All of these can be viewed as forms of inductive inference: using observed data to predict the future. In this sense, developing an algorithmic foundation for inductive inference is one of the most significant challenges in science.

Solomonoff's theory presents an elegant reduction of inductive inference to the task of computing Kolmogorov complexity. Here, the *Kolmogorov complexity* $\mathrm{K}(x)$ of a string x is defined as the length of a shortest program that prints x. Solomonoff proposed an inference algorithm that, given a sequence $\bar{x} = (x_1, \ldots, x_{i-1})$, extrapolates the next symbol x_i with probability proportional to $2^{-\mathrm{K}(\bar{x}, x_i)}$. This algorithm provides a mathematical formalization of Occam's Razor: simpler explanations (i.e., those with lower Kolmogorov complexity) are assigned higher probabilities. In other words, sequences that can be generated by shorter programs are considered more likely continuations. This approach captures the intuition that patterns with concise descriptions should generalize better than those requiring lengthy explanations. Mathematically, the algorithm of Solomonoff has the property that, given as input the first $(i-1)$ symbols of a sequence $x_1, x_2, \ldots$ generated by a randomized program of description length s, it can output a symbol, whose Kullback–Leibler divergence (and statistical distance) to the next symbol x_i is small for most choices of i, provided that $i \gg s$ [LV19, Claim 5.2.2]. Because of its strong property, Solomonoff's inductive inference profoundly influenced subsequent research in the field, especially as a basis of the theory of universal artificial intelligence [see, e.g., MF98; Hut05; HQC24].

There is one major caveat in Solomonoff's original approach, as he himself noted [Sol64a, §3.1.2.1]: the inference algorithm is not even theoretically implementable, because $\mathrm{K}(\cdot)$ is uncomputable. To address this, he proposed an approximate version using 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 defined to be the length of a shortest program that prints x in time t. This brings Solomonoff's theory into the *complexity-theoretic* realm, and raises the following natural questions: What is the computational complexity of performing inductive inference in a time-bounded setting? Is it efficiently reducible to the task of computing $\mathrm{K}^t(\cdot)$?

2 Our Results

In this paper, we present an efficient algorithm for Solomonoff's inductive inference in a time-bounded setting, under the assumption that time-bounded Kolmogorov complexity is efficiently computable (on average). This establishes a complexity-theoretic counterpart to the original theory of Solomonoff's inductive inference, laying the groundwork for a theory of complexity-theoretic universal inductive inference.

Specifically, our inductive inference algorithm is based on the assumption that "GapK^{t}-vs-K $\in$ pr-BPP," which is equivalent to the existence of 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 $\widetilde{K}$.[1] In fact, 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, follows from the average-case easiness of NP (DistNP $\subseteq$ AvgBPP). Under such a weak assumption, we present an efficient (worst-case) algorithm for inductive inference.

Theorem 2.1. *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exists a polynomial-time randomized algorithm* L *such that for every* $\epsilon, \delta \in (0,1]$ *and every randomized Turing machine* Π *described by an* s*-bit string that generates at least* m *binary strings* $x_1, \ldots, x_m \in \{0,1\}^n$ *in time* t*, the following holds: if* $m \geq O(s\epsilon^{-2}\delta^{-1})$*, then for a uniformly random round* i *chosen from* $[m] := \{1, \ldots, m\}$*, we have*

$$\Pr_{i, x_1, \ldots, x_{i-1}} \left[\Delta_{\mathrm{tv}}\left(L\left(x_1, \ldots, x_{i-1}\right), \left(\mathcal{X}_i | x_{<i}\right)\right) \leq \epsilon\right] \geq 1 - \delta,$$

where $\Delta_{\mathrm{tv}}(\cdot,\cdot)$ *denotes total variation distance, and* $\mathcal{X}_i|x_{<i}$ *is the conditional distribution of the* i*-th symbol* x_i *given the preceding symbols* $x_1, \ldots, x_{i-1}$*. The parameters* $s, t, m, \epsilon^{-1}, \delta^{-1}, t^{1/\delta}$ *are given to* L *in unary.*

In other words, our inference algorithm can efficiently extrapolate a sequence of symbols generated by an unknown randomized t-time program of size s with small statistical error on almost all rounds, provided that the total number of rounds m is sufficiently larger than s. The running time of L is at most a polynomial in ϵ^{-1} and $t^{1/\delta}$; in particular, for a constant δ, the running time is $\mathsf{poly}(t/\epsilon)$.

Prior to this work, it was well understood that the complexity of *average-case* inductive inference is equivalent to the non-existence of one-way functions [IL90; HN23] as well as the task of computing time-bounded Kolmogorov complexity on average (in the sense of error-prone average-case complexity) [LP20]. Impagliazzo and Levin [IL90] presented the ideas for showing the equivalence between the existence of efficient average-case algorithms for inductive inference and the non-existence of a one-way function. Hirahara and Nanashima [HN23] implemented the ideas of Impagliazzo and Levin and presented an inductive inference algorithm that works for most choices of randomized Turing machines under the non-existence of one-way functions. Liu and Pass [LP20] showed that this assumption is equivalent to average-case easiness of time-bounded Kolmogorov complexity. Thus, the results of [IL90; HN23] can be regarded as a reduction from average-case inductive inference to the task of computing time-bounded Kolmogorov complexity on average. Hirahara and Nanashima observed that various learning tasks studied in the literature are special cases of inductive inference, and consequently, obtained a number of average-case learning algorithms as corollaries.

The significance of Theorem 2.1 is that the algorithm is "worst-case", i.e., it works for *every* randomized Turing machine Π, despite that the assumption is weaker than the average-case easiness of NP. As corollaries, we obtain various worst-case learning algorithms in Heuristica, which is a hypothetical world in which NP is easy on average but is hard in the worst case [Imp95]. We discuss details in Sections 2.1 and 2.2. (Readers interested in understanding the inference algorithm first may wish to skip ahead to Section 3.)

2.1 Learning Distributions in Heuristica

Distributional learning, introduced by Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie [KMRRSS94], is one of the standard theoretical frameworks for unsupervised learning. In this setting, a learner aims to approximate an unknown distribution $\mathcal{D}$ over strings, given only independent

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

and identically distributed (i.i.d.) samples from $\mathcal{D}$. Specifically, the task of the learner is to output a sampler whose output distribution is statistically close to the target distribution $\mathcal{D}$, given polynomially many samples from $\mathcal{D}$ (see Definition 7.3 for a formal definition).

Despite the naturalness and centrality of learning distributions in machine learning, the computational complexity of this task remains elusive, especially in contrast to the well-studied case of supervised learning. Prior work has established a *necessary* condition: any learner for distributions generated by polynomial-time randomized machines with polynomial-length descriptions must be able to break the security of (auxiliary-input) one-way functions [KMRRSS94; Xia10]. In contrast, we present the first *sufficient* condition for this task (apart from a trivial sufficient condition of $\mathsf{NP} \subseteq \mathsf{BPP}$).

Corollary 2.2. *If* $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$*, then every distribution efficiently sampled by randomized machines with polynomial-length descriptions (i.e., randomized polynomial-size circuits) is distributionally learnable in polynomial time. In particular, any distribution over n-bit strings whose polynomial-time sampler is described using $s(n)$ bits can be learned with accuracy error ϵ and confidence error δ using $O(s(n) \cdot \epsilon^{-2} \log \delta^{-1})$ i.i.d samples.*

Here, distributional learning with accuracy error ϵ and confidence error δ means that, with probability at least $1 - \delta$ (over the choice of training samples and the learner's randomness), the learner outputs a sampler whose distribution is within total variation distance at most ϵ from the target distribution $\mathcal{D}$. Polynomial-time learnability means that the learner halts in time polynomial in the sample size and in ϵ^{-1} and δ^{-1}.

Notably, the exponential dependence on the confidence parameter δ in the time complexity of Theorem 2.1 is eliminated, and the dependence in the sample complexity is improved from δ^{-1} to $\log \delta^{-1}$. This improvement leverages the independence of the i.i.d. samples, similar to techniques from standard supervised learning theory [HKLW88]. Specifically, we execute the inference algorithm with *constant* confidence for $O(\log \delta^{-1})$ repetitions.

Next, we consider a more general setting of learning known as *adaptively changing distributions* (ACDs), introduced by Naor and Rothblum [NR06]. Informally, an ACD is a distribution with a hidden state, and whenever a sample is drawn from the ACD, its internal state is updated. The task of learning ACDs is to predict a next sample from an unknown ACD, given polynomially many samples.

In fact, it is easy to observe that their model of learning ACDs is a special case of our inductive inference framework (see Section 7.1 for formal definitions). In the ACD setting, the learner only needs to output an accurate prediction at a single round of its choosing. By contrast, the inference algorithm in Theorem 2.1 must perform well on almost all rounds, with no control over which ones are selected. Thus, learnability for ACDs immediately follows from Theorem 2.1.

Corollary 2.3. *If* $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$*, then for every constant confidence error δ, all polynomial-time samplable ACDs of s-bit initial state (which constitute the target of learning) are learnable in polynomial time within $O(s \cdot \epsilon^{-2})$ rounds for any accuracy error ϵ.*

Previously, it was shown that ACDs are learnable *on average* over initial states under the assumption that one-way functions do not exist [NR06; HN23]. In contrast, Corollary 2.3 provides the first sufficient condition for learning ACDs in the *worst case* over all initial states (but only for constant confidence error).

2.2 Expected Loss Minimization in Heuristica

The second application is supervised learning, where the learner is given labeled data and asked to predict labels in a specified manner, typically by minimizing expected loss.

We begin with one of the most standard and extensively studied frameworks: the Probably Approximately Correct (PAC) model, introduced by Valiant [Val84]. In this model, the label of an example $x \in \{0,1\}^n$ is determined by an unknown target function $f\colon \{0,1\}^n \to \{0,1\}$, and each sample takes the form $(x, f(x))$, where x is independently drawn from an unknown distribution $\mathcal{D}$, referred to as the example distribution. The learner receives i.i.d. samples $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ and is required to produce an efficiently computable hypothesis h such that $\Pr_{x\sim\mathcal{D}}[h(x) \neq f(x)] \leq \epsilon$, where ϵ is a given accuracy parameter. This condition must be satisfied with high probability over the learner's randomness and the training samples.[2]

Previously, a line of work [HN21; GKLO22; GK23] developed a method for learning binary labels from i.i.d. samples under the same assumption that NP is easy on average. Their approach yields a learner for polynomial-time computable target functions of description size $s(n)$ under any example distribution samplable with $a(n)$-bit advice, achieving a sample complexity of $\tilde{O}((s(n)+a(n))^3\epsilon^{-8})$. In contrast, we drastically reduce the sample complexity to $O((s(n)+a(n))\epsilon^{-2})$ in the PAC setting, even when each label is independently flipped with some noise.

Corollary 2.4. *If* $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$*, then for every polynomial* $s(n)$ *and* $a(n)$*, the class of polynomial-time computable functions described using* $s(n)$ *bits is PAC learnable in polynomial time under all example distributions samplable with* $a(n)$*-bit advice. The required sample complexity is at most* $O((s(n)+a(n))\epsilon^{-2})$ *for accuracy parameter* $\epsilon \in (0,1]$*, even when each label is independently flipped with probability* $\eta \in (0, 1/2]$ *(where the hypothesis is required to agree with the label with probability at least* $1-(\eta+\epsilon)$*).*

In the presence of random noise, the sample complexity above is statistically tight, as it is known that learning labels produced by $s(n)$-bit programs under samplable distributions (i.e., when $a(n) = O(1)$) generally requires $\Theta(s(n)\epsilon^{-2})$ samples [see HN23, Appendix A].

Notably, the prior learnability results in [HN21; GKLO22; GK23] are based on a different strategy: they use average-case feasibility to achieve weak prediction (i.e., performing slightly better than random guessing), and then boost the accuracy. In contrast, our approach is based on inductive inference. Interestingly, the learnability results following from two approaches appear fundamentally different and incomparable within Heuristica. For instance, although the prior method incurs a significantly higher sample complexity overhead, it can handle more general noise models that fall outside the reach of our current framework.

By contrast, we emphasize that a key advantage of our inductive inference-based framework lies in its generality beyond conventional binary classification. Our result naturally extends to multi-class classification, other loss functions, and even learning in *non-i.i.d. settings*. This generality arises from the fact that the underlying inference algorithm in Theorem 2.1 remains effective even in the presence of correlations among the data.

In particular, consider the case where the learner sequentially receives a stream of observations $(x_1, y_1), (x_2, y_2), \ldots \in \{0,1\}^n \times \{0,1\}^n$, where each x_i represents preknowledge available for prediction and y_i represents the outcome of interest. For ease of exposition, we assume here that the preknowledge and outcome lie in the same domain, though this assumption can be relaxed in the formal result. The objective of the learner is specified by a set of actions A_n and a loss function $\ell_n\colon \{0,1\}^n \times A_n \to \mathbb{R}_{\geq 0}$, where $\ell_n(y, a)$ denotes the loss incurred when the learner takes action a (before observing the outcome) and the actual outcome is y. At each round i, the learner observes the data seen so far and selects a decision rule $h_i\colon \{0,1\}^n \to A_n$ to minimize the expected loss:

$$\mathsf{loss}_i(h_i) := \mathrm{E}_{(x_i,y_i)}\left[\ell(y_i, h_i(x_i))\right]$$

[2] We omit the discussion of the confidence error δ in this overview, as any constant confidence error can be reduced to an arbitrary $\delta \in (0,1]$ by repeating the learning process $O(\log \delta^{-1})$ times [HKLW88].

Note that in the special case of i.i.d. binary-labeled samples, this setting recovers standard PAC learning by taking $A_n = \{0,1\}$ and using the 0-1 loss function $\ell(y,a) = \mathbb{1}\{y \neq a\}$.

In what follows, we consider the case where the loss function $\ell = \{\ell_n\}$ is γ-bounded for some function $\gamma\colon \mathbb{N} \to \mathbb{N}$, meaning that $\ell_n(y,a) \leq \gamma(n)$ for all $n \in \mathbb{N}$, $y \in \{0,1\}^n$, and $a \in A_n$. We show that our loss minimization algorithm can, at almost all rounds, make decisions whose expected loss approximates the minimum possible, provided that the total number of rounds is sufficiently large relative to the description size of the machine generating the data stream.

Corollary 2.5. *If* $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$*, then for every polynomial-size action sets* $\{A_n\}_{n\in\mathbb{N}}$ *and every* γ*-bounded loss function, there exist a polynomial-time randomized algorithm* P *such that the following holds: For every* $\epsilon, \delta \in (0,1]$*, and every randomized program described using* s *bits that generates at least* m *samples* $(x_1,y_1),\ldots,(x_m,y_m),\ldots \in \{0,1\}^n \times \{0,1\}^n$ *in time* t*, if* $m \geq O(s\epsilon^{-2}\delta^{-1}\gamma(n)^2)$*, then for a randomly chosen round* $i \sim [m]$*, we have*

$$\Pr_{i,x_1,y_1,\ldots,x_{i-1},y_{i-1}}\left[P(x_1,y_1,\ldots,x_{i-1},y_{i-1}) \text{ outputs } h_i \text{ s.t. } \mathsf{loss}_i(h_i) \leq \min_{f_i^* \in \mathcal{F}_n} \mathsf{loss}_i(f_i^*) + \epsilon\right] \geq 1-\delta,$$

where $\mathcal{F}_n$ *is the set of all functions* $f\colon \{0,1\}^n \to A_n$ *The parameters* $s, t, m, \epsilon^{-1}, \delta^{-1}, t^{\delta^{-1}}$ *are provided to* P *in unary.*

We note that the corollary also applies to exponential-size action sets, as long as approximate expected loss minimization can be performed efficiently given i.i.d. samples from any distribution over labels (see Definition 8.1 for the formal requirements).

This form of sequential universal decision-making is a central component in the theory of *universal artificial intelligence* [Hut05; HQC24], which has primarily been developed in the statistical setting. To the best of our knowledge, this is the first result to advance the understanding of the complexity-theoretic requirements for enabling such sequential universal decision-making, going beyond the implicit and somewhat trivial baseline assumption $\mathsf{NP} \subseteq \mathsf{BPP}$ (unless Heuristica is ruled out).

3 Technical Overviews

We now present ideas behind the inductive inference algorithm of Theorem 2.1. At a high level, the algorithm works as follows: Given a sequence of symbols $x_1,\ldots,x_{i-1}$, it partitions them into two parts, "advice" $z = (x_1,\ldots,x_j)$ and "context" $c = (x_{j+1},\ldots,x_{i-1})$ for some j, and then extrapolates the next symbol x_i of the context c according to the time-bounded universal distribution given advice z. To explain the details, we review the notion of Kolmogorov complexity, time-bounded universal distribution, and universal extrapolation [IL90; HN23] in Section 3.1.

3.1 Algorithmic Information and Universal Extrapolation

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.

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.

