CODING INFORMATION INTO ALL INFINITE SUBSETS OF A DENSE SET

MATTHEW HARRISON-TRAINOR, LU LIU, AND PATRICK LUTZ

Abstract. Suppose you have an uncomputable set X and you want to find a set A, all of whose infinite subsets compute X. There are several ways to do this, but all of them seem to produce a set A which is fairly sparse. We show that this is necessary in the following technical sense: if X is uncomputable and A is a set of positive lower density then A has an infinite subset which does not compute X. We also prove an analogous result for PA degree: if X is uncomputable and A is a set of positive lower density then A has an infinite subset which is not of PA degree. We will show that these theorems are sharp in certain senses and also prove a quantitative version formulated in terms of Kolmogorov complexity. Our results use a modified version of Mathias forcing and build on work by Seetapun, Liu, and others on the reverse math of Ramsey's theorem for pairs.

1. Introduction

Suppose you have an uncomputable set X and would like to find an infinite set $A \subseteq \mathbb{N}$ such that all infinite subsets of A compute X. Here's one way to do this, due to Dekker and Myhill [3]: identify $\mathbb{N}$ with $2^{<\omega}$ and let A be the set of all finite initial segments of X.

This is not the only way to encode an uncomputable set X into all infinite subsets of A. For example, if X is hyperarithmetic then it can be computed from any sufficiently fast growing function ([9], Theorem 6.8). If we make sure A has sufficiently large gaps between its elements then for any infinite subset B of A, the function which enumerates the elements of B grows fast enough to compute X and hence B itself computes X.

Note that both of these methods produce fairly sparse subsets of $\mathbb{N}$. If we use Dekker and Myhill's method then the set A will have just n elements less than 2^n. If we use the second method (in the case where X is hyperarithmetic) then the set A will be even sparser—the gaps between successive elements of A grow faster than any computable function.

It seems reasonable to informally conjecture that this is a necessary feature of such coding methods; in other words, to conjecture that if X is uncomputable and every infinite subset of A computes X then A must be sparse. We can turn this informal conjecture into a formal one by picking a precise definition of "sparse."

The main theorem of this paper states that the conjecture holds if we define "sparse" to mean "lower density zero." Recall that the *lower density* of a set of natural numbers $A \subseteq \mathbb{N}$ is

$$\underline{\rho}(A) = \liminf_{n\to\infty} \frac{|A \cap [n]|}{n+1}$$

This work was supported by the National Science Foundation under Grant DMS-2153823 and Grant DMS-2203072.

where $[n]$ denotes the set $\{0, 1, \ldots, n\}$. The main theorem of this paper is as follows.

Theorem 1.1. *For any uncomputable set X and any set $A \subseteq \mathbb{N}$ of positive lower density, there is some infinite subset of A which does not compute X.*

Our proof of this theorem relies on a theorem implicit in the work of Seetapun [15] and first proved explicitly by Dzhafarov and Jockusch [6].[1] For a proof, see [8], Theorem 6.63.

Theorem 1.2 (Seetapun's theorem)**.** *For any uncomputable set X and any set $A \subseteq \mathbb{N}$, either A or $\overline{A}$ must contain some infinite subset which does not compute X.*

Our proof is partially inspired by proofs of results related to Seetapun's theorem by Cholak, Jockusch and Slaman [2], Dzhafarov and Jockusch [6] and Monin and Patey [12]. More specifically, those results are proved using variations on Mathias forcing and use the low basis theorem (or the cone avoiding basis theorem) to show that certain sets of conditions are dense. Our proof also uses a variation on Mathias forcing and the cone avoiding basis theorem, but, in addition, uses Seetapun's theorem in a manner similar to the cone avoiding basis theorem. We will explain our strategy more carefully in Section 3.

It is natural to ask whether Theorem 1.1 holds for stronger notions of sparsity. In Section 4, we will show that our theorem is sharp in the sense that it fails to hold for several such notions.

We will also prove two other results which show that it is difficult to encode information into all infinite subsets of a dense set. These two results are stated in terms of having PA degree and in terms of Kolmogorov complexity, respectively.

Avoiding PA degree. Theorem 1.1 can be rephrased in terms of the property of *cone avoidance*. Say that a set $\mathcal{C} \subseteq 2^\omega$ *avoids cones* if for every nontrivial cone of Turing degrees, there is some element of $\mathcal{C}$ which is not in that cone. Then Theorem 1.1 can be restated as: for every set $A \subseteq \mathbb{N}$ of positive lower density, the set of infinite subsets of A avoids cones.

Cone avoidance can be seen as a kind of computability-theoretic weakness. Another standard weakness notion is *PA avoidance*. Say that a set $\mathcal{C} \subseteq 2^\omega$ *avoids PA degree* if there is some element of $\mathcal{C}$ which is not of PA degree. Though cone avoidance and PA avoidance are not equivalent, they do occur together relatively frequently.

One example of this comes from research on RT^2_2, a statement of Ramsey theory which has been thoroughly studied in the field of Reverse Mathematics. Seetapun showed that over RCA_0, RT^2_2 does not imply ACA_0. A core part of the proof is Seetapun's theorem above, which can be read as stating that for any set $A \subseteq \mathbb{N}$, the set of infinite subsets of A and $\overline{A}$ avoids cones. Later, Liu showed that RT^2_2 also does not imply WKL_0 and the heart of his proof was a theorem—analogous to Seetapun's theorem—stating that for every set $A \subseteq \mathbb{N}$, the set of infinite subsets of A and $\overline{A}$ avoids PA degree.

Theorem 1.3 (Liu's Theorem, [10])**.** *For any set $A \subseteq \mathbb{N}$, either A or $\overline{A}$ must contain an infinite set that is not of PA degree.*

[1] This theorem is sometimes known by the name "strong cone avoidance for RT^1_2," which originates from its connection to the reverse math of Ramsey's theorem. We decided not to use that name here since we are not concerned with reverse math in this paper.

It seems natural to ask whether our main theorem can be modified to yield PA avoidance rather than cone avoidance, in the same way that Seetapun's theorem can be modified to yield Liu's theorem. We show that this is indeed the case.

Theorem 1.4. *For any set $A \subseteq \mathbb{N}$ of positive lower density, there is an infinite subset of A which is not of PA degree.*

Just as our proof of Theorem 1.1 uses Seetapun's theorem, our proof of the theorem above uses Liu's theorem (along with several techniques first developed to prove Liu's theorem). We will also show that a certain natural-sounding common generalization of both Theorem 1.1 and Theorem 1.4 is false.

Kolmogorov complexity. We will also investigate a quantitative version of Theorem 1.1. Stated loosely, that theorem says that it is impossible to encode an infinite amount of information into all infinite subsets of a set of positive lower density. On the other hand, it is obvious that some finite information may be so encoded. For example, it is easy to encode one bit of information into all infinite subsets of a set of lower density $1/2$ (by using the parity of the elements of the set) and, more generally, n bits of information into all infinite subsets of a set of lower density $1/2^n$. But just how much information can be encoded?

We can formulate a precise version of this question using Kolmogorov complexity. To do so, it is convenient to introduce the following definition.

Definition 1.5. For a string $\sigma \in 2^{<\omega}$ and a family of sets $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$, define

$$C(\sigma \mid \mathcal{F}) = \max_{X \in \mathcal{F}} C^X(\sigma).$$

Informally, $C(\sigma \mid \mathcal{F})$ should be thought of as the Kolmogorov complexity of σ relative to an arbitrary element of $\mathcal{F}$. Recall that for any set $A \subseteq \mathbb{N}$, $[A]^\omega$ denotes the family of all infinite subsets of A. The quantity

$$C(\sigma) - C(\sigma \mid [A]^\omega)$$

can be thought of as the number of bits of information about σ that are encoded into all infinite subsets of A. Thus a somewhat more formal version of our question above is: if σ is any string and $A \subseteq \mathbb{N}$ is a set of lower density at least $\delta > 0$, then how large can $C(\sigma) - C(\sigma \mid [A]^\omega)$ be?

Based on our example above, it is perhaps natural to guess that this difference should not be much larger than $\log(1/\delta)$—in other words, that it should not be possible to encode more than about $\log(1/\delta)$ bits of information about σ into all infinite subsets of a set of lower density δ. Surprisingly, this is not the case.

Proposition 1.6. *For any string σ and $\delta \in (0, 1]$, there is some set $A \subseteq \mathbb{N}$ of lower density at least δ such that*

$$C(\sigma \mid [A]^\omega) \leq \max(0, C^{0'}(\sigma) - \log(1/\delta)) + O(\log\log(1/\delta)).$$

In other words, in addition to lowering the complexity of σ by $\log(1/\delta)$, we can also lower it to the $0'$ complexity of σ. In fact, this upper bound on $C(\sigma \mid [A]^\omega)$ is also optimal (up to a small error term).

Theorem 1.7. *For any string σ and set $A \subseteq \mathbb{N}$ of lower density at least $\delta \in (0, 1]$,*

$$C(\sigma \mid [A]^\omega) \geq C^{0'}(\sigma) - \log(1/\delta) - O(\log\log(1/\delta))$$

where the constant hidden by the $O(\cdot)$ notation does not depend on σ or A.

We will prove these two results in Section 6 and comment on some further questions around how much information can be encoded into all infinite subsets of a dense set.

Acknowledgements. We thank Andrew Marks for posing the question that motivated this paper and for several helpful conversations and Damir Dzhafarov for a useful conversation on the topic of Weihrauch reducibility.

2. Preliminaries

2.1. Notation.

We will use the following notation for finite sets of natural numbers. For natural numbers $n < m$,

$$\begin{aligned} [n] &= \{0, 1, \ldots, n\} \\ [n, m) &= \{n, n+1, \ldots, m-1\} \\ (n, m) &= \{n+1, n+2, \ldots, m-1\}. \end{aligned}$$

We will also use the following notation related to infinite sets of natural numbers. For a set $A \subseteq \mathbb{N}$,

$$\begin{aligned} A(n) = &\text{ the } n^{\text{th}} \text{ bit of } A\text{, i.e. 0 if } n \notin A \text{ and 1 if } n \in A \\ \overline{A} = &\text{ the complement of } A \\ [A]^\omega = &\text{ the set of infinite subsets of } A. \end{aligned}$$

By a *Turing functional* we mean a program Φ with oracle access which has inputs in $\mathbb{N}$ and outputs in $\{0, 1\}$. We will use the following notation related to Turing functionals. For a Turing functional Φ, oracle $A \subseteq \mathbb{N}$ and number $n \in \mathbb{N}$,

$$\begin{aligned} \Phi(A, n) = &\text{ the output of } \Phi \text{ with oracle } A \text{ on input } n \\ \Phi(A) = &\text{ the partial function } \mathbb{N} \to \{0, 1\} \text{ given by } n \mapsto \Phi(A, n) \\ \Phi(A, n) \neq b \;\; &\text{means that either } \Phi(A, n) \text{ diverges or } \Phi(A, n)\!\downarrow\, \neq b. \end{aligned}$$

We will sometimes want to consider a finite set $s \subseteq \mathbb{N}$ as an initial segment of an oracle. We will use $\Phi(s, n)$ to mean the output of Φ on input n when run for at most $\max(s)$ steps, using s as an oracle and automatically diverging if there is any query to the oracle about a number larger than $\max(s)$.