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 $\mathrm{q}^t(x \mid z) \coloneqq -\log \Pr[x \leftarrow \mathrm{Q}^{t,z}]$, where $\Pr[x \leftarrow \mathrm{Q}^{t,z}]$ 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 $\mathrm{qcd}^t(x \mid z) \coloneqq \mathrm{q}^t(x \mid z) - \mathrm{K}(x \mid z)$.[3]

Universal Extrapolation. Following [HN23], we formalize the notion of universal extrapolation. For each $k, t \in \mathbb{N}$ and $x, z \in \{0,1\}^*$, we define $\mathsf{Next}_k(x; \mathrm{Q}^{t,z})$ as the conditional distribution over the k-bit prefix of a continuation of x, sampled according to $\mathrm{Q}^{t,z}$. (If x is not in the support of $\mathrm{Q}^{t,z}$, we treat it as a distribution over the empty string.) More formally, $\mathsf{Next}_k(x; \mathrm{Q}^{t,z})$ is a distribution over $\{0,1\}^{\le k}$. The probability that $y \in \{0,1\}^{\le k}$ is sampled according to $\mathsf{Next}_k(x; \mathrm{Q}^{t,z})$ is defined as the probability, over w sampled from $\mathrm{Q}^{t,z}$, 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. *Universal extrapolation* [IL90; HN23] refers to the task of sampling from $\mathsf{Next}_k(x; \mathrm{Q}^t)$ given "context" x as input. Although x plays the role of a "condition" in the sense of probability, we refer to it as the *context* of universal extrapolation in order to avoid confusion with the notion of conditioning in Kolmogorov complexity.

3.2 Our Inference Algorithm

Our inference algorithm is obtained by carefully choosing advice $z = (x_1, \ldots, x_j)$ and context $c = (x_{j+1}, \ldots, x_{i-1})$ and performing the universal extrapolation with context c and advice z (i.e., sampling from $\mathsf{Next}_k(c; \mathrm{Q}^{t,z})$). Its performance will be guaranteed by two key technical lemmas, explained below.

To motivate the design of our algorithm, we begin by examining the limitations of previous approaches based on average-case inversion (i.e., the task of inverting one-way functions) [IL90; HN23]. As explicitly stated in [HN23, Theorem 8.1], their inference algorithm can actually sample from $\mathsf{Next}_k(x; \mathrm{Q}^t)$ for *any* input x. However, the running time of the algorithm is *exponential* in the computational depth $\mathrm{qcd}^t(x)$ of the input x. In fact, the exponential running time in the computational depth of inputs is known to characterize the running time of an average-polynomial-time algorithm with respect to arbitrary polynomial-time samplable distributions [AF09]. Unfortunately, in the worst case, the computational depth of sequences generated by some randomized machine of description size s can be as large as $\Omega(s)$, and thus the algorithm of [HN23] is inefficient.

A key insight that allows us to bypass this limitation comes from the *slow growth law* [Ben88; AFPS12; Hir23] (see also Lemma 4.16), which implies that for any t_Π-time randomized Turing

[3] This notion was denoted by cd in [HN23], whereas we use qcd here to distinguish it from other forms of computational depth based on different complexity measures.

machine of size s, if t is taken to be sufficiently larger than t_Π, then with high probability, the computational depth of the output sequence $(x_1, \ldots, x_m)$ of the Turing machine is at most s.[4] If the output sequence is divided into more than $m \geq s/\log t$ blocks and the s-bit computationally deep information is "decomposed" across them, then we may hope that the *amortized* computational depth per block would become $s/m \leq \log t$. However, simply partitioning $(x_1, \ldots, x_m)$ into m blocks $\{x_i \mid i \in [m]\}$ does not decrease the amortized computational depth $\mathrm{E}_{i\sim[m]}[\mathrm{qcd}^t(x_i)]$, for example in the case where all $x_1 = \cdots = x_m$ are computationally deep.

Instead, we consider the computational depth of each block *given all previous blocks as advice*. Intuitively, this ensures that the computational depth in each round reflects only the new information not already captured, suggesting that $\sum_{i=1}^m \mathrm{qcd}^t(x_i \mid x_1, \ldots, x_{i-1})$ would be bounded by roughly $\mathrm{qcd}^t(x_1, \ldots, x_m)$. Our first lemma justifies this, under the assumption that $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP.

Lemma 3.1 (Averaging Lemma for Computational Depth; see also Lemma 6.2)**.** *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then there exists a polynomial* τ *such that for every* $n, m, t, \delta^{-1} \in \mathbb{N}$*, and every randomized Turing machine* Π *described in* s $(\geq \log \delta^{-1})$ *bits that produces* $x_1, \ldots, x_m \in \{0,1\}^{\leq n}$ *in time* t_Π*, if* $t \geq \tau(t_\Pi)$ *and* $m \geq O(s/\log t)$*, then*

$$\mathrm{E}_{i\sim[m],x_1,\ldots,x_i}\left[\mathrm{qcd}^t(x_i \mid x_1, \ldots, x_{i-1})\right] \leq O(\log t).$$

In particular, by Markov's inequality,

$$\Pr_{i\sim[m],x_1,\ldots,x_i}\left[\mathrm{qcd}^t(x_i \mid x_1, \ldots, x_{i-1}) \leq \delta^{-1} \cdot O(\log t)\right] \geq 1 - \delta.$$

This lemma motivates the core idea behind our inductive inference strategy. In order to avoid computationally deep strings (i.e., strings with high computational depth), we reduce the "context window" (the length of the context) of universal extrapolation by treating all the preceding blocks as advice.

To implement this idea, two key questions remain: (i) how can we construct an efficient algorithm that performs universal extrapolation given an arbitrary advice string, and (ii) does universal extrapolation with a restricted context still preserve the universality guaranteed by Solomonoff's theory?

The first question has been addressed in prior work through a simple observation. The extrapolation algorithm developed in [HN23] naturally relativizes to any advice string. This yields a reduction from universal extrapolation under arbitrary advice strings to the task of inverting a family of polynomial-time computable functions $\{f_z\}_{z\in\{0,1\}^*}$, where a single algorithm must work for every z but only on average over the random input of f_z. This task is often referred to as inverting *auxiliary-input functions* in the literature, following the notion of *auxiliary-input one-way functions* introduced by Ostrovsky and Wigderson [OW93]. It is known that such inversion is efficiently reducible to $\mathrm{GapK}^{\mathrm{t}}$-vs-K [HS17]. As a result, we obtain a universal extrapolation algorithm UE, which, given parameters $n, t \in \mathbb{N}$ (in unary), a context x, and advice z, samples from $\mathsf{Next}_n(x; \mathrm{Q}^{t,z})$ in exponential time in $\mathrm{qcd}^t(x \mid z)$, with small statistical error (see Lemma 6.5 for the formal statement).

To answer the second question, we establish our second technical lemma. This result demonstrates that, with a properly large context and total size of blocks, the strategy of restricting the context and placing all other information in the advice does not compromise the universality of Solomonoff's inductive inference, assuming $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ BPP.

[4] This also follows from the definition of q or the optimal coding theorem for pK [LOZ22].

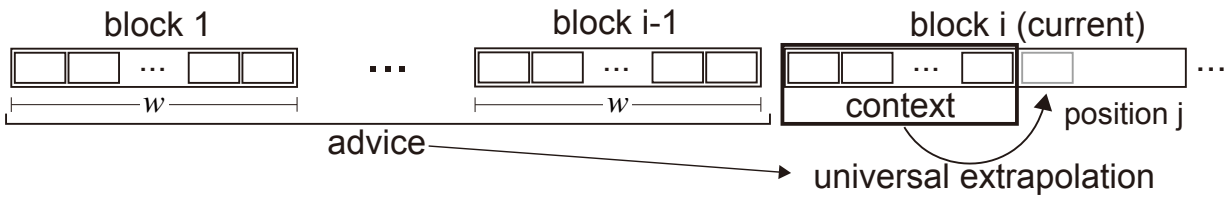

Figure 1: Our Inference Algorithm

Lemma 3.2 (Universal Extrapolation with Restricted Context)**.** *Assume* GapK^{t}-vs-K $\in$ pr-BPP*. Then there exists a polynomial p such that the following holds: For every $n, s, b, w \in \mathbb{N}$ and every randomized Turing machine Π described in s bits that produces at least $b \cdot w$ binary strings $x_1, \ldots, x_{bw}, \ldots \in \{0,1\}^n$ in time t_Π, define for each $i \in [b]$ the i-th block as $\bar{x}^i = x_{(i-1)w+1} \circ \cdots \circ x_{iw}$.*

Then for any $t \in \mathbb{N}$ satisfying $t \geq p(t_\Pi)$, and for $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, if $b \geq O(s/\log t)$ and $w \geq O(\epsilon^{-2}\delta^{-1}\log t)$, then

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

where $i \sim [b]$ is a index of the current block (treated by L as a context), $j \sim [w]$ in the position of extrapolation in the current block, $\bar{x}^{<i}$ is a concatenation of preceded blocks up to $i-1$, $\bar{x}^i_{<j}$ is a concatenation of preceded messages up to $j-1$ in the i-th block, and $X^i_j|\bar{x}^{\leq i}_{<j}$ represents the conditional distribution of $x_{(i-1)w+j}$ given preceded messages $\bar{x}^{\leq i}_{<j} = (\bar{x}^{<i}, \bar{x}^i_{<j})$.

The formal statement is given in terms of KL divergence (see Lemma 6.4); the above is a simplified version derived via Pinsker's inequality.

Final Description of the Inference Algorithm. The lemmas above yield our inference algorithm L, illustrated in Figure 1. The input sequence is partitioned into a series of blocks: each block consists of $w = O(\epsilon^{-2}\delta^{-1}\log t)$ strings, and the total number of blocks is at least $b = O(s/\log t)$. This arrangement requires $w \cdot b = O(s\epsilon^{-2}\delta^{-1})$ total rounds, matching the bound in Theorem 2.1.

At each prediction position, L identifies the block index i and the position j within it. It then performs universal extrapolation with:

- Context: the prefix up to position $j-1$ within block i;
- Advice: the full sequence of preceding blocks $1, \ldots, i-1$.

Lemma 3.2 guarantees prediction accuracy on average over all positions, while Lemma 3.1 ensures that the universal extrapolation runs in time $t^{\delta^{-1}}$ on average over the choice of block index.

Necessity of the Universal Extrapolation Framework. One might wonder whether extrapolation is essential for our inference — why not simply sample from the time-bounded universal Turing machine using the prior messages as advice? Unfortunately, this naive strategy fails to yield accuracy better than a universal constant.

The issue lies in the fact that the universal distribution with advice dominates the one without, due to the existence of programs that simply ignore the advice. Concretely, for any advice string and $x \in \{0,1\}^n$, the t-time-bounded universal distribution assigns probability at least $2^{-n}/c$ to x (for some universal constant c when $t \geq O(n)$), since there exists a trivial program of length

$|x| + O(1)$ that simply outputs x by embedding it directly. Thus, even in extremely simple cases where a program deterministically outputs a fixed string for all rounds, the total variation distance from the universal distribution is lower bounded by a constant (depending on c).

Hence, universal extrapolation is essential to our inference result, as it enables us to overcome the inherent limitations imposed by this trivial domination.

3.3 Proof Overview of Technical Lemmas

Both Lemmas 3.1 and 3.2 are derived from the following inequality, which can be seen as a multi-block extension of the inequality known as the symmetry of information in algorithmic information theory[5].

Lemma 3.3. *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP, *then for every* $n, m, t \in \mathbb{N}$ *with* $t \geq O(m+n)$ *and every* $x_1, \ldots, x_m \in \{0,1\}^{\leq n}$, *it holds*

$$\sum_{i=1}^{m} \mathsf{pK}^{p(t)}(x_i \mid x_1, \ldots, x_{i-1}) \leq \mathsf{pK}^t(x_1, \ldots, x_m) + m \cdot O(\log t).$$

Here, pK denotes the *probabilistic* Kolmogorov complexity, introduced by Goldberg, Kabanets, Lu, and Oliveira [GKLO22] (see Section 4 for the formal definition). It is known that $\mathsf{pK}^t(\cdot \mid \cdot)$ and $\mathrm{q}^t(\cdot \mid \cdot)$ are equivalent up to an additive logarithmic term, with only polynomial overhead in time [HN23].

The proof of Lemma 3.3 builds on the techniques used in establishing the time-bounded symmetry of information (i.e., the two-block case) as developed in [Hir22; GK22]. However, a naive extension of these proofs to the multi-block setting leads to an additive error of $m^2 \cdot O(\log t)$ in the inequality, rather than the desired $m \cdot O(\log t)$. Such a bound is insufficient for our purposes, as it fails to yield a meaningful averaging argument on computational depth, as outlined below. To avoid the the quadratic dependence on m, we adapt the parameter tuning approach based on Hirahara's worst-case-to-average-case reduction [Hir18; Hir20b], in an *adaptive* manner across blocks. The proof ideas for Lemma 3.3 are deferred to Section 5, following the preliminaries.

We now focus on how Lemma 3.3 underlies the proofs of Lemmas 3.1 and 3.2. To highlight the conceptual structure and avoid cluttering the exposition with polynomial-time details, we use the shorthand poly to denote time bounds implicitly bounded by some fixed polynomial. Accordingly, we treat $\mathrm{q}^{\mathsf{poly}}(\cdot \mid \cdot)$ and $\mathsf{pK}^{\mathsf{poly}}(\cdot \mid \cdot)$ interchangeably in what follows.

Proof Sketch of Averaging Lemma (Lemma 3.1). We begin by applying Lemma 3.3, which gives the inequality:

$$\begin{aligned}
\sum_{i=1}^{m} \mathrm{q}^{\mathsf{poly}}(x_i \mid x_1, \ldots, x_{i-1}) &\leq \mathrm{q}^{\mathsf{poly}}(x_1, \ldots, x_m) + m \cdot O(\log t) \\
&\leq \mathrm{qcd}^{\mathsf{poly}}(x_1, \ldots, x_m) + \mathrm{K}(x_1, \ldots, x_m) + m \cdot O(\log t) \\
&\leq \mathrm{qcd}^{\mathsf{poly}}(x_1, \ldots, x_m) + \sum_{i=1}^{m} \mathrm{K}(x_i \mid x_1 \ldots, x_{i-1}) + m \cdot O(\log t).
\end{aligned}$$

[5]The same inequality was proved in a recent concurrent work [KK25]. In [KK25], the main focus was to explore the relationship with natural proofs. By contrast, we develop and use the same inequality as a technical lemma to provide guarantees for our inference algorithm.

From this, we derive

$$\begin{aligned}\mathrm{E}_{i\sim[m]}\left[\mathrm{qcd}^{\mathsf{poly}}(x_i \mid x_1,\ldots,x_{i-1})\right] &= \frac{1}{m}\sum_{i=1}^{m}\left(\mathrm{q}^{\mathsf{poly}}(x_i \mid x_1,\ldots,x_{i-1}) - \mathrm{K}(x_i \mid x_1,\ldots,x_{i-1})\right)\\ &\le \frac{\mathrm{qcd}^{\mathsf{poly}}(x_1,\ldots,x_m)}{m} + O(\log t).\end{aligned}$$

Notice that the inequality above demonstrates the averaging of computational depth as a consequence of Lemma 3.3.

Now suppose $x_1,\ldots,x_m$ are sampled from a randomized Turing machine Π described using s bits. By the slow growth law, we have that with probability at least $1-\delta$ (over the choice of $x_1,\ldots,x_m$), the right-hand side is further bounded as

$$\frac{\mathrm{qcd}^{\mathsf{poly}}(x_1,\ldots,x_m)}{m} + O(\log t) \le \frac{s + O(\log \delta^{-1})}{m} + O(\log t) \le \frac{O(s)}{m} + O(\log t),$$

where we assumed that $s \ge \log \delta^{-1}$. Thus, the average computational depth per block is bounded by $O(\log t)$ whenever $m \ge O(s/\log t)$.

It is known that $\mathrm{qcd}^t(x) + O(\log t)$ is nonnegative, thus, by applying Markov's inequality, we obtain that $\mathrm{qcd}^{\mathsf{poly}}(x_i \mid x_1,\ldots,x_{i-1}) \le \delta^{-1}\cdot O(\log t)$ with probability $1-\delta$ over $i \sim [m]$.

We remark that the bottleneck in the running time of our inference algorithm, i.e., the dependence on $t^{\delta^{-1}}$ arises from this Markov-based step. In particular, improving this point would yield a fully polynomial-time inference algorithm, while keeping the rest of the framework and analysis unchanged.

Proof Sketch of Lemma 3.2. Suppose $\bar{x}^1,\ldots,\bar{x}^b$ are sampled from a randomized Turing machine Π described using s bits, where each $\bar{x}^i$ represents the i-th block.

The domination property for $\mathrm{Q}^{\mathsf{poly}}$ (cf. [HN23, Proposition 6.11]) shows that for any $\bar{x}^1,\ldots,\bar{x}^b$ in the support,

$$\mathrm{q}^{\mathsf{poly}}(\bar{x}^1,\ldots,\bar{x}^b) \le O(s) - \log \mathcal{D}_\Pi(\bar{x}^1,\ldots,\bar{x}^b),$$

where $\mathcal{D}_\Pi(\bar{x}^1,\ldots,\bar{x}^b)$ denotes the probability that Π outputs the sequence $\bar{x}^1,\ldots,\bar{x}^b$.

Now, applying Lemma 3.3, we obtain for all such sequences:

$$\begin{aligned}\sum_{i=1}^{b}\mathrm{q}^{\mathsf{poly}}(\bar{x}^i \mid \bar{x}^1,\ldots,\bar{x}^{i-1}) &\le \mathrm{q}^{\mathsf{poly}}(\bar{x}^1,\ldots,\bar{x}^b) + b\cdot O(\log t)\\ &\le O(s) - \log \mathcal{D}_\Pi(\bar{x}^1,\ldots,\bar{x}^b) + b\cdot O(\log t)\\ &= O(s) - \sum_{i=1}^{b}\log \mathcal{D}_\Pi(\bar{x}^i \mid \bar{x}^1,\ldots,\bar{x}^{i-1}) + b\cdot O(\log t),\end{aligned}$$

where $\mathcal{D}_\Pi(\cdot \mid \cdot)$ denotes the conditional probability under Π.