Similarly, we will sometimes want to consider the Kolmogorov complexity of a string σ relative to a finite set $s \subseteq \mathbb{N}$, which we will denote by $C^s(\sigma)$. More precisely, $C^s(\sigma)$ will denote the length of the shortest program which outputs σ when using an oracle for s and which runs for at most $\max(s)$ steps, never makes an oracle query about a number larger than $\max(s)$ and has length at most $\max(s)$. Note that it is possible that no such program exists due to the limitations on length and running time. In this case we define $C^s(\sigma) = \infty$. Also note that $C^s(\sigma)$ is computable.

2.2. Density of sets of natural numbers.

If $A \subseteq \mathbb{N}$ is a set of natural numbers then its *lower density*, denoted $\underline{\rho}$, and *upper density*, denoted $\overline{\rho}$, are defined by

$$\begin{aligned} \underline{\rho}(A) &= \liminf_{n \to \infty} \frac{|A \cap [n]|}{n+1} \\ \overline{\rho}(A) &= \limsup_{n \to \infty} \frac{|A \cap [n]|}{n+1}. \end{aligned}$$

If $\underline{\rho}(A) > 0$ then we say A has *positive lower density.* Similarly, if $\overline{\rho}(A) > 0$ then A has *positive upper density.*

In order to work with dense subsets of $\mathbb{N}$, it will be helpful to introduce some auxiliary terminology. To motivate this terminology, note that if the upper density of a set A is strictly greater than δ then there are infinitely many n such that

$$|A \cap [n]| \geq \delta \cdot (n+1).$$

It is useful to be able to speak about the collection of all such n. To that end, we introduce the following terminology. For $A, D \subseteq \mathbb{N}$ and $\delta > 0$:

- A is *δ-dense at n* if $|A \cap [n]| \geq \delta \cdot (n+1)$.
- A is *δ-dense* if it is δ-dense at every $n \in \mathbb{N}$ and *dense* if it is δ-dense for some $\delta > 0$.
- A is *δ-dense along D* if it is δ-dense at every $n \in D$ and *dense along D* if it is δ-dense along D for some $\delta > 0$.

The reason these notions are useful is that they act as lower complexity versions of the properties of having positive upper or lower density.

For example, a set A has positive lower density if and only if $\{0\} \cup A$ is δ-dense for some $\delta > 0$ (though note that δ may need to be much lower than $\underline{\rho}(A)$), but the property of having positive lower density is Σ^0_2, while the property of being δ-dense is Π^0_1.

Similarly, a set A has positive upper density if and only if it is δ-dense along D for some $\delta > 0$ and infinite set $D \subseteq \mathbb{N}$ (and this time, δ can be arbitrarily close to $\overline{\rho}(A)$), but the property of having positive upper density is Σ^0_3 while the property of being δ-dense along D is again Π^0_1.

2.3. **Mathias forcing.**

Mathias forcing is a useful tool for constructing an infinite subset of a set while ensuring that the subset being constructed satisfies various properties. We will briefly review the basics of Mathias forcing; for a more complete introduction, see [8], Section 6.5.

A *condition for Mathias forcing* is a pair (s, A) consisting of a finite set $s \subseteq \mathbb{N}$, called the *stem*, and an infinite set $A \subseteq \mathbb{N}$, called the *reservoir*, such that $\max(s) < \min(A)$. Often, the reservoir A is required to come from some restricted class of sets, such as a Turing ideal or co-cone.

A Mathias condition, (s, A), is extended by a condition, (s', A'), written $(s, A) \geq (s', A')$, if all of the following hold.

- $s \subseteq s'$
- $A \supseteq A'$
- and for all $n \in s' \setminus s$, $n \in A$.

In other words, (s', A') is formed from (s, A) by choosing finitely many elements of the reservoir to add to the stem and by removing some elements (possibly infinitely many) from the reservoir.

A Mathias condition, (s, A), should be thought of as partially specifying a subset G of $\mathbb{N}$ as follows: the stem s consists of numbers that have already been put into G and the reservoir A consists of numbers that may be put into G at some later stage. Thus any number which is not in $s \cup A$ is definitely not in G.

This can be made precise as follows. Any filter for Mathias forcing can be used to define a subset of $\mathbb{N}$ by taking the union of all the stems in the filter. We will denote this subset by G and, in a slight abuse of terminology, will often conflate it with the filter itself (i.e. we will refer to G as the generic).

It is straightforward to check that if the filter is sufficiently generic then G is infinite. Also, say that a set B is *compatible* with a condition (s, A) if $s \subseteq B \subseteq s \cup A$. If (s, A) is any element of the filter defining G then it is straightforward to check that G is compatible with (s, A).

Thus one way (though not the only way) to show that every sufficiently generic G satisfies some property P is to show that the following set of conditions

$$\{(s, A) \mid \text{all infinite sets } C \text{ compatible with } (s, A) \text{ satisfy } P\}$$

is dense in the Mathias forcing partial order.

2.4. **Seetapun's theorem.** In the introduction, we mentioned that our proof of Theorem 1.1 uses Seetapun's theorem.

Theorem 1.2 (Seetapun's theorem)**.** *For any uncomputable set X and any set $A \subseteq \mathbb{N}$, either A or $\overline{A}$ must contain some infinite subset which does not compute X.*

We will also need to use a couple corollaries of this theorem.

Corollary 2.1. *Suppose $A \subseteq \mathbb{N}$ is infinite and does not compute X. Then for any $B \subseteq A$ there is some infinite subset of either B or $A \setminus B$ which does not compute X.*

Proof. Let $\pi \colon A \to \mathbb{N}$ be a bijection which is computable from A. The idea is to consider Seetapun's theorem applied to $\pi(B)$.

Note that since π is a bijection, $\mathbb{N} \setminus \pi(B) = \pi(A \setminus B)$. Thus by Seetapun's theorem relativized to A, either $\pi(B)$ or $\pi(A \setminus B)$ has an infinite subset C such that $C \oplus A$ does not compute X. Since π is a bijection, $\pi^{-1}(C)$ is an infinite subset of either B or $A \setminus B$. And since A computes π, $C \oplus A$ computes $\pi^{-1}(C)$ and hence $\pi^{-1}(C)$ does not compute X. □

Corollary 2.2. *Suppose $A \subseteq \mathbb{N}$ is infinite and does not compute X. Then for any finite partition $B_1, \ldots, B_n$ of A, at least one of the B_i's has an infinite subset which does not compute X.*

Proof. The idea of the proof is to use induction on n and to reduce the inductive case to the previous corollary.

The base case, $n = 1$ is trivial since in this case $B_1 = A$ and thus B_1 itself does not compute X.

Now assume for induction that the statement holds for all A, X and partitions of length at n. Fix A and X as in the statement of the corollary and suppose $B_1, \ldots, B_{n+1}$ is a partition of A. By applying Corollary 2.1 to the sets $\bigcup_{i \leq n} B_i$ and B_{n+1}, we obtain an infinite set C which is a subset of either $\bigcup_{i \leq n} B_i$ or B_{n+1} such that C does not compute X.

If $C \subseteq B_{n+1}$ then we are done. If $C \subseteq \bigcup_{i \leq n} B_i$ then we can apply the inductive assumption to the partition $B_1 \cap C, \ldots, B_n \cap C$ of C to obtain an infinite set D which does not compute X and which is a subset of $B_i \cap C$ (and thus of B_i as well) for some $i \leq n$. □

3. Proof of the main theorem

For the rest of this section, suppose that X is uncomputable and $A \subseteq \mathbb{N}$ has positive lower density. We want to construct an infinite subset of A which does not compute X. We will construct this set using Mathias forcing. The basic strategy is as follows: let G be a Mathias generic which is compatible with the condition $(\varnothing, A)$ (i.e. the generic filter used to define G contains the condition $(\varnothing, A)$). Then we have

- Since G is compatible with $(\varnothing, A)$, $G \subseteq A$.
- Since G is sufficiently generic, G is infinite.

If we could also show that any sufficiently generic G does not compute X then we would be done.

However, it is not hard to see that a generic for plain Mathias forcing—with no restrictions on the possible reservoirs—does not have this property. For example, suppose B is an infinite set such that all infinite subsets of B compute X. Then any Mathias generic G which is compatible with $(\varnothing, B)$ will compute X. To solve this problem, we will impose a restriction on the reservoirs of the Mathias conditions which guarantees that every sufficiently generic G does not compute X.

3.1. **A natural idea that doesn't work.** Perhaps the most obvious restriction to put on the reservoirs is to require them to have positive lower density. Here's why this seems natural. The problem with plain Mathias forcing is that there are conditions (s, B) such that *every set* compatible with (s, B) computes X. However, if we believe the statement we are trying to prove—that every set of positive lower density contains an infinite subset which does not compute X—then this same problem cannot occur when B is required to have positive lower density.

However, this approach does not work. The problem, briefly stated is that even though every set of positive lower density contains an infinite subset which does not compute X (as we will eventually show), there is a set of positive lower density, all of whose subsets of positive lower density uniformly compute X. Thus if we only use reservoirs that have positive lower density, then given a condition (s, B) and a Turing functional Φ, there is no obvious way to find a condition $(s', B') \leq (s, B)$ forcing that the generic G does not compute X via Φ (since it could be that for any such (s', B'), $s' \cup B'$ itself computes X via Φ).

In fact, it is even possible to show that if we only use reservoirs that have positive lower density then it is possible that G is forced to compute X. This can be shown using the following proposition, which we will prove in Section 4.3.