By rearranging, we get

$$\begin{aligned}\sum_{i=1}^{b}\log\frac{\mathcal{D}_\Pi(\bar{x}^i \mid \bar{x}^1,\ldots,\bar{x}^{i-1})}{\Pr\left[\bar{x}^{i-1} \leftarrow \mathrm{Q}^{\mathsf{poly},\bar{x}^{<i}}\right]} &= \sum_{i=1}^{b}\left(\mathrm{q}^{\mathsf{poly}}(\bar{x}^i \mid \bar{x}^1,\ldots,\bar{x}^{i-1}) + \log \mathcal{D}_\Pi(\bar{x}^i \mid \bar{x}^1,\ldots,\bar{x}^{i-1})\right)\\ &\le O(s) + b\cdot O(\log t).\end{aligned}$$

Taking the expectation over $i \sim [b]$ and over samples $\bar{x}^1, \ldots, \bar{x}^b$ drawn from Π, we conclude:

$$\mathrm{E}_{i\sim[b]}\left[\mathrm{KL}\left(\bar{X}^i|\bar{X}^{<i}\middle\|\mathrm{Q}^{\mathsf{poly},\bar{X}^{<i}}\right)\right] \leq \frac{O(s)}{b} + O(\log t),$$

where $\bar{X}^i$ and $\bar{X}^{<i}$ denote the distributions over the i-th block and its preceding blocks, respectively, and $\mathrm{KL}(\cdot\|\cdot)$ denotes KL divergence.

In other words, if $b \geq O(s/\log t)$, the KL divergence between (i) the conditional distribution of the i-th block given preceding blocks and (ii) the universal distribution given preceding blocks as advice is at most $O(\log t)$ in expectation.

The remaining steps follow from standard arguments in probability theory, particularly applying the chain rule for KL divergence to distribute the bound of $O(\log t)$ across the w positions in a block, yielding an average KL divergence of $O(\log t)/w$ over positions $j \in [w]$. We refer the reader to Section 6 for the complete proof.

3.4 Short Comments: Corollaries on Learning in Heuristica

Corollaries 2.2 to 2.5 follow directly from Theorem 2.1 via the framework of the *cheating learner*, introduced in [HN23]. Below we provide a high-level overview of the derivations.

Corollary 2.3 follows immediately, since learning ACDs is a special case of inductive inference. Corollary 2.2 is also a special case. However, to improve the dependence on the confidence parameter from δ^{-2} to $\log \delta^{-1}$, we execute the inference algorithm with constant confidence error repeatedly for $O(\log \delta^{-1})$ rounds. We then identify the largest cluster among the resulting $O(\log \delta^{-1})$ hypotheses, where each hypothesis is statistically close to the others. This clustering-based identification relies on the algorithm for approximating statistical distance between samplers, developed in [NR06]. Corollary 2.5 is obtained by applying the inference algorithm to extrapolate label distributions and performing empirical loss minimization against them. A naive analysis using total variation distance would result in poor dependence on the accuracy parameter ϵ in the round complexity (see [HN23] for a more detailed argument). Instead, we leverage the γ-boundedness condition of loss functions in the analysis, following an approach similar to that in [MF98]. Finally, Corollary 2.4 follows easily as a corollary of Corollary 2.5.

Organization of This Paper. The remainder of this paper is organized as follows. In Section 4, we introduce additional preliminaries. In Section 5, we provide the proof of Lemma 3.3. In Section 6, we establish the main theorem (Theorem 2.1), which also includes the proofs of the technical lemmas (Lemmas 3.1 and 3.2). In Section 7 and Section 8, we present applications of our inference algorithm to learning distributions and sequential loss minimization, respectively. The formal proofs of Corollaries 2.2 and 2.3 are given in Section 7, while those of Corollaries 2.4 and 2.5 appear in Section 8.

4 Preliminaries

For preliminaries on the universal Turing machine U, Kolmogorov complexity, the universal distribution, q^t-complexity, computational depth, and universal extrapolation, see Section 3.1.

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 $k \leq k' \leq n$ 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 let $x_{<i} := \langle x_1, \ldots, x_{i-1} \rangle$, namely, *the binary encoding* of the $i-1$ sequence of strings.

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

For each $p \in [0,1]$, let $\mathrm{Ber}(p)$ be a Bernoulli distribution with parameter p. 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 any distribution $\mathcal{D}$ and $k \in \mathbb{N}$, let $\mathcal{D}^k$ denote the k-fold product distribution whose marginal distribution is identical to $\mathcal{D}$.

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 say that an s-size randomized Turing machine (or program) $\Pi \in \{0,1\}^{\leq s}$ that takes advice z and produces a stream $x_1, \ldots, x_m, \ldots \in \{0,1\}^*$ in t time to express that $U^{t',z,r}(\Pi)$ outputs $x_1 \circ \cdots \circ x_m$ as prefix for $t' \geq t$ and $r \sim \{0,1\}^{t'}$ (recall that we assume the output tape of U is one-way). We use the notation $x_1, \ldots, x_m \sim \Pi$ to denote sampling of the first m sequences from Π. When Π takes external advice $z \in \{0,1\}^*$, we write the sampling as $x_1, \ldots, x_m \sim \Pi(z)$.

For simplicity, we may identify a Turing machine Π with its binary encoding used as input to the universal Turing machine U.

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 also 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*

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

Lemma 4.2 (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 $(\mathcal{Y}^1, \ldots, \mathcal{Y}^m)$,

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

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 AvgBPP denote the class of distributional problems that admit a polynomial-time 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 4.3 (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 now review several relevant notions that were not introduced in Section 3.1.

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

Definition 4.4 (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 $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$ to denote that $\mathrm{Gap}_c\mathrm{K^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$ for some constant $c \geq 0$.

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

Theorem 4.5 ([Hir20b; GKLO22]). *If* $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$, *then* $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$.

Lemma 4.6 ([cf. HS17]). *If* $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$, *then there is no auxiliary-input one-way function.*

The following proposition is straightforward.

Proposition 4.7. *If* $\mathrm{GapK^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$, *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 $\mathrm{Gap}_c\mathrm{K^t}$-vs-K $\in \mathsf{pr\text{-}BPP}$ 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 $A(x, 1^{s^*-1}, 1^t) = 0$. □

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.

Lemma 4.9 ([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 4.10. *There exists a polynomial* p *such that for each* $t, n \in \mathbb{N}$ *with* $t \geq O(n)$ *and* $x, y \in \{0,1\}^{\leq n}$,

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

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

We first observe that

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

Indeed, if this inequality were false, then

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

which is trivially contradiction.

Equation (1) 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 (1) 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 4.11 (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 4.11 to the universal distribution Q^t, we immediately obtain the following upper bound:

Lemma 4.12. *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 4.13 ([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 4.14 (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 4.15 (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 4.16 ([HN23, Lemma 6.15])**.** *There exists a polynomial p such that that for every $z \in \{0,1\}^*$ and every t_Π-time randomized Turing machine Π that takes z as advice, and every $i, t, \delta^{-1} \in \mathbb{N}$ with $t \geq p(t_\Pi + |\Pi|)$,*

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

We also state a corollary that follows as a special case of Lemma 4.16:

Lemma 4.17. *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 $t \geq p(|x|)$,*

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

Proof. This follows by applying Lemma 4.16 to a program that simply embeds x and outputs it (without using randomness). □

5 Online and Offline Algorithmic Information

In this section, we prove the following key inequality, which can be seen as a multi-block extension of the time-bounded symmetry of information.

Lemma 5.1. *If* $\mathrm{GapK^t\text{-}vs\text{-}K} \in \mathsf{pr\text{-}BPP}$*, then there exists a polynomial* p *such that for every* $n, m, t \in \mathbb{N}$ *with* $t \geq O(m+n)$ *and every* $x_1, \ldots, x_m \in \{0,1\}^{\leq n}$*, it holds*

$$\sum_{i=1}^{m} \mathsf{pK}^{p(t)}(x_i \mid x_{<i}) \leq \mathsf{pK}^t(x_1, \ldots, x_m) + m \cdot \log p(t).$$

Before presenting the full proof, we briefly sketch the main idea.

5.1 Proof Ideas

We first consider the following simpler form:

$$\sum_{i=1}^{m} \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}) \leq \mathsf{pK}^t(x_1, \ldots, x_m) + m \cdot O(\log t),$$

where $k_i := \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}) - O(\log t)$ for each $i \in [m-1]$.

In other words, we allow each round to receive as additional advice the previous complexity estimates $k_1, \ldots, k_{i-1}$. This version of the lemma can in fact be proved via a natural extension of the symmetry of information argument from the two-block case, as developed by Hirahara [Hir22] and Goldberg and Kabanets [GK22]. We now outline this approach. For simplicity, we may use the shorthand poly to denote time bounds, thereby avoiding the need to explicitly track polynomial overheads.

Let p be a large enough polynomial determined by p_{DP} in Lemma 4.15. 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 inductively 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 4.7, and outputs 1 (i.e., interprets w as pseudorandom) if the approximated complexity is smaller than a threshold $\tau := \mathsf{pK}^t(x_1, \ldots, x_m) + \sum_i |z_i| + O(\log t)$. The value τ can be encoded in $O(\log t)$ bits as a natural number in binary.

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:

$$\mathrm{K}^{\mathsf{poly}}(w_{\mathsf{DP}}) \leq \mathsf{pK}^t(x_1, \ldots, x_m) + \sum_{i=1}^{m} |z_i| + O(\log t).$$

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,

$$\mathrm{K}(w) \geq \sum_{i=1}^{m}(|z_i| + k_i) - O(1) = \mathsf{pK}^t(x_1, \ldots, x_m) + \sum_{i=1}^{m}|z_i| + c\log t - O(1).$$

Thus, by choosing c sufficiently large, the distinguisher D can distinguish the pseudorandom string w_{DP} from a uniformly random string w with constant advantage $\gamma > 0$.

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, we have

$$\Pr[D(\mathsf{hyb}_m) = 1] - \Pr[D(\mathsf{hyb}_0) = 1] \geq \gamma.$$

Suppose that for every $i \in [m-1]$,

$$\Pr[D(\mathsf{hyb}_i) = 1] - \Pr[D(\mathsf{hyb}_{i-1}) = 1] \leq \frac{\gamma}{2m}. \tag{2}$$

Then by the telescoping sum, we have:

$$\begin{aligned}
&\Pr[D(\mathsf{hyb}_m) = 1] - \Pr[D(\mathsf{hyb}_{m-1}) = 1] \\
&= \Pr[D(\mathsf{hyb}_m) = 1] - \Pr[D(\mathsf{hyb}_0) = 1] - \sum_{i=1}^{m-1}\left(\Pr[D(\mathsf{hyb}_i) = 1] - \Pr[D(\mathsf{hyb}_{i-1}) = 1]\right) \\
&\geq \gamma - \frac{(m-1)\gamma}{2m} \\
&\geq \gamma/2.
\end{aligned}$$

Namely, we can construct a distinguisher D_m for $\mathsf{DP}_{k_m}(x_m; z_m)$ given advice $x_1, \ldots, x_{m-1}, k_1, \ldots, k_{m-1}$ as follows. Given an input w_m, 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 4.15) yields:

$$\begin{aligned}
\mathsf{pK}^{p(t)}(x_m \mid x_1, \ldots, x_{m-1}, k_1, \ldots, k_{i-1}) &\leq k_m + O(\log t) \\
&\leq \mathsf{pK}^t(x_1, \ldots, x_m) - \sum_{i=1}^{m-1} k_i + O(\log t) \\
&= \mathsf{pK}^t(x_1, \ldots, x_m) - \sum_{i=1}^{m-1} \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \cdots, k_{i-1}) + m \cdot O(\log t).
\end{aligned}$$

In other words, the conclusion of the lemma follows as long as the indistinguishability condition in Equation (2) holds for every $i \in [m-1]$.

Indeed, we can observe these cases, and this is precisely where the additional advice strings $k_1, \ldots, k_{i-1}$ become necessary.

The argument here mirrors the construction of D_m. For any $i \in [m-1]$, suppose that

$$\Pr[D(\mathsf{hyb}_i) = 1] - \Pr[D(\mathsf{hyb}_{i-1}) = 1] > \frac{\gamma}{2m}.$$

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 4.15) gives

$$\begin{aligned}\mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \ldots, k_{i-1}) &\leq k_i + O(\log tm\gamma^{-1}) \\ &\leq \mathsf{pK}^{p(t)}(x_i \mid x_{<i}, k_1, \cdots, k_{i-1}) - c\log t + O(\log t),\end{aligned}$$

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

$$\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{m-1}; z_{m-1}) \circ w_i \circ w_{i+1} \circ \cdots \circ w_m,$$

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.

Removing Advice Strings Since it is currently unclear whether each k_i can be computed in polynomial time, particularly because we do not yet know whether conditional pK^t is efficiently computable under the assumption $\mathsf{DistNP} \subseteq \mathsf{AvgBPP}$, we initially treat these as external advice. This introduces an additive overhead of

$$\sum_{i=1}^{m} O(\log k_1 + \ldots + \log k_{i-1}) = m^2 \cdot O(\log t).$$

This is not sufficient for our purposes, as highlighted in Section 3.3.

To eliminate these advice strings, we must replace them with quantities that are efficiently computable. In the formal proof, we achieve this by determining each value $\tilde{k}_i$ adaptively using the distinguisher D. Specifically, for each i, we define $\tilde{k}_i$ to be the largest value such that D cannot distinguish $\mathsf{DP}\tilde{k}_i(x_i; z_i)$ from uniform randomness, given the previously estimated values $\tilde{k}_1, \ldots, \tilde{k}_{i-1}$. This indistinguishability is verified empirically using sampling. The randomness ρ_i used during the empirical estimation of $\tilde{k}_i$ is then treated as external advice for future steps. This parameter tuning strategy essentially mirrors Hirahara's worst-case-to-average-case reduction technique developed in [Hir18; Hir20b]. In our setting, we apply it adaptively across blocks.

In fact, the amount of randomness ρ_i used in the empirical estimation procedure is much larger than the binary representation of each k_i. However, as shown in Proposition 4.10, adding randomness to the advice string does not increase the value of pK in expectation. This observation allows us to eliminate the need for explicitly providing the advice strings $k_1, \ldots, k_m$.

5.2 Full Proof

We now present the full proof based on the outlined ideas.

Proof of Lemma 5.1. 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 $t \in \mathbb{N}$,

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

Let $n, m, t \in \mathbb{N}$ and $x_1, \ldots, x_m \in \{0,1\}^{\leq n}$ satisfying $t \geq O(m+n)$ as in the lemma. For each i, let $n_i = |x_i|$.

Let p_1 be a large enough polynomial and c_1 be a constant we specify later. Let $N = \sum_{i=1}^m n_i$ and $M \in \mathbb{N}$ be

$$M = \mathsf{pK}^t(x_1, \ldots, x_m) + 6 - c_1\lceil m \log nt\rceil - c_0\lceil \log(p_1(t,n,m)\cdot(|r| + 2N^2 + \mathsf{pK}^t(x_1,\ldots,x_m)+6)))\rceil.$$

We consider a randomized algorithm D that is given $y \in \{0,1\}^{2N^2+M}$, selects $r \sim \{0,1\}^t$, and outputs 1 if

$$\tilde{K}\left((r,y), 1^{p_1(t,n,m)}\right) \leq \mathsf{pK}^t(x_1,\ldots,x_m) + |r| + 2N^2 + c_1 m \log nt + c_0 \log(p_1(t,n,m)\cdot(|r|+|y|));$$

outputs 0 otherwise. Here, we regard $\mathsf{pK}^t(x_1,\ldots,x_m) \in \mathbb{N}$ as embedded advice described in $O(\log mn)$ bits.

Let $k_1, \ldots, k_{m-1} \in \mathbb{N} \cup \{0\}$ be arbitrary parameters satisfying $k_i \leq 2n_i$ for each i. We define $k_m \in \mathbb{N}$ as

$$k_m = M - \sum_{i=1}^{m-1} k_i.$$

Notice 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 $\sum_{i=1}^m (2n_i^2 + k_i) = 2N^2 + M$ regardless of the choices of $k_1, \ldots, k_{m-1}$.

We first observe that D distinguishes $\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m)$ from a truly random string.