Proposition 4.5. *For any set X, there is some set $A \subseteq \mathbb{N}$ of positive lower density such that all subsets of A of positive lower density compute X uniformly.*

Proposition 3.1. *Let A and X be as in the above proposition. If G is generic for Mathias forcing with reservoirs of positive lower density and G is compatible with $(\varnothing, A)$ then G computes X.*

Proof. Let Φ be a Turing functional witnessing the fact that all subsets of A of positive lower density compute X uniformly (in other words, such that for all $B \subseteq A$ of positive lower density, $\Phi(B) = X$). For each n, we will show that the set of conditions

$$\mathcal{D}_n = \{(s, B) \mid \Phi(s, n)\downarrow = X(n)\}$$

is dense below $(\varnothing, A)$. This shows that for each n, $\Phi(G, n)\downarrow = X(n)$ and hence that $\Phi(G) = X$.

So fix n for which we will show that $\mathcal{D}_n$ is dense below $(\varnothing, A)$. Let (s, B) be an arbitrary condition extending $(\varnothing, A)$. We need to show that there is some condition (s', B') extending (s, B) such that $\Phi(s', n)\downarrow = X(n)$.

Since $s \cup B$ is a subset of A of positive lower density, we must have $\Phi(s \cup B, n)\!\downarrow = X(n)$. Let s' be an initial segment of $s \cup B$ which is long enough to witness this and set $B' = B \setminus s'$. Then $(s', B') \leq (s, B)$ and $\Phi(s', n)\!\downarrow = X(n)$ as desired. □

3.2. **Density Mathias forcing.** We have seen that if we want to use Mathias forcing to construct an infinite subset of A which does not compute X then neither allowing all infinite sets as reservoirs nor requiring the reservoirs to have positive lower density works. The problem with the former is that there are too many reservoirs available, in particular there are reservoirs whose infinite subsets all compute X. The problem with the latter is that there are too few reservoirs available, in particular, for a fixed reservoir B and Turing functional Φ it may not be possible to find a reservoir $B' \subseteq B$ witnessing that not all infinite subsets of B compute X via Φ. Thus we want a requirement that lies in-between these two extremes.

We will now describe such a requirement. In essence, we will allow a set B to be a reservoir if it has positive upper density and the fact that it has positive upper density is witnessed by a set that does not compute X. We will refer to Mathias forcing with reservoirs satisfying this condition as *density Mathias forcing.*

Definition 3.2. A condition for *density Mathias forcing* is a Mathias condition (s, B) for which there is some set D such that

(1) B is dense along D
(2) and D does not compute X.

Note that $(\varnothing, A)$ itself is a density Mathias condition: A is dense along $\mathbb{N} \setminus [\min(A)]$ and since X is uncomputable, $\mathbb{N} \setminus [\min(A)]$ does not compute X.

3.3. **The proof.** We will now show that if G is sufficiently generic for density Mathias forcing then G does not compute X. We will begin with a technical lemma which we will use in the proof. Then we will prove the main lemma, which shows that for a single program Φ there is a dense set of conditions which guarantee that $\Phi(G) \neq X$.

Lemma 3.3. *If (s, B) is a density Mathias condition and $B_1, \ldots, B_k$ is a finite partition of B then for some $i \leq k$, (s, B_i) is a density Mathias condition.*

Proof. Since (s, B) is a density Mathias condition there is some $\delta > 0$ and some infinite set D such that B is δ-dense along D and D does not compute X. Now define subsets $D_1, \ldots, D_k$ of D by

$$D_i = \{n \in D \mid i \text{ is least such that } B_i \text{ is } \delta/k\text{-dense at } n\}.$$

We claim that $D_1, \ldots, D_k$ partition D. To see why, let $n \in D$. Thus $B \cap [n] \geq \delta \cdot (n + 1)$. Since $B_1, \ldots, B_k$ partition B, at least one of the sets

$$B_1 \cap [n], B_2 \cap [n], \ldots, B_k \cap [n]$$

must have size at least $\delta \cdot (n + 1)/k$. Thus there is some i such that B_i is δ/k-dense at n.

Since D does not compute X and $D_1, \ldots, D_k$ partition D, we can apply Corollary 2.2 to Seetapun's theorem to get an infinite set E which is contained in some D_i and which does not compute X. It is straightforward to check that B_i is δ/k-dense along E and thus (s, B_i) is a density Mathias condition. □

Lemma 3.4. *For any Turing functional Φ and density Mathias condition (s, B), there is some density Mathias condition $(s', B') \leq (s, B)$ such that for any set C compatible with (s', B'), $\Phi(C) \neq X$.*

Proof. Since (s, B) is a density Mathias condition, we can fix some $\delta > 0$ and infinite set D such that B is δ-dense along D and D does not compute X. There are two cases to consider.

(1) It is possible to make Φ wrong on some input: there is some $n \in \mathbb{N}$ and finite set $t \subseteq B$ such that $\Phi(s \cup t, n)\!\downarrow\, \neq X(n)$.
(2) It is impossible to make Φ wrong on any input: for every $n \in \mathbb{N}$ and finite set $t \subseteq B$, $\Phi(s \cup t, n)$ either diverges or is equal to $X(n)$.

The first case is easy: we can simply take $s' = s \cup t$ and $B' = B \setminus [\max(t)]$. Thus we may assume we are in the second case.

Since we are in the second case, it follows that for every C compatible with (s, B), $\Phi(C)$ never disagrees with X—though it may diverge on some inputs. Our goal is to find (s', B') extending (s, B) such that for every C compatible with (s', B'), $\Phi(C)$ does diverge on at least one input.

Intuition. Here's the basic idea of the proof. By using ideas from proofs of Seetapun's theorem, it is not too hard to find a set B_0 and number n_0 such that for all C compatible with (s, B_0), $\Phi(C, n_0) \neq X(n_0)$. It is also not too hard to ensure B_0 is δ-dense along D. We would like to take $B' = B \cap B_0$. The reason is that any C compatible with $(s, B \cap B_0)$ is compatible with both (s, B) and (s, B_0) and thus for such a C, $\Phi(C, n_0)$ is neither different from $X(n_0)$ (because C is compatible with (s, B)) nor equal to $X(n_0)$ (because C is compatible with (s, B_0)) and thus $\Phi(C, n_0)$ must diverge.

The problem with this idea is that $B \cap B_0$ might not be very dense—in fact, it might even be empty. So instead, we will iterate: we will find another set B_1 and number n_1 which have the same properties as B_0 and n_0 but such that B_1 is disjoint from B_0. If neither $B \cap B_0$ nor $B \cap B_1$ work then we will keep going and find B_2 disjoint from both, and so on.

The fact that sets are disjoint and all fairly dense will ensure that this process cannot go on forever: it is not possible to have more than $1/\delta$ disjoint sets which are all δ-dense. Thus we will eventually find some B_i which works. In practice, carrying out this idea involves a lot of additional technical details.

General strategy. We will find disjoint sets $B_0, \ldots, B_k$ and numbers $n_0, \ldots, n_k$ such that

(1) for each $i \leq k$ and all C compatible with (s, B_i), $\Phi(C, n_i) \neq X(n_i)$ (i.e. it may diverge or it may converge to some output that is not equal to $X(n_i)$)
(2) and $(s, B \cap (B_0 \cup \ldots \cup B_k))$ is a density Mathias condition.

Before explaining how to find the B_i's, let's explain why this is enough to finish the proof. Since $(s, B \cap (B_0 \cup \ldots \cup B_k))$ is a density Mathias condition whose reservoir is partitioned by $B \cap B_0, \ldots, B \cap B_k$, Lemma 3.3 implies that there is some i such that $(s, B \cap B_i)$ is a density Mathias condition.

We now claim that we can take $s' = s$ and $B' = B \cap B_i$. To see why, note that if C is compatible with $(s, B \cap B_i)$ then it is compatible with both (s, B) and (s, B_i). Thus $\Phi(C, n_i)$ can neither disagree with $X(n_i)$ (since C is compatible with (s, B)) nor agree with $X(n_i)$ (since C is compatible with (s, B_i)) and therefore $\Phi(C, n_i)$ must diverge.

Strategy to construct $\boldsymbol{B_0, \ldots, B_k}$. We will build the sequence of B_i's inductively. To make the induction work, we will require the sets B_i to satisfy some additional properties. In particular, we will construct a sequence of sets $B_0, B_1, B_2, \ldots$ along with sets $D \supseteq D_0 \supseteq D_1 \supseteq D_2 \supseteq \ldots$ and numbers $n_0, n_1, n_2, \ldots$ such that for each i,

(1) D_i is infinite.
(2) B_i is $\delta/2$-dense along D_i.
(3) $B_0 \oplus B_1 \oplus \ldots \oplus B_i \oplus D_i$ does not compute X.
(4) B_i is disjoint from $B_0 \cup \ldots \cup B_{i-1}$
(5) For each C compatible with (s, B_i), $\Phi(C, n_i) \neq X(n_i)$.

We will show by induction that if we have built sequences $B_0, \ldots, B_k$, $D_0, \ldots, D_k$ and $n_0, \ldots, n_k$ satisfying these requirements then either $(s, B \cap (B_0 \cup \ldots \cup B_k))$ is a density Mathias condition (and thus we are finished building the sequence) or we can extend the sequence—i.e. find B_{k+1}, D_{k+1} and n_{k+1} satisfying the requirements. To finish, we will show that any sequence satisfying the requirements cannot be infinite.

Extending the sequence: picking $\boldsymbol{D_{k+1}}$. Suppose that we have built sequences $B_0, \ldots, B_k$, $D_0, \ldots, D_k$ and $n_0, \ldots, n_k$ satisfying the five requirements listed above. If we knew that $B \cap (B_0 \cup \ldots \cup B_k)$ was dense along D_k then we could stop: in that case $(s, B \cap (B_0 \cup \ldots \cup B_k))$ would be a density Mathias condition. More generally, if we knew that $B \cap (B_0 \cup \ldots \cup B_k)$ was dense along any infinite subset of D_k which does not compute X then we could stop. So let's assume that's not the case and show we can extend the sequence $B_0, \ldots, B_k$.

Let E be the subset of D_k defined by

$$E = \{n \in D_k \mid B \cap (B_0 \cup \ldots \cup B_k) \text{ is } \delta/2\text{-dense at } n\}.$$

By (the relativized form of) Corollary 2.2 to Seetapun's theorem, there is an infinite subset D_{k+1} of either E or $D_k \setminus E$ such that $B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}$ does not compute X. However, D_{k+1} cannot be a subset of E since then $B \cap (B_0 \cup \ldots \cup B_k)$ would be dense along D_{k+1}, which contradicts our assumption above.

Thus $D_{k+1} \subseteq D_k \setminus E$. This implies $B \setminus (B_0 \cup \ldots \cup B_k)$ is $\delta/2$-dense along D_{k+1}. To see why, consider any $n \in D_{k+1}$. Since $D_{k+1} \subseteq D_k \subseteq D$ and B is δ-dense along D, B is δ-dense at n. Since $n \notin E$, $B \cap (B_0 \cup \ldots \cup B_k)$ is not $\delta/2$-dense at n. Therefore $B \setminus (B_0 \cup \ldots \cup B_k)$ must be $\delta/2$-dense at n.

Extending the sequence: picking $\boldsymbol{B_{k+1}}$. Observe that $B \setminus (B_0 \cup \ldots \cup B_k)$ has the following properties.

(1) As we just showed, it is $\delta/2$-dense along D_{k+1}.
(2) It is disjoint from $B_0 \cup \ldots \cup B_k$.
(3) Since it is a subset of B, for every finite subset t and every $n \in \mathbb{N}$, $\Phi(s \cup t, n)$ either diverges or is equal to $X(n)$.

These properties look almost like what we want from B_{k+1} but in the last item above we would like to have "not equal to $X(n)$" rather than "equal to $X(n)$." In other words, we want to find a set which is very similar to $B \setminus (B_0 \cup \ldots \cup B_k)$ but differs in one key respect. We will argue that if we cannot find such a set then X is computable from $B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}$.

For each $n \in \mathbb{N}$ and $b \in \{0,1\}$, define a set $\mathcal{C}_{n,b} \subseteq \mathcal{P}(N)$ by

$$\begin{aligned}\mathcal{C}_{n,b} = \{Y \subseteq \mathbb{N} \mid\ & Y \text{ is disjoint from } B_0 \cup \ldots \cup B_k \\ & \text{and } Y \text{ is } \delta/2\text{-dense along } D_{k+1} \\ & \text{and for all finite } t \subseteq Y, \text{ either } \Phi(s \cup t, n)\uparrow \text{ or } \Phi(s \cup t, n)\downarrow = b\}.\end{aligned}$$

Note that each $\mathcal{C}_{n,b}$ is a Π^0_1 class relative to $B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}$. Thus for each (n,b), there is some $B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}$-computable binary tree $T_{n,b}$ whose paths are exactly the elements of $\mathcal{C}_{n,b}$. Furthermore, it is easy to see that $T_{n,b}$ can be computed uniformly in (n,b).

We would now like to show that for some n and $b \neq X(n)$, $\mathcal{C}_{n,b}$ is nonempty. Suppose not. Then for each $n \in \mathbb{N}$ and $b \in \{0,1\}$, we have the following

(1) If $X(n) = b$ then $\mathcal{C}_{n,b}$ is nonempty (as witnessed by $B \setminus (B_0 \cup \ldots \cup B_k)$) and thus $T_{n,b}$ is infinite.
(2) If $X(n) \neq b$ then $\mathcal{C}_{n,b}$ is empty and so, by König's lemma, $T_{n,b}$ is finite.

Therefore to compute $X(n)$ using $B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}$, we simply need to check which of $T_{n,0}$ and $T_{n,1}$ is finite.

Thus there is some n and $b \neq X(n)$ such that $\mathcal{C}_{n,b}$ is nonempty. By the cone avoiding basis theorem, relativized to $B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}$, we can find some $B_{k+1} \in \mathcal{C}_{n,b}$ such that

$$(B_0 \oplus \ldots \oplus B_k \oplus D_{k+1}) \oplus B_{k+1} \not\geq_T X.$$

It is straightforward to check that this B_{k+1} satisfies all the necessary requirements.

What about the base case? Since we are forming the sequences $B_0, B_1, \ldots$ and $D_0, D_1, \ldots$ inductively, it would seem that we need to explain the base case: how to pick B_0, D_0 and n_0. However, the argument we gave in the inductive case to pick B_{k+1} and D_{k+1} works just as well for the base case, with the exception that instead of having to pick D_0 carefully we can just take $D_0 = D$. Also in this case, we should interpret the union $B_0 \cup \ldots \cup B_k$ as the empty set.

The sequence cannot be infinite. We have shown that if we have formed sets $B_0, \ldots, B_k$ and $D_0, \ldots, D_k$ satisfying the five conditions listed above then either $(s, B \cap (B_0 \cup \ldots \cup B_k))$ is a density Mathias condition (in which case we are done) or we can find B_{k+1} and D_{k+1} extending the sequence. But how do we know the first option ever holds? In other words, how do we know we cannot just extend the sequence indefinitely? We will now show that no sequence satisfying the five requirements can be longer than $2/\delta$.

Suppose for contradiction that $B_0, \ldots, B_k$ and $D_0, \ldots, D_k$ do satisfy the five requirements and $k > 2/\delta$. Let n be any element of D_k. Note that for each $i \leq k$, $D_k \subseteq D_i$ and hence B_i is $\delta/2$-dense at n. In other words,

$$|B_i \cap [n]| \geq \frac{\delta}{2} \cdot (n+1).$$

However, since the B_i's are all disjoint, this gives us $k > 2/\delta$ disjoint subsets of $[n]$, all of size at least $\delta/2 \cdot (n+1)$, which is impossible. □

We can now prove Theorem 1.1.

Theorem 1.1. *For any uncomputable set X and any set $A \subseteq \mathbb{N}$ of positive lower density, there is some infinite subset of A which does not compute X.*

Proof. Let G be a generic for density Mathias forcing which is compatible with the condition $(\varnothing, A)$. We claim that G is an infinite subset of A which does not compute X. Since G is generic, it is infinite and since G is compatible with $(\varnothing, A)$, $G \subseteq A$. Now let Φ be any Turing functional. By Lemma 3.4, the following set of density Mathias conditions is dense

$$\{(s, B) \mid \text{for all } C \text{ compatible with } (s, B),\ \Phi(C) \neq X\}$$

and so $\Phi(G) \neq X$. Since this holds for all Φ, G does not compute X. □

3.4. **An open question.** Monin and Patey have proved the following variation on Seetapun's theorem, in which computability is replaced by hyperarithmetic reducibility [14].

Theorem 3.5 (Monin and Patey)**.** *For any non-hyperarithmetic set X and any set $A \subseteq \mathbb{N}$, there is an infinite subset B of either A or $\overline{A}$ such that X is not hyperarithmetic in B.*

It seems natural to ask whether the analogous variation on Theorem 1.1 is true. Answering this question seems to be beyond the techniques we used to prove our theorem.

Question 3.6. *Suppose X is not hyperarithmetic and $A \subseteq \mathbb{N}$ has positive lower density. Must A have an infinite subset B such that X is not hyperarithmetic in B?*

4. The main theorem is sharp

Our goal in proving Theorem 1.1 was to formalize the intuition that if every infinite subset of a set A can compute an uncomputable set X then A must be sparse. In our theorem, we interpreted sparse to mean lower density zero. However, this is not the only reasonable interpretation of what it means for a set of natural numbers to be sparse. In this section, we will consider a couple other notions of sparsity and give counterexamples showing that for each, our main theorem becomes false.

In other words, for various notions of sparsity, we will prove that there is an uncomputable X and a set $A \subseteq \mathbb{N}$ which is not sparse such that all infinite subsets of A compute X. In nearly all cases, we will be able to strengthen the counterexample in two ways.

First, instead of just finding a single X that can be encoded into a non-sparse set in this way, we will show that any set X can be so encoded—i.e. for every X there is some non-sparse $A \subseteq \mathbb{N}$ such that all infinite subsets of A compute X.

Second, instead of every subset of A simply computing X, we can ensure that they all compute X *uniformly*—i.e. there is a single Turing functional Φ such that for all infinite subsets B of A, $\Phi(B) = X$.

4.1. **Sets of positive upper density.** Our first counterexample concerns replacing positive lower density with positive upper density. In this case, we can even find a counterexample where A has upper density one.

Proposition 4.1. *For any set X, there is some set $A \subseteq \mathbb{N}$ such that $\overline{\rho}(A) = 1$ and all infinite subsets of A compute X uniformly.*

Proof. Let $\widetilde{A}$ be any infinite set, all of whose infinite subsets compute X uniformly (for example, $\widetilde{A}$ could be the set produced by the Dekker-Myhill method explained in the introduction). The idea is to build A by copying $\widetilde{A}$, but add in a lot of redundancy so that A is occasionally very dense.

To that end, pick a computable sequence of numbers $n_0 < n_1 < n_2 < \ldots$ which grows fast enough that $n_{i+1} > i \cdot n_i$ and define

$$A = \bigcup_{i \in \widetilde{A}} [n_i, n_{i+1}).$$

In other words, for each $i \in \widetilde{A}$, A contains the entire interval $[n_i, n_{i+1})$.

First observe that A has upper density 1. Indeed, for each $i \in \widetilde{A}$, A is $\frac{i-1}{i}$-dense at n_{i+1}.

Second, observe that all infinite subsets of A compute X uniformly: given an infinite subset $B \subseteq A$, we can uniformly compute the following infinite subset $\widetilde{B}$ of $\widetilde{A}$,

$$\widetilde{B} = \{i \mid \exists n \in [n_i, n_{i+1}) \cap A\},$$

and then use $\widetilde{B}$ to compute X. □

In spite of this counterexample, our main theorem can be strengthened to hold for certain sets of positive upper density.

Proposition 4.2. *For any uncomputable X and set $A \subseteq \mathbb{N}$ such that A is dense along some infinite set D which does not compute X, there is some infinite subset of A which does not compute X.*

Proof. Note that $(\varnothing, A)$ is a density Mathias condition, as witnessed by D. Thus there is a generic G for density Mathias forcing which is compatible with $(\varnothing, A)$. Exactly as in the proof of Theorem 1.1, G is an infinite subset of A which does not compute X. □

4.2. **Sets whose density goes to zero slowly.** Our next counterexample concerns sets of lower density zero where the density approaches 0 very slowly. Define the *density function* of a set A to be the function

$$d_A(n) = \frac{|A \cap [n]|}{n+1}.$$

Note that if a set A has positive lower density then $d_A(n)$ is bounded away from 0. By contrast, most of the methods we have seen so far of coding a set X into all infinite subsets of a set produce sets whose density functions converge to 0 very rapidly. For example, the Dekker-Myhill method we mentioned in the introduction produces a set A such that $d_A(n)$ is approximately $\frac{\log(n)}{n}$. In the previous subsection, we saw a method for which this is not quite true—in particular, we saw a method which produces a set A of upper density one, which implies that $d_A(n)$ is close to 1 infinitely often. However, even for this method, $d_A(n)$ is also infinitely often smaller than $\frac{\log(n)}{n}$.