Notice that we can efficiently compute $(r, \mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m))$ from $r, x_1, \ldots, x_m$, $z_1, \ldots, z_m$, and $k_1, \ldots, k_m$. 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 $r, z_1, \ldots, z_m$ (where $r \in \{0,1\}^t$ and $z_i \in \{0,1\}^{2n_i^2}$ for each i),

$$\begin{aligned}
&\mathrm{K}^{p_1(t,n,m)}(r, \mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m)) \\
&\leq \mathrm{K}^t(x_1, \ldots, x_m \mid r) + |r| + |z_1| + \cdots + |z_m| + O(\log k_1 + \ldots + \log k_m) + O(\log nmt) \\
&\leq \mathrm{K}^t(x_1, \ldots, x_m \mid r) + |r| + 2N + c_1 \cdot m \log nt.
\end{aligned}$$

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

$$\tilde{K}\left((r, y_1 \circ \cdots \circ y_m), 1^{p_1(t,n,m)}\right) \leq \mathrm{K}^{p_1(t,n,m)}(r, y_1 \circ \cdots \circ y_m) + c_0 \log(p_1(t,n,m)\cdot(|r| + |y_1 \circ \cdots \circ y_m|))$$

In addition, by Proposition 4.8, $\mathrm{K}^t(x \mid r) \leq \mathsf{pK}^t(x)$ with probability at least $2/3$ over $r \sim \{0,1\}^t$. If these two events occur simultaneously, it holds that

$$\begin{aligned}
&\tilde{K}\left((r, y_1 \circ \cdots \circ y_m), 1^{p_1(t,n,m)}\right) \\
&\leq \mathrm{K}^{p_1(t,n,m)}(r, y_1 \circ \cdots \circ y_m) + c_0 \log(p_1(t,n,m)\cdot(|r| + |y_1 \circ \cdots \circ y_m|)) \\
&\leq \mathrm{K}^t(x_1, \ldots, x_m \mid r) + |r| + 2N^2 + c_1 \cdot m \log nt + c_0 \log(p_1(t,n,m)\cdot(|r| + |y_1 \circ \cdots \circ y_m|)) \\
&\leq \mathsf{pK}^t(x_1, \ldots, x_m) + |r| + 2N^2 + c_1 \cdot m \log nt + c_0 \log(p_1(t,n,m)\cdot(|r| + |y_1 \circ \cdots \circ y_m|)).
\end{aligned}$$

Thus, we have

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

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 $r,w_1,\ldots,w_m$,

$$\begin{aligned}\mathrm{K}(r,w_1\circ\cdots\circ w_m)&\geq|r|+|w_1|+\cdots+|w_m|-6\\&=|r|+2N^2+M-6\\&\geq\mathsf{pK}^t(x_1,\ldots,x_m)+|r|+2N^2+c_1m\log nt+c_0\log(p_1(t,n,m)\cdot(|r|+2N^2+M)),\end{aligned}$$

where we used the definition of M and $\mathsf{pK}^t(x_1,\ldots,x_m)+6\geq M$ in the last inequality.

Since $\tilde{K}((r,w_1\circ\cdots\circ w_m),1^t)>\mathrm{K}(r,w_1\circ\cdots\circ w_m)$ with probability at least $2/3$, the union bound implies that

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

Thus,

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

Now, we consider a randomized procedure K that generates $k_1,\ldots,k_{m-1}$ 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 $\pm1/(216m)$ and negligible (in n,m,t) confidence error, where $|w_i|=2n_i^2+j$, and r is chosen so that the total length of the input becomes $2N^2+M$. 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 $\tilde{dp}_j^i-\tilde{tr}_j^i\leq1/(54m)$ (if there is no such j, let $k_i=0$). For each $i\in[m-1]$, 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 n, m, and t) from $x_1,\ldots,x_i$, $2N^2+M$, 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 determines each value of k_i. Then, it must hold, for each $i\in[m-1]$,

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

and

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

Now we define k_m depending on $k_1, \ldots, k_{m-1}$ as above. For notational simplicity, let $dp^i = dp^i_{k_i}$ for each $i \in [m-1]$, and let

$$\begin{aligned}
dp^0 &:= \Pr_{D,w_1,\ldots,w_m}\left[D\left(w_1 \circ \cdots \circ w_m\right) = 1\right] \\
dp^m &:= \Pr_{D,z_1,\ldots,z_m}\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_m}(x_m; z_m)\right) = 1\right] \\
tr^m &:= \Pr_{D,z_1,\ldots,z_{m-1},w_m}\left[D\left(\mathsf{DP}_{k_1}(x_1; z_1) \circ \cdots \circ \mathsf{DP}_{k_{m-1}}(x_{m-1}; z_{m-1}) \circ w_m\right) = 1\right].
\end{aligned}$$

Then, we can observe that

$$\begin{aligned}
\frac{1}{18} \leq dp^m - dp^0 = \sum_{i=1}^{m}(dp^i - dp^{i-1}) &= \sum_{i=1}^{m}(dp^i - tr^i) \\
&= dp^m - tr^m + \sum_{i=1}^{m-1}(dp^i - tr^i) \leq dp^m - tr^m + m \cdot \frac{1}{36m}.
\end{aligned}$$

By arranging the above,

$$dp^m - tr^m \geq \frac{1}{18} - \frac{1}{36} = \frac{1}{36}.$$

We will show that for a large enough polynomial p_2, the following holds for each $i \in [m-1]$:

$$\mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i}) \leq k_i + 1 + \log p_2(n,m,t) \tag{3}$$

and

$$\mathsf{pK}^{p_2(t,n,m)}(x_m \mid x_{<m}, \rho_{<m}) \leq k_m + \log p_2(n,m,t) \tag{4}$$

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

The inequalities above imply that

$$\begin{aligned}
&\mathsf{pK}^{p_2(t,n,m)}(x_m \mid x_{<m}, \rho_{<m}) \\
&\quad\leq k_m + \log p_2(n,m,t) \\
&\quad= M - \sum_{i=1}^{m-1} k_i + \log p_2(n,m,t) \\
&\quad\leq M + m \cdot (\log p_2(n,m,t) + 1) + \log p_2(n,m,t) - \sum_{i=1}^{m-1} \mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i}).
\end{aligned}$$

By arranging the above, we obtain

$$\begin{aligned}
\sum_{i=1}^{m} \mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i}) &\leq M + m \cdot (\log p_2(n,m,t) + 1) + \log p_2(n,m,t) \\
&\leq \mathsf{pK}^t(x_1, \ldots, x_m) + 6 + m \cdot (\log p_2(n,m,t) + 1) + \log p_2(n,m,t) \\
&\leq \mathsf{pK}^t(x_1, \ldots, x_m) + m \cdot \log p_3(n,m,t),
\end{aligned}$$

for a large enough polynomial p_3.

Now, we take into account the choices of $\rho_1, \ldots, \rho_{m-1}$. Since the empirical estimations in K are performed with negligible confidence error,

$$\Pr_{\rho_1,\ldots,\rho_{m-1}}\left[\sum_{i=1}^{m} \mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i}) < \mathsf{pK}^t(x_1, \ldots, x_m) + m \cdot \log p_3(n,m,t)\right] \geq 1 - \mathsf{negl}(nm).$$

Thus, for $t \geq O(m+n)$,

$$\begin{aligned}\mathrm{E}_{\rho_1,\ldots,\rho_{m-1}}\left[\sum_{i=1}^{m} \mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i})\right] &\leq \mathsf{pK}^t(x_1,\ldots,x_m) + m \cdot \log p_3(n,m,t) + 2nm \cdot \mathsf{negl}(nm) \\ &\leq \mathsf{pK}^t(x_1,\ldots,x_m) + m \cdot \log p_4(n,m,t),\end{aligned}$$

for a large enough polynomial p_4.

From Proposition 4.10, for a large enough polynomial p_5,

$$\sum_{i=1}^{m} \mathsf{pK}^{p_5(t,n,m)}(x_i \mid x_{<i}) \leq \mathrm{E}_{\rho_1,\ldots,\rho_{m-1}}\left[\sum_{i=1}^{m} \mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i})\right] + m \cdot O(\log t).$$

By taking p as a large enough polynomial, we obtain that for $t \geq O(m+n)$,

$$\begin{aligned}\sum_{i=1}^{m} \mathsf{pK}^{p(t)}(x_i \mid x_{<i}) &\leq \sum_{i=1}^{m} \mathsf{pK}^{p_5(t,n,m)}(x_i \mid x_{<i}) \\ &\leq \mathrm{E}_{\rho_1,\ldots,\rho_{m-1}}\left[\sum_{i=1}^{m} \mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i})\right] + m \cdot O(\log t) \\ &\leq \mathsf{pK}^t(x_1,\ldots,x_m) + m \cdot \log p_4(n,m,t) + m \cdot O(\log t) \\ &\leq \mathsf{pK}^t(x_1,\ldots,x_m) + m \cdot \log p(t).\end{aligned}$$

Therefore, it suffices to show Equations (3) and (4). We prove them simultaneously for all $i \in [m]$ as follows.

Let $\ell_i = k_i + 1$ if $i \in [m-1]$, or $\ell_i = k_m$ if $i = m$. Then,

$$\begin{aligned}&\Pr_{D,z_1,\ldots,z_i,r}\left[D\left(\mathsf{DP}_{k_1}(x_1;z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1};z_{i-1}) \circ \mathsf{DP}_{\ell_i}(x_i;z_i) \circ r\right) = 1\right] \\ &\qquad - \Pr_{D,z_1,\ldots,z_{i-1},w_i,r}\left[D\left(\mathsf{DP}_{k_1}(x_1;z_1) \circ \cdots \circ \mathsf{DP}_{k_{i-1}}(x_{i-1};z_{i-1}) \circ w_i \circ r\right) = 1\right] > \frac{1}{108m}. \quad (5)\end{aligned}$$

Based on the above, we construct the following algorithm D_i that distinguishes $\mathsf{DP}_{\ell_i}(x_i;z_i)$ from truly random strings *given* D, $x_{<i} = (x_1,\ldots,x_{i-1})$, and $\rho_{<i} = (\rho_1,\ldots,\rho_{i-1})$: On input $y \in \{0,1\}^{2n_i^2+\ell_i}$, the distinguisher D_i first executes K with $x_{<i}$ and randomness $\rho_{<i}$ 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 $z_i \sim \{0,1\}^{2n_i^2}$, and r is selected so that the total length of the input becomes $2N^2 + M$ (note that $2N^2 + M$ is embedded into the description of D).

Then, Equation (5) is rewritten as

$$\Pr_{D_i,z_i}\left[D_i\left(\mathsf{DP}_{\ell_i}(x_i;z_i)\right) = 1\right] - \Pr_{D_i,w_i}\left[D_i\left(w_i\right) = 1\right] \geq \frac{1}{108m}.$$

From Lemma 4.15,

$$\mathsf{pK}^{p_6(t,n,m)}(x_i \mid D_i) \leq \ell_i + \log p_6(t,n,m),$$

for a large enough polynomial p_6. Thus, by taking p_2 large enough,

$$\begin{aligned}\mathsf{pK}^{p_2(t,n,m)}(x_i \mid x_{<i}, \rho_{<i}) &\leq \mathsf{pK}^{p_3(t,n,m)}(x_i \mid D_i) + |D| + O(\log nm) \\ &\leq \ell_i + \log p_2(t,n,m),\end{aligned}$$

as desired. □

6 Complexity-Theoretic Inductive Inference

In this section, we prove the main theorem.

Theorem 6.1. *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exists a polynomial-time randomized algorithm* L *such that for every* $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ *and every* s*-size randomized Turing machine* Π *that produces at least* m *binary strings* $x_1, \dots, x_m, \dots \in \{0,1\}^n$ *in* t $(\geq s)$ *time, if* $m \geq O(s\epsilon^{-2}\delta^{-1})$*, then*

$$\Pr_{i, x_{<i}}\left[\Delta_{\mathrm{tv}}\left(L\left(x_{<i}; 1^{\langle n,m,s,t,\epsilon^{-1},\delta^{-1},t^{\delta^{-1}}\rangle}\right), X_i | x_{<i}\right) \leq \epsilon\right] \geq 1-\delta,$$

where $i \sim [m]$*, and* $X_i|x_{<i}$ *represents the conditional distribution of* x_i *given the previous samples are* $x_{<i}$*.*

6.1 Technical Lemmas

We begin by proving the two technical lemmas that underpin our analysis.

Lemma 6.2 (Averaging Lemma for Computational Depth)**.** *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exist a constant* c *and a polynomial* τ *such that for every* $n, m, t, \delta^{-1} \in \mathbb{N}$*, and every randomized Turing machine* Π *that produces* $x_1, \dots, x_m \in \{0,1\}^{\leq n}$ *in time* t_Π $(\geq |\Pi|)$*, if* $t \geq \tau(t_\Pi)$ *and* $m \geq (\mathrm{qcd}^{t^{1/c}}(\Pi) + c\log\delta^{-1})/\log t$*, then*

$$\Pr_{i\sim[m], x_1, \dots, x_i}\left[\mathrm{qcd}^t(x_i \mid x_{<i}) \leq c\delta^{-1}\log t\right] \geq 1-\delta.$$

Proof. Without loss of generality, we assume that $n = \max_i |x_i|$.

Let p be the polynomial in Lemma 5.1, c_0 be the constant in Proposition 4.13, p_0 be the polynomial in Lemma 4.12, and p_1 be the polynomial in Lemma 4.16. Without loss of generality, we assume that any $t \geq \tau(t_\Pi)$ satisfies $p^{-1}(t/c_0) \geq O(n+m)$ and $p_1^{-1}(p_0^{-1}(p^{-1}(t/c_0)/2)) \geq t^{1/c}$ by taking τ and c large enough (notice that $t_\Pi \geq n+m$ is always required for Π to output $x_1, \dots, x_m$).

From Lemma 4.16, with probability at least $1-\delta/2$ over the choice of $x_1, \dots, x_m \leftarrow \Pi$,

$$\begin{aligned}\mathrm{qcd}^{p_0^{-1}(p^{-1}(t/c_0)/2)}(x_1 \circ \cdots \circ x_m) &\leq \mathrm{qcd}^{p_1^{-1}(p_0^{-1}(p^{-1}(t/c_0)/2))}(\Pi)\\ &\leq \mathrm{qcd}^{t^{1/c}}(\Pi) + O(\log t\delta^{-1}),\end{aligned}$$

where we used $p_1^{-1}(p_0^{-1}(p^{-1}(t/c_0)/2)) \geq t^{1/c}$. Below, we consider such $x_1, \dots, x_m$.

From Lemma 5.1 and the inequalities above,

$$\begin{aligned}&\sum_{i=1}^m \mathsf{pK}^{t/c_0}(x_i \mid x_{<i})\\ &\leq \mathsf{pK}^{p^{-1}(t/c_0)}(x_1, \dots, x_m) + m \cdot O(\log t)\\ &\leq \mathsf{pK}^{p^{-1}(t/c_0)/2}(x_1 \circ \cdots \circ x_m) + m \cdot O(\log t) + \sum_{i=1}^m O(\log |x_i|)\\ &\leq \mathsf{pK}^{p^{-1}(t/c_0)/2}(x_1 \circ \cdots \circ x_m) + m \cdot O(\log t)\\ &\leq \mathrm{q}^{p_0^{-1}(p^{-1}(t/c_0)/2)}(x_1 \circ \cdots \circ x_m) + m \cdot O(\log t)\\ &\leq \mathrm{qcd}^{p_0^{-1}(p^{-1}(t/c_0)/2)}(x_1 \circ \cdots \circ x_m) + \mathrm{K}(x_1, \dots, x_m) + m \cdot O(\log t)\\ &\leq \mathrm{qcd}^{t^{1/c}}(\Pi) + \mathrm{K}(x_1, \dots, x_m) + O(\log\delta^{-1}) + m \cdot O(\log t)\end{aligned}$$

$$\leq \mathrm{qcd}^{t^{1/c}}(\Pi) + \mathrm{K}(x_1, \ldots, x_m \mid t/c_0) + O(\log \delta^{-1}) + m \cdot O(\log t)$$
$$\leq \mathrm{qcd}^{t^{1/c}}(\Pi) + \sum_{i=1}^{m} \mathrm{K}(x_i \mid x_{<i}, t/c_0) + O(\log \delta^{-1}) + m \cdot O(\log t).$$

By rearranging the above, we obtain that for any large enough constant $c_1 > 0$ and any $m \geq (\mathrm{qcd}^{1/c}(\Pi) + c_1 \log \delta^{-1})/\log t$,

$$\begin{aligned}
&\mathrm{E}_{i\sim[m]}\left[\mathsf{pK}^{t/c_0}(x_i \mid x_{<i}) - \mathrm{K}(x_i \mid x_{<i}, t/c_0) + \log(t/c_0)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\mathsf{pK}^{t/c_0}(x_i \mid x_{<i}) - \mathrm{K}(x_i \mid x_{<i}, t/c_0)\right) + \log(t/c_0) \\
&\leq \frac{\mathrm{qcd}^{t^{1/c}}(\Pi) + O(\log \delta^{-1}) + m \cdot O(\log t)}{m} \\
&\leq \frac{\mathrm{qcd}^{t^{1/c}}(\Pi) + O(\log \delta^{-1})}{m} + O(\log t) \\
&\leq c_1 \log t.
\end{aligned}$$

Since $\mathsf{pK}^t(x_i \mid x_{<i}) - \mathrm{K}(x_i \mid x_{<i}, t/c_0) + \log(t/c_0)$ is always nonnegative from Lemma 4.9, Markov's inequality implies that

$$\Pr_{i\sim[m]}\left[\mathsf{pK}^{t/c_0}(x_i \mid x_{<i}) - \mathrm{K}(x_i \mid x_{<i}, t/c_0) + \log(t/c_0) \leq c_1 \delta^{-1} \log t\right] \geq 1 - \delta.$$

We observe that if the event above is satisfied, then from Proposition 4.13,

$$\begin{aligned}
\mathrm{q}^t(x_i \mid x_{<i}) - \mathrm{K}(x_i \mid x_{<i}) &\leq \mathsf{pK}^{t/c_0}(x_i \mid x_{<i}) - \mathrm{K}(x_i \mid x_{<i}, t/c_0) + O(\log t) \\
&\leq c_1 \delta^{-1} \log t + O(\log t).
\end{aligned}$$

By taking the constant c large enough, $\mathrm{qcd}^t(x_i \mid x_{<i}) \leq c \cdot \delta^{-1} \log t$. Thus, we obtain the lemma. □

Indeed, we use Lemma 6.2 in the following form.