Based on these examples, one might guess that if all infinite subsets of a set A compute an uncomputable set X then $d_A(n)$ cannot be lower bounded by any monotone function that goes to zero much slower than $\frac{\log(n)}{n}$—in other words that there must be infinitely many places where the density of A is exponentially small.

However, this is false: we will show that it is possible to encode any set X into all infinite subsets of a set A whose density function goes to zero arbitrarily slowly.

Definition 4.3. For any function $f\colon \mathbb{N} \to [0,1]$, a set $A \subseteq \mathbb{N}$ is *f-dense* if for all n,

$$d_A(n) \geq f(n).$$

Proposition 4.4. *Suppose $f\colon \mathbb{N} \to [0,1]$ is a function such that*

$$\lim_{n\to\infty} f(n) = 0.$$

Then for any set X there is some A such that A is f-dense and all infinite subsets of A compute X uniformly.

Proof. The idea is to modify the Dekker-Myhill coding method explained in the introduction, but slow it down so that it produces an f-dense set. In Dekker and Myhill's scheme, each element of A encodes one more bit of X than the previous element. In our modified version, we will repeatedly encode the same number of bits of X until f becomes small enough to allow us to encode more. Roughly speaking, we will only start using elements of A to encode the first n bits of X once f drops below $1/2^n$.

One comment before we go into the details of the construction: it might seem necessary to require f to be computable but there is a trick that allows us to avoid requiring that. Essentially, we can use elements of A to encode not just some bits of X but also to encode how many bits are encoded.

To define A formally, first pick a sequence $0 < n_1 < n_2 < \ldots$ which grows fast enough that for all i:

(1) for all $m \geq n_i$, $f(m) \leq 1/(5 \cdot 2^{2i+1})$
(2) and $n_{i+1} - n_i$ is divisible by 2^{2i+1}.

The idea is that in the interval $[n_i, n_{i+1})$, each element of A will encode the first i bits of X. Note that the sequence $n_0, n_1, n_2, \ldots$ is not required to be computable.

Next, for each i, let σ_i denote the length $2i+1$ binary string consisting of the first i bits of X, followed by a 1, followed by a i zeros—i.e.

$$\sigma_i = (X \restriction i)^\frown 1^\frown \underbrace{00\ldots0}_{i \text{ times}}.$$

Now define A as follows.

(1) First put $[0, n_1)$ into A.
(2) Next, for each $i \geq 1$ and $m \in [n_i, n_{i+1})$, put m into A if the binary expansion of m ends with the string σ_i.

We will now show that A has the properties desired.

Claim. *A is f-dense.*

Proof. Consider a single interval $[n_i, n_{i+1})$. Note that this interval is composed of some number of disjoint intervals of length 2^{2i+1} and that each one of these smaller intervals contains exactly one element of A. Also note that A contains all numbers less than n_1. From these two observations, it is easy to see that for each i, A is $1/2^{2i+1}$-dense at n_{i+1}. In other words, A is sufficiently dense at the endpoints of each interval $[n_i, n_{i+1})$. It remains to check that A is also sufficiently dense in the interior of these intervals.

Note that any number $m \in (n_i, n_{i+1})$ can be written as $m = n_i + k \cdot 2^{2i+1} + l$ for some k and $l < 2^{2i+1}$. Our observations above imply that A is $1/2^{2i+1}$-dense at

$n_i + k \cdot 2^{2i+1}$. We now want to show that $f(m)$ and l are small enough that A is still sufficiently dense at m.

To see why, note that by assumption, $n_i > 2^{2i-1}$ and thus $l < 4 \cdot (n_i + k \cdot 2^{2i+1})$. Thus, since A is $1/2^{2i+1}$-dense at $n_i + k \cdot 2^{2i+1}$, it is also $1/(5 \cdot 2^{2i+1})$-dense at m. By our choice of n_i, this implies that A is $f(m)$-dense at m. □

Claim. *Every infinite subset of A computes X uniformly.*

Proof. Suppose B is an infinite subset of A. For any element $m \in B$ such that $m \geq n_1$, the binary expansion of m must end with some number of zeros. Suppose it ends with k zeros. Then the binary expansion of m must end with a string σ of length k, followed by a one, followed by k zeros. By construction of A, this string σ is guaranteed to be an initial segment of X. So to compute any bit $X(i)$ of X, we simply need to find an element $m \geq n_1$ of B whose binary expansion ends with a long enough string of zeros. And such an element is guaranteed to exist by our construction of A and the fact that B is infinite. □

This concludes the proof. □

4.3. **Dense subsets of dense sets.** Our final counterexample is of a somewhat different nature than the first two. In our previous counterexamples, we considered possible strengthenings of our main theorem given by relaxing the density requirement on A. We will now consider possible strengthenings given by restricting which subsets of A we are allowed to use.

The main theorem of this paper shows that it is impossible to encode an uncomputable set X into all infinite subsets of a set A of positive lower density. But what if we only want to encode X into all subsets of A of positive lower density? Or into all subsets of positive upper density? We will show that such encoding is possible in both cases, but that it can only be done uniformly in the first case.

Proposition 4.5. *For any set X, there is some set $A \subseteq \mathbb{N}$ of positive lower density such that all subsets of A of positive lower density compute X uniformly.*

Proof. Suppose we only had to consider $1/2$-dense subsets of A. For each n, any such subset must contain at least one element of A between n and $2n$. Thus we could code one bit of X into the parity of the elements of A between n and $2n$ and be sure that any $1/2$-dense subset of A will be able to recover this bit. However, we want our procedure to work for subsets of A of any constant density greater than zero, not just $1/2$-dense subsets. So we will encode each bit of X infinitely many times, each time making a more conservative assumption about the density of the subset of A doing the decoding.

More formally, pick a computable bijection $\pi\colon \mathbb{N} \to \mathbb{N} \times \mathbb{Q}^+$ and a computable sequence $0 = n_0 < n_1 < n_2 < \ldots$ such that for all i,

(1) if $\pi(i) = (j, \delta)$ then $\delta \cdot n_{i+1} > n_i$
(2) and n_i is even.

The idea is that in the interval $[n_i, n_{i+1})$, elements of A will code the j^{th} bit of X in such a way that any δ-dense subset of A will be able to decode it.

Now define A as follows. For each $i \in \mathbb{N}$ and $m \in [n_i, n_{i+1})$, let $(j, \delta) = \pi(i)$ and put m into A if

$$\begin{cases} m \text{ is even and } X(j) = 0 \\ \text{or } m \text{ is odd and } X(j) = 1. \end{cases}$$

In other words, in the interval $[n_i, n_{i+1})$, the j^{th} bit of X is encoded into the parity of the elements of A.

First observe that A has lower density $1/2$: for each m, either $2m$ or $2m+1$ is in A (this is why we wanted to make sure each n_i is even).

Second, observe that all subsets of A of positive lower density can compute X uniformly. Given a subset $B \subseteq A$ of positive lower density and a $j \in \mathbb{N}$, we can compute $X(j)$ as follows: search for any i such that $\pi(i) = (j, \delta)$ for some δ and $[n_i, n_{i+1}) \cap B$ is nonempty. Then from the parity of any element of $[n_i, n_{i+1}) \cap B$ we can recover $X(j)$. The fact that such an i must exist follows from our choice of n_i's and the fact that B has positive lower density—if B is δ-dense and $\pi(i) = (j, \delta)$ then B must have at least one element in the interval $[n_i, n_{i+1})$ because otherwise B cannot be δ-dense at n_{i+1}. □

For the case of subsets of positive upper density, the counterexample can be constructed using a result of Bienvenu, Day and Hölzl [1].

Theorem 4.6 (Bienvenu, Day and Hölzl)**.** *For any set X, there is a set A such that for all partial functions $f \colon \mathbb{N} \to \{0,1\}$, if the domain of f has positive upper density and for all $n \in \operatorname{dom}(f)$, $f(n) = A(n)$ then f computes X.*

Corollary 4.7. *For any set X, there is a set A of positive lower density such that all subsets of A of positive upper density compute X.*

Proof. Let A be the set from Bienvenu, Day and Hölzl's theorem. Let $\widetilde{A} = \{2n \mid n \in A\} \cup \{2n+1 \mid n \notin A\}$. Note that $\widetilde{A}$ has lower density $1/2$ and that from each element of $\widetilde{A}$, we can recover one bit of the characteristic function of A.

Thus any subset B of $\widetilde{A}$ can compute a partial function $f_B \colon \mathbb{N} \to \{0,1\}$ which agrees with A everywhere it is defined. Furthermore, the density of $\operatorname{dom}(f_B)$ at any n is about twice the density of B at $2n+2$. Therefore if B has positive upper density then so does $\operatorname{dom}(f_B)$. So for any such B, our choice of A implies that f_B, and hence B itself, computes X. □

As we noted above, it is impossible to strengthen this corollary to make all subsets of A of positive upper density compute X uniformly.

Proposition 4.8. *For any uncomputable set X, set A of positive lower density and Turing functional Φ, there is some subset $B \subseteq A$ of positive upper density such that $\Phi(B) \neq X$.*

Proof. This follows directly from Lemma 3.4. In particular, using that lemma we can find a density Mathias condition (s, B) extending $(\varnothing, A)$ such that for every set C compatible with (s, B), $\Phi(C) \neq X$. In particular, if $C = s \cup B$ then $\Phi(C) \neq X$. Since (s, B) is a density Mathias condition, C has positive upper density and so we are done. □

5. Avoiding PA degree

In this section we prove the theorem promised in the introduction which modifies our main theorem to avoid PA degree rather than cones of Turing degrees.

Theorem 1.4. *For any set $A \subseteq \mathbb{N}$ of positive lower density, there is an infinite subset of A which is not of PA degree.*

In order to prove this theorem, it will be convenient to use the following alternative definition of PA degree.

Definition 5.1. Given a partial function $f\colon \mathbb{N} \to \{0,1\}$, a *completion* of f is a total function $g\colon \mathbb{N} \to \{0,1\}$ such that for every n in the domain of f, $f(n) = g(n)$.

Definition 5.2. A set A is said to be of *PA degree* if for every partial computable function $f\colon \mathbb{N} \to \{0,1\}$, A computes a completion of f.