Lemma 6.3. *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exist a constant* c *and a polynomial* τ *such that for every* $n, m, t, s, \delta^{-1} \in \mathbb{N}$ *and every* s*-size randomized Turing machine* Π *that produces* $x_1, \ldots, x_m \in \{0,1\}^{\leq n}$ *in time* t_Π $(\geq s)$*, if* $t \geq \tau(t_\Pi)$ *and* $m \geq (s + c \log \delta^{-1})/\log t$*, then*

$$\Pr_{i\sim[m], x_1, \ldots, x_i}\left[\mathrm{qcd}^t(x_i \mid x_{<i}) \leq c\delta^{-1} \log t\right] \geq 1 - \delta.$$

Proof. This follows directly from Lemma 6.2, noting that for $t^{1/c} \geq O(t_\Pi) \geq O(s)$, we have $\mathrm{qcd}^{t^{1/c}}(\Pi) \leq \mathrm{q}^{t^{1/c}}(\Pi) \leq s + O(1)$. □

We now turn to establishing the performance of our inference rule.

Lemma 6.4 (Universal Extrapolation with Restricted Context)**.** *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exist a constant* c *and a polynomial* p *satisfying the following for every* $n, s, b, w \in \mathbb{N}$ *and every* $z \in \{0,1\}^*$*: Let* Π *be an* s*-size randomized Turing machine that takes* z *as advice*[6] *and*

[6]While the advice z is not needed for the results in this work, we state the lemma in its general form to avoid reproving it in future applications, such as learning involving prior knowledge z.

produces at least $b \cdot w$ binary strings $x_1, \dots, x_{bw}, \dots \in \{0,1\}^n$ in t_Π ($\geq s$) 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 $t \geq p(t_\Pi + |z|)$,

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

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

In particular, if $b \geq (s+\mathrm{qcd}^{t^{1/c}}(z)))/\log t$ and $w \geq 2c\epsilon^{-1}\delta^{-1}\log t$ for $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$,

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

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

Proof. Let c_0 be the constant in Proposition 4.13, p_0 be the polynomial in Lemma 5.1, and p_1 be the polynomial in Lemma 4.12.

Without loss of generality, we assume that $t/c_0 \geq O(w+n+|z|)$, $p_0^{-1}(t/c_0) \geq O(bw+n+|z|)$, and $p_0^{-1}(t/c_0)/4 \geq p_1(t^{1/c})$ for any $t \geq p(t_\Pi+|z|)$ (notice that $t_\Pi \geq bwn$ must hold for Π to output bw strings of each length n) by taking p and c large enough relative to p_0, p_1, c_0, and the hidden constant in $O(\cdot)$.

From Lemma 5.1, we obtain that

$$\begin{aligned}\mathsf{pK}^{t/c_0}(z)+\sum_{i=1}^{b}\mathsf{pK}^{t/c_0}(\bar{x}^i\mid z,\bar{x}^{<i}) &\leq \mathsf{pK}^{p_0^{-1}(t/c_0)}(z,\bar{x}^1,\dots,\bar{x}^b)+b\cdot O(\log t)\\ &\leq \mathsf{pK}^{p_0^{-1}(t/c_0)/4}(\bar{x}^1,\dots,\bar{x}^b\mid z)+\mathsf{pK}^{p_0^{-1}(t/c_0)/4}(z)+b\cdot O(\log t).\end{aligned}$$

In addition, we assume that p is enough large so that any $t \geq p(t_\Pi+|z|)$ satisfies that the time bound $p_0^{-1}(t/c_0)/4$ is large enough for optimal coding for pK (Theorem 4.11). Then, we obtain

$$\begin{aligned}\mathsf{pK}^{p^{-1}(t/c_0)/4}(\bar{x}^1,\dots,\bar{x}^b\mid z) &\leq O(s)+O(\log t)-\log\Pr\left[(\bar{X}^1,\dots,\bar{X}^b)=(\bar{x}^1,\dots,\bar{x}^b)\right]\\ &= O(s)+O(\log t)-\sum_{i=1}^{b}\log\Pr\left[\bar{X}^i=\bar{x}^i\middle|\bar{X}^{<i}=\bar{x}^{<i}\right].\end{aligned}$$

From the two inequalities above and Proposition 4.13, we have

$$\begin{aligned}&\sum_{i=1}^{b}\left(q^t(\bar{x}^i\mid z,\bar{x}^{<i})+\log\Pr\left[\bar{X}^i=\bar{x}^i\middle|\bar{X}^{<i}=\bar{x}^{<i}\right]\right)\\ &\quad\leq\sum_{i=1}^{b}\left(\mathsf{pK}^{t/c_0}(\bar{x}^i\mid z,\bar{x}^{<i})+\log\Pr\left[\bar{X}^i=\bar{x}^i\middle|\bar{X}^{<i}=\bar{x}^{<i}\right]\right)+b\cdot O(\log t)\\ &\quad\leq O(s)+\mathsf{pK}^{p_0^{-1}(t/c_0)/4}(z)-\mathsf{pK}^{t/c_0}(z)+b\cdot O(\log t).\end{aligned}$$

From Lemma 4.9, we have

$$\mathrm{K}(z)\leq\mathrm{K}(z\mid t/c_0)+O(\log t)\leq\mathsf{pK}^{t/c_0}(z)+O(\log t).$$

From Lemma 4.12 and $p_0^{-1}(t/c_0)/4 \geq p_1(t^{1/c})$,

$$\mathsf{pK}^{p_0^{-1}(t/c_0)/4}(z) \leq \mathsf{pK}^{p_1(t^{1/c})}(z) \leq \mathrm{q}^{t^{1/c}}(z) + O(\log t).$$

The inequalities above imply

$$\mathsf{pK}^{p_0^{-1}(t/c_0)/4}(z) - \mathsf{pK}^{t/c_0}(z) \leq \mathrm{q}^{t^{1/c}}(z) - \mathrm{K}(z) + O(\log t) = \mathrm{qcd}^{t^{1/c}}(z) + O(\log t).$$

Therefore, we obtain

$$\begin{aligned}
&\sum_{i=1}^{b} \left(q^t(\bar{x}^i \mid z, \bar{x}^{<i}) + \log \Pr\left[\bar{X}^i = \bar{x}^i \middle| \bar{X}^{<i} = \bar{x}^{<i}\right] \right) \\
&\quad \leq O(s) + \mathsf{pK}^{p_0^{-1}(t/c_0)/4}(z) - \mathsf{pK}^{t/c_0}(z) + b \cdot O(\log t) \\
&\quad \leq O(s) + \mathrm{qcd}^{t^{1/c}}(z) + b \cdot O(\log t).
\end{aligned}$$

Notice that

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

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

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

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

$$\mathrm{E}_{i\sim[b]}\left[\mathrm{KL}\left(\bar{X}^i | \bar{X}^{<i} \middle\| \mathrm{Q}^{t,z,\bar{X}^{<i}}\right)\right] = \frac{1}{b}\sum_{i=1}^{b} \mathrm{KL}\left(\bar{X}^i | \bar{X}^{<i} \middle\| \mathrm{Q}^{t,z,\bar{X}^{<i}}\right) \leq \frac{c(s + \mathrm{qcd}^{t^{1/c}}(z))}{b} + c\log t. \quad (6)$$

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

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

Along with Equation (6), we derive the first part of the lemma as follows:

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

Next, we derive the second part of the lemma from the above.

If $b \geq (s + \mathrm{qcd}^{t^{1/c}}(z))/\log t$ and $w \geq 2c\epsilon^{-1}\delta^{-1}\log t$ are satisfied for $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, then

$$\frac{1}{w} \cdot \left(\frac{c(s + \mathrm{qcd}^{t^{1/c}}(z))}{b} + c\log t \right) \leq \frac{\epsilon\delta}{2c\log t} \cdot \left(\frac{c(s + \mathrm{qcd}^{t^{1/c}}(z))}{(s + \mathrm{qcd}^{t^{1/c}}(z))/\log t} + c\log t \right) = \epsilon\delta.$$

Therefore,

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

Notice that, by the definition of conditional KL divergence,

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

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

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

as desired. □

6.2 Proof of Inductive Inference

Now, we present the proof of the main theorem. Our argument relies on the following key lemma.

Lemma 6.5 (Universal Extrapolation Lemma)**.** *If there is no auxiliary-input one-way function, then there exists a randomized polynomial-time algorithm* UE *such that for all* $k, t, \epsilon^{-1}, \alpha \in \mathbb{N}$ *and all* $z, x \in \{0,1\}^*$ *with* $\mathrm{qcd}^t(x \mid z) \leq \alpha$,

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

Proof. The proof is the same as that of [HN23, Theorem 8.1] since it holds even in the presence of auxiliary inputs (i.e., advice). Thus, we refer the reader to [HN23, Section 8] for a complete proof. □

Proof of Theorem 6.1. Let p_0 and c_0 be the maximum of the polynomials and constants in Lemmas 4.17, 6.3 and 6.4, 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, $t'' \geq t' \geq 2$, $c_0 \geq 1$, and $c_0 \log t' \geq \log t''$ for each t.

For each $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ with $m \geq 34c_0^2 \cdot s\epsilon^{-2}\delta^{-1}$, let $b = \lceil 2s/\log t' \rceil$. 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 context-restricting universal extrapolation as in Lemma 6.4 on $\tilde{m}$ strings $x_1, \ldots, x_{\tilde{m}}$.

Below, we assume that $c_0 \log \delta^{-1} \leq s$. Otherwise, $2^s \leq \delta^{-c} = \mathsf{poly}(\delta^{-1})$. 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

$$\Pr_{i\sim[m]}[i > \tilde{m}] = \frac{m - \tilde{m}}{m} < \frac{b}{m} \leq \left(\frac{2s}{\log t'} + 1 \right) \cdot \frac{1}{34c_0^2 \cdot s\epsilon^{-2}\delta^{-1}} \leq \frac{\delta}{17c_0^2\epsilon^{-2}\log t'} + \frac{\delta}{34c_0^2 \cdot s\epsilon^{-2}} \leq \frac{3}{34}\delta.$$

Since $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in \mathsf{pr\text{-}BPP}$, there is no auxiliary-input one-way function by Lemma 4.6 and thus there exists the polynomial-time randomized algorithm UE in Lemma 6.5.

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

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

where $\bar{x}^{i'}_{<j'} = x_{(i'-1)\cdot w+1} \circ \cdots \circ x_{(i'-1)\cdot w+j'-1}$ and $\bar{x}^{<i'} = x_1 \circ \cdots \circ x_{(i'-1)\cdot w}$ (notice that $x_{<i} = \bar{x}^{<i'} \circ \bar{x}^{i'}_{<j'}$). Notice 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 $|x_{<i}|$ and the parameters $n, m, s, t, \epsilon^{-1}, \delta^{-1}, t^{\delta^{-1}}$. 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 $(w+1)b > m$,

$$w > \frac{m}{b} - 1 \geq \frac{34c_0^2 \cdot s\epsilon^{-2}\delta^{-1}}{2s/\log t' + 1} - 1 \geq 16c_0^2\epsilon^{-2}\delta^{-1}\log t'.$$

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 6.4 implies

$$\Pr_{i', \bar{x}^{<i'}, j', x^{i'}_{<j'}}\left[\mathrm{KL}\left(X_i | x_{<i} \middle\| \mathsf{Next}_n\left(\bar{x}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{x}^{<i'}}\right)\right) \leq \frac{\epsilon^2}{2}\right] \geq 1 - \frac{\delta}{4}. \tag{7}$$

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 4.17 and 6.3 implies that for every j',

$$\Pr_{i', \bar{x}^{<i'}, x^{i'}_{<j'}}\left[\mathrm{qcd}^{t''}(x^{i'}_{<j'} \mid \bar{x}_{<i}) \leq 3c_0\delta^{-1}\log t' + \log t' + c_0\right] \geq 1 - \frac{\delta}{3}. \tag{8}$$

When this occurs, by Lemma 6.5,

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

Therefore, if both the events in Equations (7) and (8) occur,

$$\Delta_{\mathrm{tv}}\left(L\left(x_{<i}; 1^{\langle n, m, s, t, \epsilon^{-1}, \delta^{-1}, t^{\delta^{-1}}\rangle}\right), X_i | x_{<i}\right) \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

By the union bound, we conclude that

$$\Pr_{i, x_{<i}}\left[\Delta_{\mathrm{tv}}\left(L\left(x_{<i}; 1^{\langle n, m, s, t, \epsilon^{-1}, \delta^{-1}, t^{\delta^{-1}}\rangle}\right), X_i | x_{<i}\right) \leq \epsilon\right] \geq 1 - \left(\frac{3}{34}\delta + \frac{\delta}{4} + \frac{\delta}{3}\right) > 1 - \delta.$$

□

7 Learning Distributions in Heuristica

In the following sections, we present the implications of Theorem 6.1 in learning distributions.

7.1 Learning Adaptively Changing Distributions for All Initial States

One intermediate consequence is learning adaptively changing distributions in the *worst-case* setting. The learning model was initiated by Naor and Rothblum [NR06] in the average-case setting and extended to the worst-case setting by Hirahara and Nanashima [HN23]. To state the learnability result formally, we introduce some notations and define their learning model.

An adaptively changing distribution (ACD) is a randomized Turing machine D satisfying the following syntax: For every sample size $n \in \mathbb{N}$,

1. D takes two inputs 1^n and $\sigma \in \{0,1\}^*$, where σ is called an internal state and initialized by some initial state $s_0 \in \{0,1\}^*$.
2. For any $\sigma \in \{0,1\}^*$, the algorithm $D(1^n, \sigma)$ randomly generates a sample $x \in \{0,1\}^*$ and a next state $s' \in \{0,1\}^*$ (x and s' can be correlated).

Then, any ACD D determines an example oracle $\mathsf{EX}_{n,s_0,D}$ for each sample size $n \in \mathbb{N}$ and each initial state s_0, as follows:

1. $\mathsf{EX}_{n,s_0,D}$ has a hidden internal state σ, which is initialized by s_0.
2. For each query access (without input), $\mathsf{EX}_{n,s_0,D}$ generates $(x, s') \leftarrow A(1^n, \sigma)$ and returns x as a sample. Then, $\mathsf{EX}_{n,s_0,D}$ updates the internal state σ as $\sigma := s'$.

For any functions $s(n)$ and $t(n)$, we say that an ACD D is $t(n)$-time samplable and has an $s(n)$-bit initial state if for every $n \in \mathbb{N}$ and every initial state $\sigma_0 \in \{0,1\}^{\leq s(n)}$, for every possible state σ in the execution with initial state σ_0, $D(1^n, \sigma)$ halts in $t(n)$ time (i.e., $\sigma \in \{0,1\}^{\leq t(n)}$).

In learning ACD D, a learner has query access to $\mathsf{EX}_{n,s_0,D}$ for a given parameter 1^n, where s_0 is a hidden initial state. The goal of the learner is to select some stage $i \geq 0$ and, after observing the first i samples $x^1, \ldots, x^i$ from $\mathsf{EX}_{n,s_0,D}$, to statistically simulate the conditional distribution of the next sample x^{i+1} given the initial state s_0 and $x^1, \ldots, x^i$. For convenience, we use the notation $D_i^{s_0}(x^1, \ldots, x^i)$ to refer to the conditional distribution that the learner attempts to simulate at stage i.

Definition 7.1 (Learning ACDs)**.** *Let $s(n)$ and $t(n)$ be polynomials. We say that $t(n)$-time samplable ACDs of $s(n)$-bit initial state are learnable if there exists a randomized algorithm L such that for every $t(n)$-time samplable ACD D of $s(n)$-bit initial state, every sufficiently large $n \in \mathbb{N}$, every $\epsilon^{-1}, \delta^{-1} \in \mathbb{N}$, the algorithm L satisfies the following with probability at least $1 - \delta$ over the choice of samples from $\mathsf{EX}_{n,s_0,D}$ and randomness for L:*

1. *$L^{\mathsf{EX}_{n,s_0,D}}(1^n, 1^{\epsilon^{-1}}, 1^{\delta^{-1}})$ obtains samples $x^1, x^2, \ldots,$ from $\mathsf{EX}_{n,s_0,D}$.*
2. *After obtaining i samples $x^1, \ldots, x^i$ (where i is selected by L), $L^{\mathsf{EX}_{n,s_0,D}}(1^n, 1^{\epsilon^{-1}}, 1^{\delta^{-1}})$ outputs some circuit h as a hypothesis without additional access to $\mathsf{EX}_{n,s_0,D}$.*
3. *The hypothesis h satisfies $\Delta_{\mathrm{tv}}(D_i^{s_0}(x^1, \ldots, x^i), h(r)) \leq \epsilon$, where r represents a uniformly random seed for h.*

We define the sample complexity $m(n, \epsilon, \delta)$ as the upper bound of the number of oracle accesses by $L(1^n, 1^{\epsilon^{-1}}, 1^{\delta^{-1}})$.

The implications for learning ACDs are stated below, restating Corollary 2.3 from Section 2.

Theorem 7.2. *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then for all polynomials* $s(n)$ *and* $t(n)$*, all* $t(n)$*-time samplable ACDs of* $s(n)$*-bit initial state are learnable in time* $(n\epsilon^{-1}\delta^{-1})^{O(\delta^{-1})}$ *with sample complexity* $O(s(n)\cdot\epsilon^{-2}\delta^{-1})$*.*

Proof. The theorem follows from Theorem 6.1. Let L be the algorithm in Theorem 6.1. We consider a program Π of size $s' = O(s+\log n)$ that embeds the target initial state $s_0 \in \{0,1\}^{\le s}$, a description of the sampler D, and the parameter $n \in \mathbb{N}$ (in binary). The program Π simulates the internal memory state and produces outputs from $\mathsf{EX}_{n,s_0,D}$ at each round. Then, we run L on the distribution generated by Π.

The learner for ACDs just selects the round i uniformly at random from $[m]$ where $m = cs'\epsilon^{-2}\delta^{-1}$ for a large enough universal constant $c > 0$, takes the samples $x_1,\dots,x_{i-1}$ from its oracle, and executes L on $x_{<i}$ and parameters $n, m, s', t, \epsilon^{-1}, \delta^{-1}, t^{\delta^{-1}}$, where the time-bound $t = \mathsf{poly}(n,s,m) = \mathsf{poly}(n,\epsilon^{-1},\delta^{-1})$ is set to be large enough so that Π can produce at least m samples. Then by Theorem 6.1, the total variation distance between the distributions of L and the i-th sample given $x_1,\dots,x_{i-1}$ is at most ϵ with probability at least $1-\delta$ over $x_{<i}$ and i. Thus, the learner that outputs a description of L that embeds $x_{<i}$ and the parameters satisfies the requirement of learning ACDs.

The sample complexity is at most $O((s+\log n)\epsilon^{-2}\delta^{-1}) = O(s\epsilon^{-2}\delta^{-1})$, where we assumed that $\log n \le s$. Otherwise, $2^s \le \mathsf{poly}(n)$ holds and we can learn every ACDs of s-bit initial states just by executing UE without any advice, taking time in $\mathsf{poly}(2^s) \le \mathsf{poly}(n)$. □

7.2 Worst-Case Distributional Learning from Independent Samples

One natural and special case of learning ACDs is *distributional learning*, initiated by Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie [KMRRSS94]. Here, the target of learning is an unknown distribution over strings, and the task of the learner is, given independently and identically distributed samples from the target, to produce a description of a distribution (in the form of circuits) that approximates the target with a small statistical error.

Since learning distributions from i.i.d. samples is regarded as a special case of learning ACDs where the initial state will never change throughout the learning process, Theorem 7.2 shows that all distributions samplable by a polynomial-time machine of description size s are learnable with the same time and sample complexity. In this section, we improve the time complexity to polynomial in n, ϵ^{-1} and δ^{-1} and sample complexity from $O(s\epsilon^{-2}\delta^{-1})$ to $O(s\epsilon^{-2}\log\delta^{-1})$ under the same assumption that $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP.

First, we present the learning model formally.

Definition 7.3 (Distributional Learning [KMRRSS94])**.** *Let* $\mathscr{D} = \{\mathscr{D}_n\}$ *be a class of distributions, where* $\mathscr{D}_n$ *is a set of distributions over* $\{0,1\}^n$ *for each* $n \in \mathbb{N}$*, and* $m\colon \mathbb{N}\times(0,1]\times(0,1] \to \mathbb{N}$ *be a function.*

The class $\mathscr{D}$ *is said to be distributionally learnable in polynomial time with query complexity* $m(n,\epsilon,\delta)$ *if there exists a polynomial-time oracle machine* L *such that for any* $n,\epsilon^{-1},\delta^{-1} \in \mathbb{N}$ *and any distribution* $\mathcal{D} \in \mathscr{D}_n$*, the algorithm* $L^{\mathcal{D}}(1^n, 1^{\epsilon^{-1}}, 1^{\delta^{-1}})$ *makes at most* $m(n,\epsilon,\delta)$ *query access to* $\mathcal{D}$ *and produces a description of circuit* h *such that* $\Delta_{\mathrm{tv}}(h(r),\mathcal{D}) \le \epsilon$*, where* r *is a random seed, with probability at least* $1-\delta$ *over the choice of samples and randomness for* L*.*

We now present the main result of this section, restated from Corollary 2.2 in Section 2.

Theorem 7.4. *If* $\mathrm{GapK}^{\mathrm{t}}$*-vs-*K $\in$ pr-BPP*, then for all polynomials* $s(n)$ *and* $t(n)$*, all distributions samplable by* $t(n)$*-time Turing machine of description length* $s(n)$ *is distributionally learnable in polynomial time with sample complexity* $O(s(n) \cdot \epsilon^{-2} \log \delta^{-1})$*.*

The improvement in time and sample complexity relies on the testability of the statistical distance between hypotheses, which is stated formally as follows:

Lemma 7.5. *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*.*

We may call ϵ and δ above an accuracy error and confidence error, respectively.

The proof of the lemma above is based on the same technique as that developed in the proof of [NR06, Theorem 4.1]. At a high level, we use the inverter for auxiliary-input functions to (distributionally) invert a function $f_{D_0,D_1}(b, r) = D_b(r)$, where $b \sim \{0, 1\}$ and D_0 and D_1 are regarded as auxiliary input. Then we empirically estimate how well a random label $b \sim \{0, 1\}$ is predictable only from $x \sim D_b(r)$ based on the (distributional) inverter and estimate the statistical distance from the estimated success probability of the prediction. For the formal proof, see Appendix A.

Now, we prove Theorem 7.4 based on Lemma 7.5.

Proof of Theorem 7.4. We apply the same learning algorithm L as that of learning ACDs in Theorem 7.2, where we set the parameters ϵ_L and δ_L for L as $\epsilon_L = \epsilon/5$ and $\delta_L = 1/4$ with respect to the parameter ϵ for distributional learning. Since distributional learning is a special case of learning ACDs, L produces a sampler h whose distribution is statistically close to the target distribution $\mathcal{D}$ within total variation distance at most $\epsilon/5$ with probability at least $3/4$. Notice that the time complexity of the learner is at most $\mathsf{poly}(n, \epsilon^{-1})$ and the sample complexity is at most $m_0 := m_0(s, \epsilon) = O(s\epsilon^{-2})$.

To reduce the confidence error from $1/4$ to the given parameter $\delta < 1/4$, we execute L repeatedly and obtain $N = \lceil 32 \ln \delta^{-1} \rceil$ hypotheses $h_1, \ldots, h_N$ by using independent choices of samples and randomness of L. By Hoeffding's inequality, with probability at least $1 - \delta/2$, the fraction of hypotheses that approximate $\mathcal{D}$ within distance $\epsilon/5$ is at least $3/4 - 1/8 = 5/8$. That is, a majority of the hypotheses are desirable.

The remaining task is to identify one such good hypothesis. For this, we use the statistical distance estimator $\tilde{\Delta}$ from Lemma 7.5 below. For each $i \in [N]$, let $\mathcal{H}_i$ denote the distribution defined by $h_i(r)$, where r is a random seed.

By applying $\tilde{\Delta}$ for every pair of (h_i, h_j) with $i < j$ with accuracy error $\epsilon/5$ and confidence error $\delta/(2N^2)$, we obtain the estimations $\tilde{\delta}_{i,j}$ of $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j)$. Then, we identify the maximum set $C \subseteq [N]$ such that

$$i, j \in C \text{ and } i < j \implies \tilde{\delta}_{i,j} \leq \frac{3}{5}\epsilon.$$

by solving maximum clique problem in time $O(2^N) = \mathsf{poly}(\delta^{-1})$. Then, we let the learning algorithm output h_i for arbitrary $i \in C$.

By the union bound, all the estimations of $\tilde{\Delta}$ are successfully performed with probability at least $1 - \delta/2$. Thus, with probability at least $1 - \delta$, (i) $|\tilde{\delta}_{i,j} - \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j)| \leq \epsilon/5$ for each $i < j$, and (ii) at least $5/8$-fraction of $i \in [N]$ satisfy $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \leq \epsilon/5$. To show the correctness of the algorithm, it suffices to show $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \leq \epsilon$ for all $i \in C$ under these conditions.

Let $G \subseteq [N]$ be the set of indices i such that $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \leq \epsilon/5$. Thus, all $i, j \in G$ with $i < j$ satisfy that

$$\tilde{\delta}_{i,j} \leq \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j) + \frac{\epsilon}{5} \leq \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) + \Delta_{\mathrm{tv}}(\mathcal{H}_j, \mathcal{D}) + \frac{\epsilon}{5} \leq \frac{3}{5}\epsilon.$$

Since $|G| > N/2$, the set G must be contained in the maximum set C, i.e., $G \subseteq C$.

Select $i \in C$ and $j \in G \subseteq C$ arbitrarily. We evaluate $\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D})$ as follows:

$$\Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{D}) \leq \Delta_{\mathrm{tv}}(\mathcal{H}_i, \mathcal{H}_j) + \Delta_{\mathrm{tv}}(\mathcal{H}_j, \mathcal{D}) \leq \left(\tilde{\delta}_{i,j} + \frac{\epsilon}{5}\right) + \frac{\epsilon}{5} \leq \frac{3}{5}\epsilon + \frac{\epsilon}{5} + \frac{\epsilon}{5} = \epsilon,$$

which completes the proof. □

8 Expected Loss Minimization in Heuristica

We discuss the consequence of our new inductive inference techniques for supervised learning.

Notations and Models. We use $Y = \{Y_n\}_{n\in\mathbb{N}}$ and $A = \{A_n\}_{n\in\mathbb{N}}$ to refer to label and action sets, respectively. In addition, we assume that for each $n \in \mathbb{N}$, $Y_n = \{0,1\}^{\beta(n)}$ and $A_n = \{0,1\}^{\alpha(n)}$ for polynomial-time computable functions $\beta, \alpha \colon \mathbb{N} \to \mathbb{N}$. For example, in the standard case of binary classification, $\alpha(n) = \beta(n) = 1$ for all $n \in \mathbb{N}$. In addition, let $X_n = \{0,1\}^n$ and $\mathscr{F}_n = \{f \colon X_n \to A_n\}$ for each $n \in \mathbb{N}$. We define a loss function as a family of functions $\ell = \{\ell_n\}$ such that $\ell_n \colon Y_n \times A_n \to \mathbb{R}_{\geq 0}$ for each $n \in \mathbb{N}$. For notational simplicity, we often drop the subscript n from A_n, Y_n, ℓ_n when n is clear in context.

The task of agnostic learning under a loss function ℓ is formulated as follows: A learner receives a sample set, where each sample $(x, y) \in X_n \times Y_n$ is drawn from an unknown distribution $\mathcal{D}$. Then, the learner produces a hypothesis $h \colon X_n \to A_n$ that approximates the best predictor in a concept class $\mathscr{C} = \{\mathscr{C}_n\}$, where $\mathscr{C}_n \subseteq \mathscr{F}_n$, that minimizes the expected loss for unseen data in future, specifically,

$$\mathrm{E}_{(x,y)\sim\mathcal{D}}\left[\ell(y, h(x))\right] \leq \min_{f^* \in \mathscr{C}_n} \mathrm{E}_{(x,y)\sim\mathcal{D}}\left[\ell(y, f^*(x))\right] + \epsilon,$$

where $\epsilon \in (0, 1]$ is a given accuracy parameter.

Note that for $A_n = Y_n = \{0,1\}$ and for the 0-1 loss defined as $\ell(y, a) = \mathbb{1}\{a \neq y\}$, the requirement above is paraphrased in the following familiar form:

$$\Pr_{(x,y)\sim\mathcal{D}_n}[h(x) \neq y] \leq \min_{f^* \in \mathscr{C}_n} \Pr_{(x,y)\sim\mathcal{D}_n}[f^*(x) \neq y] + \epsilon.$$

In this work, we allow the learner to produce only hypotheses represented as randomized circuits, to ensure their efficient computability. In addition, we consider the concept class as the whole class, i.e., $\mathscr{C}_n = \mathscr{F}_n$ for each $n \in \mathbb{N}$.

The requirement above is easily extended to non i.i.d. cases, where the task of learner is given samples $(x_1, b_1), \ldots, (x_{i-1}, y_{i-1})$ generated from an unknown environment (they are positively correlated) and produces a hypothesis h_i that minimizes the expected loss for the next i-th sample, i.e.,

$$\mathrm{E}_{(x_i,y_i)}\left[\ell(y_i, h(x_i))\right] \leq \min_{f^* \in \mathscr{C}_n} \mathrm{E}_{(x_i,y_i)}\left[\ell(y_i, f^*(x_i))\right] + \epsilon.$$

Next, we introduce some properties on loss functions.

For $\gamma \colon \mathbb{N} \to \mathbb{N}$, we say that a loss function ℓ is γ-bounded if $\ell_n(y, a) \in [0, \gamma(n)]$ for each $n \in \mathbb{N}$ and each $(y, a) \in Y_n \times A_n$. For instance, the 0-1 loss function is 1-bounded.

Definition 8.1 (Loss-Minimizer)**.** *We define a loss-minimizer M for a loss function ℓ as a randomized oracle machine such that for every $n, \epsilon^{-1} \in \mathbb{N}$ and every distribution $\mathcal{Y}$ over Y_n,*

$$\mathrm{E}_{M,y\sim\mathcal{Y}}\left[\ell(y, M^{\mathcal{Y}}(1^n, 1^{\epsilon^{-1}}))\right] \leq \min_{a\in A_n} \mathrm{E}_{y\sim\mathcal{Y}}\left[\ell(y,a)\right] + \epsilon.$$

Based on the standard empirical estimation, we can observe that a natural class of loss functions, including 0-1 loss, admits a polynomial-time loss-minimizer.

Proposition 8.2. *If action sets are polynomial size (i.e., $|A_n| \leq \mathsf{poly}(n)$), then all γ-bounded polynomial-time-coumutable loss functions admit polynomial-time loss-minimizers for all polynomials $\gamma(n)$.*

For the formal proof of Proposition 8.2, see the proof of [HN23, Theorem 10.8].

8.1 Sequential Expected Loss Minimization

In this section, we prove that minimizing loss function according to the context-restricting universal distribution indeed approximates the best predictor in the whole class $\mathscr{F}$ of predictors. As a result, we will derive the following theorem.

Theorem 8.3. *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in \mathsf{pr\text{-}BPP}$*, then for every γ-bounded loss function ℓ that admits a polynomial-time loss-minimizer, there exists a polynomial-time randomized algorithm P such that for every $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ and every s-size randomized program Π that produces at least m samples $(x_1, y_1), \ldots, (x_m, y_m), \ldots \in X_n \times Y_n$ in t $(\geq s)$ time, if $m \geq O(s\epsilon^{-2}\delta^{-1}\gamma(n)^2)$, then*

$$\Pr_{i\sim[m],(xy)_{<i}}\left[\mathrm{E}_{P,(x_i,y_i)}\left[\ell(y, P(x_i; (xy)_{<i}, 1^{\mathsf{param}}))\right] \leq \min_{f^*\in\mathscr{F}_n} \mathrm{E}_{(x_i,y_i)}\left[\ell(y_i, f^*(x_i))\right] + \epsilon\right] \geq 1-\delta,$$

where $(xy)_{<i}$ represents a sample set $\{(x_1, y_1), \ldots, (x_{i-1}, y_{i-1})\}$ selected according to Π, and $1^{\mathsf{param}} := 1^{\langle n,m,s,t,\epsilon^{-1},\delta^{-1},\gamma(n),t^{\delta^{-1}}\rangle}$.

Note that Corollary 2.5 in Section 2 follows directly from Theorem 8.3, combined with Proposition 8.2.

To prove the theorem, we begin by observing that minimizing loss with respect to a distribution that is statistically close to the true label distribution still yields an approximation of the optimal predictor.