We can thus obtain Theorem 1.4 as a corollary of the following theorem.

Theorem 5.3. *Suppose $f\colon \mathbb{N} \to \{0,1\}$ is a computable partial function with no computable completion. Then every set $A \subseteq \mathbb{N}$ of positive lower density contains an infinite subset which does not compute a completion of f.*

We show in Theorem 5.11 that the assumption that f is computable is neccesary.

Our proof of this theorem follows the same general strategy as the proof of our main theorem, with a few notable changes. First, we replace Seetapun's theorem with Liu's theorem (as discussed in the introduction) and also make some more-or-less standard changes to the proof which are relevant for avoiding PA degrees (these changes are mostly taken from the work of Liu [10] and of Monin and Patey [13]). Second, and more interestingly, we can no longer rely on the cone avoiding basis theorem (the point is that there is no such thing as a "PA degree avoiding basis theorem" for somewhat obvious reasons). Thus we are forced to replace the argument using the cone avoiding basis theorem with a somewhat more elaborate argument. Third, we need an extra combinatorial fact about intersections of dense sets. This fact is not hard to prove, but it is not as simple as the combinatorics which appeared in the proof of our main theorem.

5.1. **Liu's theorem and Liu's lemma.**

To prove the theorem above, we will use two results due to Liu. First, Liu's theorem, which, as we discussed in the introduction, is the analogue of Seetapun's theorem for this setting (and will play an analogous role in the proof). We stated a version of this theorem as Theorem 1.3, but we will use the following more general version.

Theorem 5.4 (Liu's Theorem, [10])**.** *Suppose $f\colon \mathbb{N} \to \{0,1\}$ is a computable partial function with no computable completion. Then for every set $A \subseteq \mathbb{N}$, there is an infinite set B which is a subset of either A or the complement of A such that B does not compute a completion of f.*

As with Seetapun's theorem, it is easy to use this theorem to prove the following corollary.

Corollary 5.5. *Suppose $f\colon \mathbb{N} \to \{0,1\}$ is a computable partial function and $A \subseteq \mathbb{N}$ is an infinite set which does not compute a completion of f. Then for every finite partition $B_1, \ldots, B_n$ of A, at least one of the B_i's has an infinite subset which does not compute a completion of f.*

In the course of proving Theorem 1.3, Liu implicitly used the following lemma (see Lemma 6.6 of [10]). It was also used in a slightly different form by Monin and Patey in [13] (see Lemmas 2.13 and 3.12).

Definition 5.6. A *valuation* is a finite partial function $p\colon \mathbb{N} \to \{0,1\}$ represented as a lookup table (rather than, say, a code for a Turing machine which computes p).

Definition 5.7. Given a valuation p and a partial function $f\colon \mathbb{N} \to \{0,1\}$, p is *f-correct* if $p \subseteq f$—in other words if for all n in the domain of p, n is also in the domain of f and $p(n) = f(n)$.

Lemma 5.8 (Liu [10])**.** *If W is a c.e. set of valuations and f is a computable partial function with no total completion then either W contains some f-correct valuation or for every k, there are at least k many incompatible valuations outside of W.*

Proof. Suppose W does not contain any f-correct valuations and fix a number k. For any finite set $s \subseteq \mathbb{N}$ and valuation p, say that p is *f-correct mod s* if p is f-correct when we ignore the elements of s—in other words, for all $n \in \mathrm{dom}(p) \setminus s$, n is in the domain of f and $p(n) = f(n)$. We will show that there are numbers $n_1, \ldots, n_k$ such that no element of W is f-correct mod $\{n_1, \ldots, n_k\}$. In particular, this implies that W does not contain any element with domain contained in $\{n_1, \ldots, n_k\}$. Since it is easy to see that there are at least k such incompatible functions (in fact, at least 2^k), this is sufficient to prove the lemma.

We will construct $n_1, \ldots, n_k$ by induction. Suppose we have constructed $n_1, \ldots, n_i$ and want to find n_{i+1}. In other words, we know W contains no element which is f-correct mod $\{n_1, \ldots, n_i\}$ and we want to find some $n \notin \{n_1, \ldots, n_i\}$ such that no element is f-correct mod $\{n_1, \ldots, n_i, n\}$.

Suppose for contradiction that no such n exists. Then for every $n \notin \{n_1, \ldots, n_i\}$, W contains some element p which is f-correct mod $\{n_1, \ldots, n_i, n\}$. At the same time, p cannot be f-correct mod $\{n_1, \ldots, n_i\}$. By unrolling the definitions, it is possible to see that this implies that n is in the domain of p and either not in the domain of f or $p(n) \neq f(n)$. The key point is that in either case, $1 - p(n)$ does not disagree with $f(n)$.

But this gives us a method to compute a completion g of f: For each $n \in \{n_1, \ldots, n_i\}$, hard-code some appropriate value for $g(n)$. For each $n \notin \{n_1, \ldots, n_i\}$, search for some $p \in W$ which is f-correct mod $\{n_1, \ldots, n_i, n\}$ (note that there is a computable enumeration of f-correct valuations). For the first such p that is found, set $g(n) = 1 - p(n)$. Since f has no computable completions, this gives us the desired contradiction. □

5.2. **A combinatorial lemma.**

As we referred to above, we will need the following combinatorial lemma, which says that when you have enough dense sets, two of them must have fairly dense intersection. The proof is a standard counting argument, which we phrase as a probabilistic proof based on estimating the variance of a certain random variable.

Lemma 5.9. *For every $\delta > 0$, there is some k with the following property: For any $n > 0$ and any k subsets $A_1, \ldots, A_k \subseteq [0, n)$, all of which have size at least δn, there is some pair of indices $i \neq j$ such that $\frac{A_i \cap A_j}{n} \geq \delta^2/2$.*

Proof. We will begin by thinking of k as a variable and fixing a number n and sets $A_1, \ldots, A_k \subseteq [0, n)$. We will then find a lower bound on the maximum of $\frac{A_i \cap A_j}{n}$ over all $i \neq j$. At the end, we will check that if k is large enough then this lower bound is at least $\delta^2/2$ (and does not depend on n).

So: fix n and sets $A_1, \ldots, A_k$. Consider choosing x from $[0, n)$ uniformly at random and counting the number of sets A_i which contain x. Let X be the random variable denoting this count. For each i, let 1_{A_i} be the random variable indicating whether x is in A_i or not ($1_{A_i} = 1$ if $x \in A_i$ and 0 otherwise). Note that we have

$$X = \sum_{i \leq k} 1_{A_i}.$$

The key to the proof is to estimate the variance of X. First, by a standard calculation we have

$$\begin{aligned}
\mathrm{Var}[X] &= \mathrm{E}[X^2] - \mathrm{E}[X]^2 \\
&= \mathrm{E}\Big[\big(\sum_{i \leq k} 1_{A_i}\big)^2\Big] - E\Big[\sum_{i \leq k} 1_{A_i}\Big]^2 \\
&= \sum_{i,j \leq k} \mathrm{E}\left[1_{A_i} 1_{A_j}\right] - \big(\sum_{i \leq k} \mathrm{E}[1_{A_i}]\big)^2 \\
&= \sum_{i,j} \frac{|A_i \cap A_j|}{n} - \big(\sum_i \frac{|A_i|}{n}\big)^2 \\
&= \sum_{i \neq j} \frac{|A_i \cap A_j|}{n} + \sum_i \frac{|A_i|}{n} - \big(\sum_i \frac{|A_i|}{n}\big)^2.
\end{aligned}$$

Using the fact that $\mathrm{Var}[X] \geq 0$ and letting $y = \sum_i \frac{|A_i|}{n}$, we have

$$\sum_{i \neq j} \frac{|A_i \cap A_j|}{n} \geq y^2 - y.$$

Since each A_i has size at least δn, $y \geq \delta k$. Let's assume $k > 1/\delta$ and thus $y > 1$. Since $y^2 - y$ is monotonic in y for $y > 1$, we have

$$\sum_{i \neq j} \frac{|A_i \cap A_j|}{n} \geq y^2 - y \geq (\delta k)^2 - \delta k.$$

We can now calculate a lower bound ε on the maximum value of $\frac{|A_i \cap A_j|}{n}$ over all $i \neq j$. For we have

$$\sum_{i \neq j} \frac{|A_i \cap A_j|}{n} \leq \varepsilon k(k-1) \leq \varepsilon k^2$$

and thus

$$\varepsilon k^2 \geq (\delta k)^2 - \delta k.$$

Simple algebraic manipulation then yields $\varepsilon \geq \delta^2 - \delta/k$.

We are now in position to finish the proof. Recall that we wanted k large enough that $\varepsilon \geq \delta^2/2$. The lower bound above shows that it's enough to take $k \geq 2/\delta$. Since this value does not depend on n, we are done. □

5.3. **The proof.** For the rest of this section, fix a computable partial function f with no computable completion. Define a modified version of density Mathias forcing where conditions are Mathias conditions (s, A) such that there is some infinite set D which does not compute a completion of f and so that A is dense along D. It suffices to prove the following key lemma; the rest of the proof can be completed as in the proof of Theorem 1.1.

Lemma 5.10. *For any condition (s, A) and Turing functional Φ, there is some condition $(s', A') \leq (s, A)$ such that for all sets B compatible with (s', A'), $\Phi(B)$ is not a completion of f.*

Proof. Fix a set D and number $\delta > 0$ such that A is δ-dense along D and D does not compute a completion of f.

Given a set B and a valuation p, say that p is *consistent* with B if for all finite sets $t \subseteq B$, $\Phi(s \cup t)$ and p are compatible (i.e. for every n either at least one of $\Phi(s \cup t, n)$ and $p(n)$ is not defined or they are both defined and are equal). Say that a sequence of sets $B_1, \ldots, B_l$ is *lawful* if it satisfies the following criteria:

(1) The B_i's are all disjoint.
(2) For each B_i, there are incompatible valuations p and q such that B_i is consistent with both p and q. Note that this implies that for all C compatible with (s, B_i), $\Phi(C)$ must be compatible with both p and q and hence cannot be total.
(3) There is an infinite set E such that each B_i is $\delta^2/8$-dense along E, $A \setminus (B_1 \cup \ldots \cup B_l)$ is $\delta/2$-dense along E and E does not compute a completion of f.[2]

We will prove that if there is a lawful sequence of length l then either there is a condition (s', A') with the property we want or there is a lawful sequence of length $l+1$. Also, since the B_i are disjoint and $\delta^2/8$-dense along E, there is no lawful sequence of length strictly greater than $8/\delta^2$, which suffices to finish the proof.