Lemma 8.4. *For every $n, m, \epsilon^{-1} \in \mathbb{N}$, every γ-bounded loss function ℓ, every loss-minimizer M_ℓ for ℓ, and every distributions $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{X} \times \tilde{\mathcal{Y}}$ over $X_n \times Y_n$, if*

$$\mathrm{E}_{x\sim\mathcal{X}}\left[\Delta_{\mathrm{tv}}(\tilde{\mathcal{Y}}_x, \mathcal{Y}_x)\right] \leq \frac{\epsilon}{8\gamma},$$

where $\mathcal{Y}_x$ and $\tilde{\mathcal{Y}}_x$ are conditional distributions of $\mathcal{Y}$ and $\tilde{\mathcal{Y}}$ given $x \sim \mathcal{X}$, respectively, then

$$\mathrm{E}_{M_\ell,(x,y)\sim\mathcal{X}\times\mathcal{Y}}\left[\ell(y, M_\ell^{\tilde{\mathcal{Y}}_x}(1^n, 1^{4\epsilon^{-1}}))\right] \leq \min_{f^*\in\mathcal{F}_n} \mathrm{E}_{(x,y)\sim\mathcal{X}\times\mathcal{Y}}\left[\ell(y, f^*(x))\right] + \epsilon.$$

Proof. The lemma is verified as the following calculation:

$$
\begin{aligned}
&\mathrm{E}_{M_\ell,(x,y)\sim\mathcal{X}\times\mathcal{Y}}\left[\ell(y, M_\ell^{\tilde{\mathcal{Y}}_x}(1^n, 1^{4\epsilon^{-1}}))\right] \\
&\le \mathrm{E}_x\left[\sum_y \mathcal{Y}_x(y)\mathrm{E}_{M_\ell}[\ell(y, M_\ell^{\tilde{\mathcal{Y}}_x}(1^n, 1^{4\epsilon^{-1}}))]\right] \\
&\le \mathrm{E}_x\left[\sum_y \left(\left|\mathcal{Y}_x(y) - \tilde{\mathcal{Y}}_x(y)\right| + \tilde{\mathcal{Y}}_x(y)\right)\mathrm{E}_{M_\ell}[\ell(y, M_\ell^{\tilde{\mathcal{Y}}_x}(1^n, 1^{4\epsilon^{-1}}))]\right] \\
&\le \gamma\cdot\mathrm{E}_x\left[\sum_y \left|\mathcal{Y}_x(y) - \tilde{\mathcal{Y}}_x(y)\right|\right] + \mathrm{E}_x\left[\mathrm{E}_{y\sim\tilde{\mathcal{Y}}_x,M_\ell}\left[\ell(y, M_\ell^{\tilde{\mathcal{Y}}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&\le \gamma\cdot 2\cdot\frac{\epsilon}{8\gamma} + \mathrm{E}_x\left[\mathrm{E}_{y\sim\tilde{\mathcal{Y}}_x,M_\ell}\left[\ell(y, M_\ell^{\tilde{\mathcal{Y}}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&\le \frac{\epsilon}{4} + \mathrm{E}_x\left[\min_{a\in A_n}\mathrm{E}_{y\sim\tilde{\mathcal{Y}}_x}\left[\ell(y,a)\right] + \frac{\epsilon}{4}\right] \\
&\le \frac{\epsilon}{2} + \mathrm{E}_x\left[\mathrm{E}_{y\sim\tilde{\mathcal{Y}}_x,M_\ell}\left[\ell(y, M_\ell^{\mathcal{Y}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&= \frac{\epsilon}{2} + \mathrm{E}_x\left[\sum_y \tilde{\mathcal{Y}}_x(y)\mathrm{E}_{M_\ell}\left[\ell(y, M_\ell^{\mathcal{Y}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&\le \frac{\epsilon}{2} + \mathrm{E}_x\left[\sum_y \left(\left|\mathcal{Y}_x(y) - \tilde{\mathcal{Y}}_x(y)\right| + \mathcal{Y}_x(y)\right)\mathrm{E}_{M_\ell}\left[\ell(y, M_\ell^{\mathcal{Y}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&\le \frac{\epsilon}{2} + \gamma\cdot\mathrm{E}_x\left[\sum_y \left|\mathcal{Y}_x(y) - \tilde{\mathcal{Y}}_x(y)\right|\right] + \mathrm{E}_x\left[\sum_y \mathcal{Y}_x(y)\mathrm{E}_{M_\ell}\left[\ell(y, M_\ell^{\mathcal{Y}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&\le \frac{\epsilon}{2} + \gamma\cdot 2\cdot\frac{\epsilon}{8\gamma} + \mathrm{E}_x\left[\mathrm{E}_{y\sim\mathcal{Y}_x,M_\ell}\left[\ell(y, M_\ell^{\mathcal{Y}_x}(1^n, 1^{4\epsilon^{-1}}))\right]\right] \\
&\le \frac{3\epsilon}{4} + \mathrm{E}_x\left[\min_{a\in A_n}\mathrm{E}_{y\sim\mathcal{Y}_x}\left[\ell(y,a)\right] + \frac{\epsilon}{4}\right] \\
&\le \mathrm{E}_x\left[\min_{a\in A_n}\mathrm{E}_{y\sim\mathcal{Y}_x}\left[\ell(y,a)\right]\right] + \epsilon \\
&= \min_{f^*\in\mathcal{F}_n}\mathrm{E}_{(x,y)\sim\mathcal{X}\times\mathcal{Y}}\left[\ell(y, f^*(x))\right] + \epsilon.
\end{aligned}
$$

□

Next, we apply the context-restricted universal extrapolation framework to construct a sampler that approximates the label distribution needed in Lemma 8.4.

Lemma 8.5. *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then there exists a polynomial-time randomized algorithm* L *such that for every* $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ *and every* s*-size randomized program* Π *that produces at least* m *pairs of strings* $(x_1, y_1), \ldots, (x_m, y_m), \ldots \in X_n \times Y_n$ *in* t $(\ge s)$ *time, if* $m \ge O(s\epsilon^{-2}\delta^{-1})$*, then*

$$
\Pr_{i\sim[m],(xy)_{<i}}\left[\mathrm{E}_{x_i}\left[\Delta_{\mathrm{tv}}\left(L\left((xy)_{<i}, x_i, 1^{\langle n,s,t,\epsilon^{-1},\delta^{-1},t^{\delta^{-1}}\rangle}\right), \mathcal{Y}_{i|<i}\right)\right] \le \epsilon\right] \ge 1-\delta,
$$

where $(xy)_{<i}$ *represents* $((x_1, y_1), \ldots, (x_{i-1}, y_{i-1}))$ *selected according to* Π*, and* $\mathcal{Y}_{i|<i}$ *represents the distribution of* y_i *given the previous strings* $(xy)_{<i}$ *and* x_i*.*

Since the proof of Lemma 8.5 closely mirrors that of Theorem 6.1, we defer it to Section 8.3.

We now complete the proof of Theorem 8.3 based on the lemmas established above.

Proof of Theorem 8.3. Let L be the algorithm in Lemma 8.5 and M_ℓ be the polynomial-time loss-minimizer for the loss function ℓ. Let $\gamma := \gamma(n)$.

For given input $x_i, (xy)_{<i}$, and parameters $n, m, s, t, \epsilon^{-1}, \delta^{-1}, \gamma, t^{\delta^{-1}}$ (represented in unary), the predictor P just executes $M_\ell(1^n, 1^{4\epsilon^{-1}})$, where P answers the query using samples from the distribution $\tilde{\mathcal{Y}}_{(xy)_{<i},x_i}$, defined as the output distribution of

$$L\left((xy)_{<i}, x_i; 1^{\langle n,s,t,8\epsilon^{-1}\gamma,\delta^{-1},t^{\delta^{-1}}\rangle}\right).$$

From Lemma 8.5, if $m \geq O(s\epsilon^{-2}\delta^{-1}\gamma^2)$, then

$$\Pr_{i\sim[m],(xy)_{<i}}\left[\mathrm{E}_{x_i}\left[\Delta_{\mathrm{tv}}\left(\tilde{\mathcal{Y}}_{(xy)_{<i},x_i}, \mathcal{Y}_{i|<i}\right)\right] \leq \frac{\epsilon}{8\gamma}\right] \geq 1-\delta.$$

Under this event, from Lemma 8.4,

$$\mathrm{E}_{M_\ell,(x_i,y_i)}\left[\ell(y_i, M_\ell^{\tilde{\mathcal{Y}}_{(xy)_{<i},x_i}}(1^n, 1^{4\epsilon^{-1}}))\right] \leq \min_{f^*\in\mathcal{F}_n} \mathrm{E}_{(x_i,y_i)}\left[\ell(y_i, f^*(x_i))\right] + \epsilon.$$

Thus, we conclude that

$$\Pr_{i\sim[m],(xy)_{<i}}\left[\mathrm{E}_{P,(x_i,y_i)}\left[\ell(y, P(x_i; (xy)_{<i}, 1^{\mathsf{param}}))\right] \leq \min_{f^*\in\mathscr{F}_n} \mathrm{E}_{(x_i,y_i)}\left[\ell(y_i, f^*(x_i))\right] + \epsilon\right] \geq 1-\delta.$$

□

8.2 Example: Worst-Case Binary Classification with Independent Noise

In this section, we specifically apply Theorem 8.3 in the case of binary classification from i.i.d. samples. The following is a restatement of Corollary 2.4 from Section 2.

Theorem 8.6. *If* $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP*, then there exist a polynomial-time randomized oracle machine* L *and a function* $m\colon \mathbb{N}\times\mathbb{N}\times(0,1]\times(0,1]\to\mathbb{N}$ *with* $m(s,a,\epsilon,\delta) = O((s+a)\epsilon^{-2}\log\delta^{-1})$ *such that for all* $n, s, a, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ *and all distribution* $\mathcal{D} = (\mathcal{X}, \mathcal{B})$ *over* $\{0,1\}^n \times \{0,1\}$*, if*

- *the example distribution* $\mathcal{X}$ *is samplable by a* t*-time* a*-size randomized program, and*
- *the label distribution* $\mathcal{B}$ *given* $x \sim \mathcal{X}$ *is statistically equivalent to a distribution of* $f(x)+\xi$*, where* $\xi \sim \mathrm{Ber}(\eta)$*, for some function* $f\colon\{0,1\}^n\to\{0,1\}$ *computable by a* t*-time* s*-size program and parameter* $\eta \in [0, 1/2]$*,*

then

$$\Pr_{S^m,L}\left[h \leftarrow L(S^m; 1^{\langle n,\epsilon^{-1},\delta^{-1}\rangle}, 1^{\langle s,a,t\rangle}) \text{ and } \Pr_{(x,b)\sim\mathcal{D},h}[h(x)\neq b] \leq \eta+\epsilon\right] \geq 1-\delta,$$

where S^m *is a sample set of size* $m(s,a,\epsilon,\delta) = O((s+a)\cdot\epsilon^{-2}\log\delta^{-1})$ *drawn i.i.d. from* $\mathcal{D}$*.*

Proof. It suffices to construct a learning algorithm L with time complexity $(n\epsilon^{-1}\delta^{-1}sat)^{O(\delta^{-1})}$ and sample complexity $O((s+a)\cdot\epsilon^{-2}\delta^{-1})$. The improvement to polynomial time and sample complexity $O((s+a)\cdot\epsilon^{-2}\log\delta^{-1})$ follows from the well-known technique of [HKLW88]: we run the base learner

repeatedly with a *constant* confidence parameter for $O(\log \delta^{-1})$ times, empirically evaluate the error of each hypothesis, and output the best-performing one.

Without loss of generality, we assume that $\log \epsilon^{-1} \leq s$. Otherwise, $2^s < \epsilon^{-1}$ and we can find the best hypothesis of description size s by exhaustive search in time $O(2^s) \leq \mathsf{poly}(\epsilon^{-1})$. Based on the same argument, we also assume that $\log \delta^{-1} \leq s$ and $\log a \leq s$.

Let $\tilde{\eta} \in [0, 1/2]$ be the real value specified with the first $7s$ bits in the binary representation of $\eta \in [0, 1/2]$. Then, it holds that $\eta - \tilde{\eta} \leq 2^{-7s}$.

Let P denote the predictor guaranteed by Theorem 8.3 when applied with the 0-1 loss function (which admits a polynomial-time loss-minimizer by Proposition 8.2). We consider the distribution $\tilde{\mathcal{D}}$ of (x, b), where $x \sim \mathcal{X}$ and $b = f(x) \oplus \xi$ for $\xi \sim \mathrm{Ber}(\tilde{\eta})$. Notice that $\tilde{\mathcal{D}}$ is samplable by $O(t+s)$-time program $\tilde{\Pi}$ specified with the sampler for $\mathcal{X}$, f, and $\tilde{\eta}$. Namely, for an absolute constant $c > 0$, $\tilde{\Pi}$ is described in $c(s+a)$ bits, and $m = O((s+a)\epsilon^{-2}\delta^{-1})$ samples are sufficient for the predictor P to succeed to minimize the expected loss under samples generated from $\tilde{\mathcal{D}}$ with accuracy $\epsilon/2$ and confidence $\delta/2$. Let $\tilde{t}$ be the time required for $\tilde{\Pi}$ to generate m samples. Let τ be a large enough polynomial such that $\tau(n, \epsilon^{-1}, \delta^{-1}, s, a, t) \geq \tilde{t}$ for any choice of $n, \epsilon^{-1}, \delta^{-1}, s, a, t$.

Furthermore, it is easily checked that $\Delta_{\mathrm{tv}}(\mathcal{D}, \tilde{\mathcal{D}}) \leq 2^{-7s}$ as

$$\Delta_{\mathrm{tv}}(\mathcal{D}, \tilde{\mathcal{D}}) = \frac{1}{2}\sum_{x,b} \mathcal{X}(x) \left| \Pr_{\xi \sim \mathrm{Ber}(\eta)}[\xi \oplus f(x) = b] - \Pr_{\tilde{\xi} \sim \mathrm{Ber}(\tilde{\eta})}[\tilde{\xi} \oplus f(x) = b] \right| = |\eta - \tilde{\eta}| \leq 2^{-7s}.$$

Now, we present the learner L. On given parameters $n, \epsilon^{-1}, \delta^{-1}, s, a, t \in \mathbb{N}$ and m samples $S^m = \{(x_1, b_1), \ldots, (x_m, b_m)\}$ from $\mathcal{D}$, the learner L selects $i \sim [m]$ and outputs the hypothesis

$$h(x) := P(x; (xb)_{<i}, 1^{\mathsf{param}}),$$

where $(xb)_{<i} = \{(x_1, b_1), \ldots, (x_{i-1}, b_{i-1})\}$ and

$$\mathsf{param} := \langle n, m, c(s+a), \tau(n, \epsilon^{-1}, \delta^{-1}, s, a, t), 2\epsilon^{-1}, 2\delta^{-1}, 1, \tau(n, \epsilon^{-1}, \delta^{-1}, s, a, t)^{2\delta^{-1}} \rangle.$$

First, suppose that the set $\tilde{S}^m$ of m samples are generated from $\tilde{\mathcal{D}}$ instead of the true distribution $\mathcal{D}$. In this case, L executes P in the setting where the requirements in Theorem 8.3 are satisfied, resulting in

$$\Pr_{\tilde{S}^m, L}\left[\Pr_{(x,b)\sim\tilde{\mathcal{D}}, h}[h(x) \neq b] \leq \min_{\tilde{f}^* \in \mathcal{F}_n} \Pr_{(x,b)\sim\tilde{\mathcal{D}}}\left[\tilde{f}^*(x) \neq b\right] + \frac{\epsilon}{2}\right] \geq 1 - \frac{\delta}{2}.$$

From the triangle inequality,

$$\Delta_{\mathrm{tv}}(\mathcal{D}^m, \tilde{\mathcal{D}}^m) \leq m \cdot \Delta_{\mathrm{tv}}(\mathcal{D}, \tilde{\mathcal{D}}) \leq \frac{O((s+a)\epsilon^{-2}\delta^{-2})}{2^{7s}} \leq \frac{O(s)\cdot 2^{5s}}{2^{7s}} \leq \frac{1}{2\cdot 2^s} \leq \frac{\delta}{2},$$

for all large enough $s \in \mathbb{N}$.

Thus,

$$\begin{aligned}
&\Pr_{S^m \sim \mathcal{D}^m, L}\left[\Pr_{(x,b)\sim\tilde{\mathcal{D}}, h}[h(x) \neq b] \leq \min_{\tilde{f}^* \in \mathcal{F}_n} \Pr_{(x,b)\sim\tilde{\mathcal{D}}}\left[\tilde{f}^*(x) \neq b\right] + \frac{\epsilon}{2}\right] \qquad (9)\\
&\geq \Pr_{\tilde{S}^m \sim \tilde{\mathcal{D}}^m, L}\left[\Pr_{(x,b)\sim\tilde{\mathcal{D}}, h}[h(x) \neq b] \leq \min_{\tilde{f}^* \in \mathcal{F}_n} \Pr_{(x,b)\sim\tilde{\mathcal{D}}}\left[\tilde{f}^*(x) \neq b\right] + \frac{\epsilon}{2}\right] - \Delta_{\mathrm{tv}}(\mathcal{D}^m, \tilde{\mathcal{D}}^m)\\
&\geq 1 - \frac{\delta}{2} - \frac{\delta}{2} = 1 - \delta.
\end{aligned}$$

Let h be a hypothesis satisfying the event in (9) and $f^* \in \mathcal{F}_n$ be the best predictor under $\mathcal{D}$, i.e.,

$$\Pr_{(x,b)\sim\mathcal{D}}[f^*(x) \neq b] = \min_{f^*\in\mathcal{F}_n} \Pr_{(x,b)\sim\mathcal{D}}[f^*(x) \neq b].$$

Then,

$$\begin{aligned}
\Pr_{(x,b)\sim\mathcal{D},h}[h(x) \neq b] &\leq \Pr_{(x,b)\sim\tilde{\mathcal{D}},h}[h(x) \neq b] + \Delta_{\mathrm{tv}}(\mathcal{D},\tilde{\mathcal{D}}) \\
&\leq \min_{\tilde{f}^*\in\mathcal{F}_n} \Pr_{(x,b)\sim\tilde{\mathcal{D}}}\left[\tilde{f}^*(x) \neq b\right] + \frac{\epsilon}{2} + \Delta_{\mathrm{tv}}(\mathcal{D},\tilde{\mathcal{D}}) \\
&\leq \Pr_{(x,b)\sim\tilde{\mathcal{D}}}[f^*(x) \neq b] + \frac{\epsilon}{2} + \Delta_{\mathrm{tv}}(\mathcal{D},\tilde{\mathcal{D}}) \\
&\leq \Pr_{(x,b)\sim\mathcal{D}}[f^*(x) \neq b] + \frac{\epsilon}{2} + 2\Delta_{\mathrm{tv}}(\mathcal{D},\tilde{\mathcal{D}}) \\
&= \min_{f^*\in\mathcal{F}_n} \Pr_{(x,b)\sim\mathcal{D}}[f^*(x) \neq b] + \frac{\epsilon}{2} + 2\Delta_{\mathrm{tv}}(\mathcal{D},\tilde{\mathcal{D}}) \\
&\leq \min_{f^*\in\mathcal{F}_n} \Pr_{(x,b)\sim\mathcal{D}}[f^*(x) \neq b] + \frac{\epsilon}{2} + \frac{2}{2^{7s}} \\
&\leq \min_{f^*\in\mathcal{F}_n} \Pr_{(x,b)\sim\mathcal{D}}[f^*(x) \neq b] + \epsilon \\
&\leq \Pr_{(x,b)\sim\mathcal{D}}[f(x) \neq b] + \epsilon \\
&= \eta + \epsilon.
\end{aligned}$$

Therefore, we conclude that

$$\Pr_{S^m\sim\mathcal{D}^m,L}\left[\Pr_{(x,b)\sim\mathcal{D},h}[h(x) \neq b] \leq \eta + \epsilon\right] \geq 1 - \delta.$$

□

8.3 Proof of Lemma 8.5

Lemma (A reminder of Lemma 8.5)**.** *If* $\mathrm{GapK^t}$-vs-K $\in$ pr-BPP*, then there exists a polynomial-time randomized algorithm* L *such that for every* $n, m, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ *and every* s*-size randomized program* Π *that produces at least* m *pairs of strings* $(x_1, y_1), \ldots, (x_m, y_m), \ldots \in X_n \times Y_n$ *in* t $(\geq s)$ *time, if* $m \geq O(s\epsilon^{-2}\delta^{-1})$*, then*

$$\Pr_{i\sim[m],(xy)_{<i}}\left[\mathrm{E}_{x_i}\left[\Delta_{\mathrm{tv}}\left(L\left((xy)_{<i}, x_i, 1^{\langle n,s,t,\epsilon^{-1},\delta^{-1},t^{\delta^{-1}}\rangle}\right), \mathcal{Y}_{i|<i}\right)\right] \leq \epsilon\right] \geq 1 - \delta,$$

where $(xy)_{<i}$ *represents* $((x_1, y_1), \ldots, (x_{i-1}, y_{i-1}))$ *selected according to* Π*, and* $\mathcal{Y}_{i|<i}$ *represents the distribution of* y_i *given the previous strings* $(xy)_{<i}$ *and* x_i*.*

Proof. Let p_0 and c_0 be the maximum of the polynomials and constants in Lemmas 4.17, 6.3 and 6.4, respectively. Let p_1 be a large enough polynomial we specify later. For each $t \in \mathbb{N}$, let $t' := p_0(t)$, $t'' := p_0(t')$, and $t''' := p_1(t'')$ below. Without loss of generality, we assume that, $t''' \geq t'' \geq t' \geq 2$, $c_0 \geq 1$, and $c_0 \log t' \geq \log t''$ for each t.

For each $n, \beta, s, t, \epsilon^{-1}, \delta^{-1} \in \mathbb{N}$ with $m \geq 2050c_0^2 \cdot s\epsilon^{-2}\delta^{-1}$, let $b = \lceil 2s/\log t' \rceil$. 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

context-restricting universal extrapolation as in Lemma 6.4 on $\tilde{m}$ strings $x_1 \circ y_1, \ldots, x_{\tilde{m}} \circ y_{\tilde{m}}$. For simplicity, let $z_i = x_i \circ y_i$.

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 $\mathsf{poly}(2^s, \epsilon^{-1}) \leq \mathsf{poly}(\epsilon^{-1}, \delta^{-1})$ (this is exactly the same setting as [HN23]) to extrapolate a prefix string produced by an s-size program under Q^t within statistical error ϵ.

From the same argument as that of Theorem 6.1, we have

$$\Pr_{i \sim [m]}[i > \tilde{m}] < \frac{3}{34}\delta.$$

Since $\mathrm{GapK}^{\mathrm{t}}$-vs-K $\in$ pr-BPP, there is no auxiliary-input one-way function by Lemma 4.6 and thus there exists the polynomial-time randomized algorithm UE in Lemma 6.5.

Now, we present the construction of the algorithm L based on UE. On input $z_{<i}$, x_i, $i \sim [m]$, and parameters $n, s, t, \epsilon^{-1}, \delta^{-1}, t^{\delta^{-1}}$ (given in unary), the algorithm L calculates $\beta := \beta(n), t', t'', t''', w, b$, and $\tilde{m}$ as above. If $i > \tilde{m}$, the algorithm L halts with arbitrary message (we do not care this case as discussed above). Otherwise if $i \leq \tilde{m}$, the algorithm L calculates the unique pair $(i', j') \in [b] \times [w]$ such that $i = (i' - 1) \cdot w + j'$ and outputs a sample from

$$\mathsf{UE}\left(\bar{z}^{i'}_{<j'} x_i; \bar{z}^{<i'}, 1^{\langle \beta, t''', 4\epsilon^{-1}, p_1(\epsilon^{-1} t^{\delta^{-1}}) \rangle}\right),$$

where $\bar{z}^{i'}_{<j'} = z_{(i'-1)\cdot w+1} \circ \cdots \circ z_{(i'-1)\cdot w+j'-1}$ and $\bar{z}^{<i'} = z_1 \circ \cdots \circ z_{(i'-1)\cdot w}$. Notice that $z_{<i} = \bar{z}^{<i'} \circ \bar{z}^{i'}_{<j'}$.

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

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

$$w > \frac{m}{b} - 1 \geq \frac{2050 c_0^2 \cdot s\epsilon^{-2}\delta^{-1}}{2s/\log t' + 1} - 1 \geq 1024 c_0^2 \epsilon^{-2} \delta^{-1} \log t'.$$

Since $b \geq s/\log t' \geq s/\log t''$ and $w \geq 1024 c_0^2 \epsilon^{-2}\delta^{-1} \log t' \geq 2c_0 \cdot 128\epsilon^{-2} \cdot 4\delta^{-1} \cdot \log t''$, Lemma 6.4 implies

$$\Pr_{i', \bar{z}^{<i'}, j', z^{i'}_{<j'}}\left[\mathrm{KL}\left(\mathcal{X}_i\mathcal{Y}_i | z_{<i} \middle\| \mathsf{Next}_{n+\beta}\left(\bar{z}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{z}^{<i'}}\right)\right) \leq \frac{\epsilon^2}{128}\right] \geq 1 - \frac{\delta}{4}, \tag{10}$$

where $\mathcal{X}_i$ and $\mathcal{Y}_i$ represents the distributions of x_i and y_i, respectively.

By the chain rule of KL divergence,

$$\begin{aligned}
&\mathrm{KL}\left(\mathcal{X}_i\mathcal{Y}_i | z_{<i} \middle\| \mathsf{Next}_{n+\beta}\left(\bar{z}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{z}^{<i'}}\right)\right) \\
&= \mathrm{KL}\left(\mathcal{X}_i | z_{<i} \middle\| \mathsf{Next}_n\left(\bar{z}^{i'}_{<j'}; \mathrm{Q}^{t'', \bar{z}^{<i'}}\right)\right) + \mathrm{E}_{x_i \sim \mathcal{X}_i | z_{<i}}\left[\mathrm{KL}\left(\mathcal{Y}_i | z_{<i} x_i \middle\| \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'} x_i; \mathrm{Q}^{t'', \bar{z}^{<i'}}\right)\right)\right] \\
&\geq \mathrm{E}_{x_i \sim \mathcal{X}_i | z_{<i}}\left[\mathrm{KL}\left(\mathcal{Y}_i | z_{<i} x_i \middle\| \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'} x_i; \mathrm{Q}^{t'', \bar{z}^{<i'}}\right)\right)\right].
\end{aligned}$$

If the event in Equation (10) occurs, it holds that

$$\begin{aligned}
&\mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\Delta_{\mathrm{tv}}\left(\mathcal{Y}_i|z_{<i}x_i,\mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i;\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\right]\\
&\le \mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\sqrt{2^{-1}\cdot\mathrm{KL}\left(\mathcal{Y}_i|z_{<i}x_i\middle\|\mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i;\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)}\right]\\
&\le 2^{-1/2}\sqrt{\mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\mathrm{KL}\left(\mathcal{Y}_i|z_{<i}x_i\middle\|\mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i;\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\right]}\\
&\le 2^{-1/2}\sqrt{\mathrm{KL}\left(\mathcal{X}_i\mathcal{Y}_i|z_{<i}\middle\|\mathsf{Next}_{n+\beta}\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)}\\
&\le 2^{-1/2}\sqrt{\frac{\epsilon^2}{128}}=\frac{\epsilon}{16},
\end{aligned}$$

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

We also obtain that

$$\begin{aligned}
&\Delta_{\mathrm{tv}}\left(\mathcal{X}_i|z_{<i},\mathsf{Next}_n\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\\
&\le \Delta_{\mathrm{tv}}\left(\mathcal{X}_i\mathcal{Y}_i|z_{<i},\mathsf{Next}_{n+\beta}\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\\
&\le 2^{-1/2}\sqrt{\mathrm{KL}\left(\mathcal{X}_i\mathcal{Y}_i|z_{<i}\middle\|\mathsf{Next}_{n+\beta}\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)}\\
&\le 2^{-1/2}\sqrt{\frac{\epsilon^2}{128}}=\frac{\epsilon}{16}.
\end{aligned}$$

Since $b\ge 2s/\log t'\ge(s+c_0\log\delta^{-1})/\log t'$, Lemmas 4.17 and 6.3 implies that for every j',

$$\Pr_{i',\bar{z}^{<i'},z^{i'}_{<j'}}\left[\mathrm{qcd}^{t''}(\bar{z}^{i'}_{<j'}\mid\bar{z}^{<i'})\le 3c_0\delta^{-1}\log t'+\log t'+c_0\right]\ge 1-\frac{\delta}{3}. \tag{11}$$

Below, we assume that both events in Equations (10) and (11) occur. By Lemma 6.5,

$$\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{z}^{i'}_{<j'};\bar{z}^{<i'},1^{\langle n,t'',16\epsilon^{-1},2^{c_0}t'(t')^{3c_0\delta^{-1}}\rangle}\right),\mathsf{Next}_n\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\le\frac{\epsilon}{16}.$$

Let $\tilde{\mathcal{X}}_i$ denote the distribution of $\mathsf{UE}\left(\bar{z}^{i'}_{<j'};\bar{z}^{<i'},1^{\langle n,t'',16\epsilon^{-1},2^{c_0}t'(t')^{3c_0\delta^{-1}}\rangle}\right)$. Then,

$$\begin{aligned}
\Delta_{\mathrm{tv}}\left(\tilde{\mathcal{X}}_i,\mathcal{X}_i|z_{<i}\right)&\le\Delta_{\mathrm{tv}}\left(\tilde{\mathcal{X}}_i,\mathsf{Next}_n\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)+\Delta_{\mathrm{tv}}\left(\mathcal{X}_i|z_{<i},\mathsf{Next}_n\left(\bar{z}^{i'}_{<j'};\mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\\
&\le\frac{\epsilon}{16}+\frac{\epsilon}{16}=\frac{\epsilon}{8}.
\end{aligned}$$

By Lemma 4.16, for a large enough polynomial p_1 (recall that $t'''=p_1(t'')$),

$$\begin{aligned}
&\Pr_{x_i\sim\tilde{\mathcal{X}}_i}\left[\mathrm{qcd}^{t'''}(\bar{z}^{i'}_{<j'}x_i\mid\bar{z}^{<i'})\le\log p_1(\epsilon^{-1}t^{\delta^{-1}})\right]\\
&\ge\Pr_{x_i\sim\tilde{\mathcal{X}}_i}\left[\mathrm{qcd}^{t'''}(\bar{z}^{i'}_{<j'}x_i\mid\bar{z}^{<i'})\le 3c_0\delta^{-1}\log t'+\log t'+2c_0+\log p_0(8t''\epsilon^{-1})\right]\\
&\ge\Pr_{x_i\sim\tilde{\mathcal{X}}_i}\left[\mathrm{qcd}^{t'''}(\bar{z}^{i'}_{<j'}x_i\mid\bar{z}^{<i'})\le\mathrm{qcd}^{t''}(z^{i'}_{<j'}\mid\bar{z}^{<i'})+\log p_0(8t''\epsilon^{-1})+c_0\right]\\
&\ge 1-\frac{\epsilon}{8}.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\Pr_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\mathrm{qcd}^{t'''}(\bar{z}^{i'}_{<j'}x_i \mid \bar{z}^{<i'}) \le \log p_1(\epsilon^{-1}t^{\delta^{-1}})\right]\\
&\ge \Pr_{x_i\sim\tilde{\mathcal{X}}_i}\left[\mathrm{qcd}^{t'''}(\bar{z}^{i'}_{<j'}x_i \mid \bar{z}^{<i'}) \le \log p_1(\epsilon^{-1}t^{\delta^{-1}})\right] - \Delta_{\mathrm{tv}}\left(\tilde{\mathcal{X}}_i, \mathcal{X}_i|z_{<i}\right)\\
&\ge 1-\frac{\epsilon}{8}-\frac{\epsilon}{8} = 1-\frac{\epsilon}{4}.
\end{aligned}$$

As long as $\mathrm{qcd}^{t'''}(\bar{z}^{i'}_{<j'}x_i \mid \bar{z}^{<i'}) \le \log p_1(\epsilon^{-1}t^{\delta^{-1}})$ is satisfied, by Lemma 6.5,

$$\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{z}^{i'}_{<j'}x_i; \bar{z}^{<i'}, 1^{\langle\beta,t''',4\epsilon^{-1},p_1(\epsilon^{-1}t^{\delta^{-1}})\rangle}\right), \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i; \mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right) \le \frac{\epsilon}{4}.$$

This implies that

$$\Pr_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{z}^{i'}_{<j'}x_i; \bar{z}^{<i'}, 1^{\langle\beta,t''',4\epsilon^{-1},p_1(\epsilon^{-1}t^{\delta^{-1}})\rangle}\right), \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i; \mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right) \le \frac{\epsilon}{4}\right] \ge 1-\frac{\epsilon}{4},$$

and

$$\mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{z}^{i'}_{<j'}x_i; \bar{z}^{<i'}, 1^{\langle\beta,t''',4\epsilon^{-1},p_1(\epsilon^{-1}t^{\delta^{-1}})\rangle}\right), \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i; \mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\right] \le \frac{\epsilon}{4}+\frac{\epsilon}{4} = \frac{\epsilon}{2}.$$

Therefore, if both the events in Equations (10) and (11) occur,

$$\begin{aligned}
&\mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{z}^{i'}_{<j'}x_i; \bar{z}^{<i'}, 1^{\langle\beta,t''',4\epsilon^{-1},p_1(\epsilon^{-1}t^{\delta^{-1}})\rangle}\right), \mathcal{Y}_i|z_{<i}x_i\right)\right]\\
&\le \mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\Delta_{\mathrm{tv}}\left(\mathsf{UE}\left(\bar{z}^{i'}_{<j'}x_i; \bar{z}^{<i'}, 1^{\langle\beta,t''',4\epsilon^{-1},p_1(\epsilon^{-1}t^{\delta^{-1}})\rangle}\right), \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i; \mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\right]\\
&\qquad + \mathrm{E}_{x_i\sim\mathcal{X}_i|z_{<i}}\left[\Delta_{\mathrm{tv}}\left(\mathcal{Y}_i|z_{<i}x_i, \mathsf{Next}_\beta\left(\bar{z}^{i'}_{<j'}x_i; \mathrm{Q}^{t'',\bar{z}^{<i'}}\right)\right)\right]\\
&\le \frac{\epsilon}{2}+\frac{\epsilon}{16} < \epsilon.
\end{aligned}$$

By the union bound, we conclude that

$$\Pr_{i,z_{<i}}\left[\mathrm{E}_{x_i}\left[\Delta_{\mathrm{tv}}\left(L\left(x_{<i}; 1^{\langle n,m,s,t,\epsilon^{-1},\delta^{-1},t^{\delta^{-1}}\rangle}\right), Y_{i|<i}\right)\right] \le \epsilon\right] \ge 1-\left(\frac{3}{34}\delta+\frac{\delta}{4}+\frac{\delta}{3}\right) > 1-\delta.$$

□

References

[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.

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

[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.

[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.

[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.

[Hir22] 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.

[HKLW88] David Haussler, Michael J. Kearns, Nick Littlestone, and Manfred K. Warmuth. "Equivalence of Models for Polynomial Learnability". In: *Proceedings of the First Annual Workshop on Computational Learning Theory, COLT '88, Cambridge, MA, USA, August 3-5, 1988*. Ed. by David Haussler and Leonard Pitt. ACM/MIT, 1988, pp. 42–55. URL: http://dl.acm.org/citation.cfm?id=93040.

[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.

[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.

[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.

[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.

[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: *Electron. Colloquium Comput. Complex.* 27 (2020), p. 52.

[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.

[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.

[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.

[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.

A Estimating Statistical Distance

In this section, we prove the following technical lemma, building on the ideas from [NR06].

Lemma (A reminder of Lemma 7.5)**.** *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 A.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 A.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 A.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 A.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 A.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. □

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