Finding lawful sequences. Suppose that $B_1, \ldots, B_l$ is a lawful sequence, as witnessed by E and $(p_1, q_1), (p_2, q_2), \ldots, (p_l, q_l)$. We want to either find $(s', A') \leq (s, A)$ satisfying the conclusion of the lemma or a lawful sequence of length $l+1$.[3]

For every valuation r, define $\mathcal{C}_r$ to be the set of tuples $(X_1, \ldots, X_l, Y)$ such that

(1) $X_1, \ldots, X_l, Y$ are all disjoint.
(2) Each X_i is $\delta^2/8$-dense along E.
(3) Y is $\delta/2$-dense along E.
(4) For each i, X_i is consistent with both p_i and q_i.
(5) Y is consistent with r.

Note that the sets $\mathcal{C}_r$ are uniformly $\Pi^0_1(E)$. Let W be the set of valuations r such that $\mathcal{C}_r$ is empty. Note that W is c.e. relative to E. Since E does not compute a completion of f, we can apply Liu's Lemma relative to E to get that one of the following holds.

(1) There is some $r \in W$ which is f-correct.
(2) For every k, there are at least k incompatible valuations outside of W.

Case 1: an f-correct valuation in W. Suppose that some $r \in W$ is f-correct. Thus $\mathcal{C}_r$ is empty. In particular, $(B_1, \ldots, B_l, A \setminus (B_1 \cup \ldots \cup B_l))$ is not in $\mathcal{C}_r$. By our choice of $B_1, \ldots, B_l$ and E, this can only be because r is not consistent with $A \setminus (B_1 \cup \ldots \cup B_l)$. In other words, there is some finite set $t \subset A$ and some n such that $\Phi(s \cup t, n)$ and $r(n)$ are both defined and not equal. Moreover, since r is

[2] For avoiding a cone in our main theorem, we were also able to have that $B_1 \oplus \cdots \oplus B_l \oplus E$ avoids the cone. This was because of the cone avoiding basis theorem. Here, we are not able to ask that $B_1 \oplus \cdots \oplus B_l \oplus E$ does not compute a completion of f. Before, we used $B_1, \ldots, B_l$ in the definition of $\mathcal{C}_{n,b}$, so that it was a Π^0_1 class relative to $B_1 \oplus \cdots \oplus B_k \oplus E$. Now, when defining the analogous class, it must be a Π^0_1 class relative only to E alone.

One consequence of this is that while previously we constructed our sequences $B_1, \ldots, B_l$ by extension, here, the sequence of length $l+1$ will not necessarily extend the sequence of length l.

[3] A slightly subtle point here is that we allow $l = 0$—that is, we allow the starting sequence to be empty. As in the proof of our main theorem, it is possible to check that the whole proof still works even in this case.

f-correct, this implies that $\Phi(s \cup t, n)$ and $f(n)$ are both defined and not equal. In this case we are done because we can set $(s', A') = (s \cup t, A \setminus [\max(s \cup t)])$.

Case 2: many incompatible valuations outside W. Suppose that there are k many incompatible valuations $r_1, \ldots, r_k$ outside W, where k is chosen as in the combinatorial lemma (but with $\delta/2$ rather than δ). Thus we can find sequences

$$\begin{gathered}(B_1^1, \ldots, B_l^1, C_1) \in \mathcal{C}_{r_1}\\ (B_1^2, \ldots, B_l^2, C_2) \in \mathcal{C}_{r_2}\\ \vdots\\ (B_1^k, \ldots, B_l^k, C_k) \in \mathcal{C}_{r_k}.\end{gathered}$$

Since each C_i is $\delta/2$-dense along E, it follows from the combinatorial lemma (and our choice of k) that for each $n \in E$, there are $i \neq j$ such that $C_i \cap C_j$ is $\delta^2/8$-dense at n. We can now apply the same trick we used in the proof of our main theorem (but using Liu's theorem instead of Seetapun's theorem) to get an infinite set $E_1 \subseteq E$ and a pair $i \neq j$ such that $C_i \cap C_j$ is $\delta^2/8$-dense along E_1 and E_1 does not compute a completion of f.

Now let $B'_1, B'_2, \ldots, B'_l$ denote $B^i_1, B^i_2, \ldots, B^i_l$ and let B'_{l+1} denote $C_i \cap C_j$. We have the following facts about the sequence $B'_1, \ldots, B'_{l+1}$:

(1) They are all disjoint. This follows from the facts that $B^i_1, \ldots, B^i_l$ are all disjoint from each other and from C_i and that $B'_{l+1} \subseteq C_i$.
(2) For each B'_h, there are incompatible valuations p and q such that B'_h is consistent with both p and q. For $B'_1, \ldots, B'_l$ we can just take p_h and q_h. For B'_{l+1} we can take r_i and r_j.
(3) E_1 is infinite, does not compute a completion of f and each B'_h is $\delta^2/8$-dense along E_1. For $B'_1, \ldots, B'_l$, this last part is because $E_1 \subseteq E$. For B'_{l+1}, this follows from our choice of E_1.

The point is that $B'_1, \ldots, B'_{l+1}$ is very close to a lawful sequence of length $l + 1$. All that's missing is that $A \setminus (B'_1 \cup \ldots \cup B'_{l+1})$ be $\delta/2$-dense along E_1. However, this is easily fixed.

Again using the same trick as in the proof of our main theorem, we can get an infinite set $E_2 \subseteq E_1$ which does not compute a completion of f and such that either

- $A \cap (B'_1 \cup \ldots \cup B'_{l+1})$ is $\delta/2$-dense along E_2
- or $A \setminus (B'_1 \cup \ldots \cup B'_{l+1})$ is $\delta/2$-dense along E_2.

In the first case, we can use the trick once more to find some infinite $E_3 \subseteq E_2$ which does not compute a completion of f and some $h \leq l + 1$ such that $A \cap B'_h$ is dense along E_3 and so we can finish by taking $(s', A') = (s, A \cap B'_h)$. In the latter case, we have found a lawful sequence of length $l + 1$. □

5.4. **A false generalization.** There is a common generalization of both Theorem 1.1 and Theorem 1.4 which at one point the authors thought might be true. Namely, that given *any* partial function $f \colon \mathbb{N} \to \{0, 1\}$ with no computable completion, and any set $A \subseteq \mathbb{N}$ of positive lower density, there is an infinite subset of A that does not compute any completion of f. (Theorem 1.1 is the case when f is already total, and Theorem 1.4 is the case when f is computable.) However, this is false.

Theorem 5.11. *There is a partial function $f\colon \mathbb{N} \to \{0,1\}$ and a set $A \subseteq \mathbb{N}$ of positive lower density such that any infinite subset $B \subseteq A$ computes a completion of f.*

Proof. We will define a partial function f while simulateneously defining $x_0, x_1, x_2, \ldots$. We will make use of a computable function $\pi\colon \mathbb{N} \to \mathbb{N}$ such that $\pi^{-1}(n)$ is infinite for each n. For each x_i, we will have $\pi(x_i) = i$. The idea is that while the construction will be non-computable, the function $\pi(x)$ will allow us to computably recover from x the function $\Phi_{\pi(x)}$ that x would have been used to diagonalize against if x was used to diagonalize at all.

Begin with $x_0 = 0$. Suppose that we have defined x_i. We have two cases:

(1) If $\Phi_i(x_i)\downarrow$, let s_i be the number of stages it takes to converge, and define $f(x_i) = 1 - \Phi_i(x_i)$. Let $x_{i+1} > s_i$ be such that $\pi(x_{i+1}) = i+1$.
(2) Otherwise, if $\Phi_i(x_i)\uparrow$, leave $f(x_i)$ undefined, and let $x_{i+1} > x_i$ be such that $\pi(x_{i+1}) = i+1$.

For any x outside of $\{x_0, x_1, \ldots\}$, $f(x)$ is undefined.

Now define $A \subseteq 2^{<\omega}$ as follows. Put $\sigma \in A$ if whenever $\Phi_i(x_i)\downarrow$ but $\Phi_{i,|\sigma|}(x_i)\uparrow$, $\sigma(x_i) = f(x_i)$. We claim that A has lower density at least $1/4$. For this, it is important that in identifying $2^{<\omega}$ with $\mathbb{N}$, we list out the finite binary strings in order of increasing length. Thus it suffices to show that for each n, there is some $x < n$ and $t \in \{0,1\}$ such that $A \cap 2^n \supseteq \{\sigma \in 2^n \mid \sigma(x) = t\}$. If not, there would be $i < j$ with $x_i < x_j < n$ such that

(1) $\Phi_i(x_i)\downarrow$ but $\Phi_{i,n}(x_i)\uparrow$ and
(2) $\Phi_j(x_j)\downarrow$ but $\Phi_{j,n}(x_j)\uparrow$.

From (1), it follows that $n < s_i$, but by construction $x_j > s_i$. This contradicts the choice of $x_j < n$.

Now suppose that B is an infinite subset of A. We will compute from B a completion g of f. For each x, look for some $\sigma \in B$ with $\sigma(x)\downarrow$. Check whether $\Phi_{\pi(x),|\sigma|}(x)\downarrow$. If so, define $g(x) = 1 - \Phi_{\pi(x),|\sigma|}(x)$. Otherwise, define $g(x) = \sigma(x)$. It is easy to see that g is total.

To see that g is a completion of f, it suffices to check that for all i, $g(x_i) = f(x_i)$. Let $\sigma \in B$ be the string used to define $g(x_i)$. If $\Phi_{i,|\sigma|}(x_i)\downarrow$, then we defined both $g(x_i)$ and $f(x_i)$ to be $1 - \Phi_i(x_i)$. Otherwise, if $\Phi_{i,|\sigma|}(x_i)\uparrow$, then we defined $g(x_i) = \sigma(x_i)$, and if $f(x_i)$ is defined then $\Phi_i(x_i)\downarrow$ and so, as $\sigma \in A$, $\sigma(x_i) = f(x_i)$. □

6. Preserving Kolmogorov complexity

Recall from the introduction that for any string σ and any collection of sets $\mathcal{F} \subseteq \mathcal{P}(\mathbb{N})$, we define

$$C(\sigma \mid \mathcal{F}) = \max_{X \in \mathcal{F}} C^X(\sigma).$$

Also recall that $C(\sigma) - C(\sigma \mid [A]^\omega)$ can be thought of as the number of bits of information about σ coded into all infinite subsets of A. The goal of this section is to investigate this quantity in the case where A is a set of positive lower density.

We noted in the introduction that it is possible to encode about $\log(1/\delta)$ bits of information into all infinite subsets of a set of lower density δ. For example, if $\delta = 1/2^n$ then for any binary string σ of length n, we can encode σ into all infinite subsets of a set A of lower density $1/2^n$ by letting A consist of all numbers m such that the last (i.e. least significant) bits of the binary expansion of m match σ.

However, this is not the only way to encode information into all infinite subsets of a dense set. Roughly speaking, for any fixed number N, it is possible to encode "an arbitrary integer greater than N." We will now make this precise.

Definition 6.1. For any string $\sigma \in 2^{<\omega}$ and number $N \in \mathbb{N}$, define

$$C(\sigma \mid \geq N) = \max_{n \geq N} C(\sigma \mid n).$$

Proposition 6.2. *For any string σ and number N, there is some set A of lower density 1 such that*

$$C(\sigma \mid [A]^\omega) \leq C(\sigma \mid \geq N) + O(1)$$

where the constant hidden by the $O(1)$ does not depend on σ or N.

Proof. Let $A = \{n \mid n \geq N\}$. If B is an infinite subset of A then all elements of B are greater than or equal to N. In other words, from any infinite subset of A we can uniformly extract a number greater than or equal to N. Thus $C(\sigma \mid [A]^\omega)$ is at most $C(\sigma \mid \geq N) + O(1)$, where the extra constant represents the complexity of the extraction procedure. □

It may not be immediately apparent exactly how much information can be gained from knowing "an arbitrary integer greater than N" (i.e. how large $C(\sigma) - C(\sigma \mid \geq N)$ can be). A result of Vereshchagin [16] gives a precise characterization.

Theorem 6.3 (Vereshchagin)**.** *For any string σ,*

$$C^{0'}(\sigma) = \min_N C(\sigma \mid \geq N) \pm O(1).$$

Together with our proposition above, this shows that for any string σ, it is possible for all infinite subsets of a dense set A to lower the complexity of σ to its $0'$ complexity. This can be combined with the first coding method we mentioned to give a result claimed in the introduction.

Proposition 1.6. *For any string σ and $\delta \in (0, 1]$, there is some set $A \subseteq \mathbb{N}$ of lower density at least δ such that*

$$C(\sigma \mid [A]^\omega) \leq \max(0, C^{0'}(\sigma) - \log(1/\delta)) + O(\log\log(1/\delta)).$$

Proof. Let $k = C^{0'}(\sigma)$ and let ρ be a string witnessing this fact—i.e. a string such that $|\rho| = k$ and for the universal oracle machine U, $U^{0'}(\rho) = \sigma$. It is a standard fact about Kolmogorov complexity that there is some N such that for all $n \geq N$, $C(\sigma \mid \rho, n) = O(1)$ (this fact is more or less the easy direction of Vereshchagin's theorem mentioned above).

Next, let $m \in \mathbb{N}$ be such that $2^{-m-1} < \delta \leq 2^{-m}$ and define A by

$$\begin{aligned} A = \{n \geq N \mid{} & \text{the binary representation of } n \bmod 2^m \\ & \text{is equal to the first } m \text{ bits of } \rho\}. \end{aligned}$$

If the length of ρ is less than m then first extend ρ to length m by adding a 1 followed by a run of 0s to the end.

Note that the lower density of A is 2^{-m}, which is greater than or equal to δ. Furthermore, from any infinite subset $B \subseteq A$, we can uniformly extract a number $n \geq N$. And if we know m then we can also uniformly extract the first m bits of ρ. Since $C(\sigma \mid \rho, n) = O(1)$, to reconstruct σ from this information, we only need to know the last $k - m$ bits of ρ. In other words, if we are given an infinite subset

$B \subseteq A$ then to reconstruct σ we just need to know m and the last $k - m$ bits of ρ. Thus we have

$$C(\sigma \mid [A]^\omega) \leq (k - m) + O(C(m)) \leq k - m + O(\log(m)).$$

Since $\log(1/\delta) \geq m \geq \log(1/\delta) - 1$, this gives us

$$C(\sigma \mid [A]^\omega) \leq k - \log(1/\delta) + O(\log\log(1/\delta))$$

as desired. □

In this section, we will show that this upper bound is essentially optimal. In particular, we will prove the following theorem.

Theorem 1.7. *For any string σ and set $A \subseteq \mathbb{N}$ of lower density at least $\delta \in (0, 1]$,*

$$C(\sigma \mid [A]^\omega) \geq C^{0'}(\sigma) - \log(1/\delta) - O(\log\log(1/\delta))$$

where the constant hidden by the $O(\cdot)$ notation does not depend on σ or A.

It is also possible to prove a corresponding result for Seetapun's theorem. In particular, for any set A, define

$$\mathsf{RT}^1_2(A) = [A]^\omega \cup [\overline{A}]^\omega.$$

The quantity $C(\sigma) - C(\sigma \mid \mathsf{RT}^1_2(A))$ can be viewed as the number of bits of information about σ that can be encoded into all infinite subsets of A and $\overline{A}$. Just as in the case of subsets of a dense set, it is always possible to lower the complexity of σ to its $0'$ complexity.

Proposition 6.4. *For any string σ, there is some set $A \subseteq \mathbb{N}$ such that*

$$C(\sigma \mid \mathsf{RT}^1_2(A)) \leq C^{0'}(\sigma) + O(1).$$

Proof. Once again, by Vereshchagin's theorem it is enough to prove that for any N there is some set A such that $C(\sigma \mid \mathsf{RT}^1_2(A)) \leq C(\sigma \mid \geq N) + O(1)$. And for this, we can simply take $A = \{n \mid n \geq N\}$. □

Just like in the case of dense sets, it is possible to show that this is optimal (and unlike in the case of dense sets, the error term here is constant).

Theorem 6.5. *For any string σ and set $A \subseteq \mathbb{N}$,*

$$C(\sigma \mid \mathsf{RT}^1_2(A)) \geq C^{0'}(\sigma) - O(1)$$

where the constant hidden by the $O(1)$ does not depend on σ or A.

The remainder of this section is focused on proving Theorems 1.7 and 6.5. As a warm-up, and to demonstrate our proof strategy, we will first prove a much easier version of Theorem 6.5. We will then prove the full theorem and, finally, adapt the techniques from the proof to prove Theorem 1.7.

6.1. **Kolmogorov complexity and Seetapun's theorem: easy version.** We will now prove a version of Theorem 6.5 where the $0'$ oracle is replaced by an oracle for a complete Σ^1_2 set. In particular, let W denote a complete Σ^1_2 set and fix a string σ and set $A \subseteq \mathbb{N}$. We will prove that

$$C(\sigma \mid \mathsf{RT}^1_2(A)) \geq C^W(\sigma) - O(1).$$

This may appear a bit absurd—W is ridiculously powerful—but it helps demonstrate the basic strategy that we will use in our proofs of Theorems 6.5 and 1.7.

Here's the core idea of the proof. Suppose k is a number such that $C(\sigma \mid \mathsf{RT}^1_2(A)) < k$. We would like to show that $C^W(\sigma) \leq k + O(1)$. To do this, we will identify some property of σ which is shared by at most 2^k other strings and which can be recognized using an oracle for W. One obvious special property of σ is that there is a set X such that $C(\sigma \mid \mathsf{RT}^1_2(X)) < k$. It is not hard to show that at most 2^k strings have this property and that W is powerful enough to check which strings have it (in fact, we chose W to make this part obvious).

We will now give a more formal version of this argument. We will describe a program E with the following properties.

- For each k, $E^W(k)$ enumerates a set of at most 2^k strings.
- For any set A, string σ and number k such that $C(\sigma \mid \mathsf{RT}^1_2(A)) < k$, σ is in the set enumerated by $E^W(k)$.

By general properties of Kolmogorov complexity, for any number k and string τ enumerated by $E^W(k)$, $C^W(\tau) \leq k + O(1)$. In particular, if $C(\sigma \mid \mathsf{RT}^1_2(A)) < k$ then $C^W(\sigma) \leq k + O(1)$.

We will now describe E. First, say that a string τ is *good-for-k* if there is a set X such that $C(\tau \mid \mathsf{RT}^1_2(X)) < k$. We claim that both of the following hold.

(1) It is computable in W to check if a string τ is good-for-k (i.e. the set of pairs (τ, k) such that τ is good-for-k is computable relative to W).
(2) There are at most 2^k strings which are good-for-k.

To see why the first claim holds, simply note that the statement "τ is good-for-k" is equivalent to the statement "there is some X such that for every Y, if $Y \subseteq X$ or $Y \subseteq \overline{X}$, then $C^Y(\tau) < k$," which is a Σ^1_2 formula of τ and k.

To see why the second claim holds, suppose that $\tau_1, \ldots, \tau_l$ are distinct good-for-k strings, as witnessed by $X_1, \ldots, X_l$. Note that the collection of all Boolean combinations of the sets $X_1, \ldots, X_l$ forms a finite partition of $\mathbb{N}$. Thus one of these Boolean combinations is infinite. Let B be this Boolean combination and note that for each $i \leq l$, B is either a subset of X_i or of $\overline{X_i}$ and, either way, is an element of $\mathsf{RT}^1_2(X_i)$. Therefore for each $i \leq l$, $C^B(\tau_i) < k$, which is impossible unless $l \leq 2^k$.

The program $E^W(k)$ works as follows. Given the input k, it goes through all strings in some fixed order and uses W to check whether each one is good-for-k. Each time it finds a string which is good-for-k, it enumerates it. As we showed above, $E^W(k)$ will never enumerate more than 2^k strings. Also, for any set A, string σ and number k such that $C(\sigma \mid \mathsf{RT}^1_2(A)) < k$, it is obvious that σ is good-for-k and thus will be enumerated by $E^W(k)$. It follows that for such a σ and k, $C^W(\sigma) \leq k + O(1)$.

6.2. **Kolmogorov complexity and Seetapun's theorem: hard version.** We will now prove Theorem 6.5, which states that for any string σ and set $A \subseteq \mathbb{N}$, $C(\sigma \mid \mathsf{RT}^1_2(A)) \geq C^{0'}(\sigma) - O(1)$. Our strategy is the same as in the previous subsection—i.e. let's assume that $C(\sigma \mid \mathsf{RT}^1_2(A)) < k$ and identify a property of σ

which is shared by at most 2^k strings and which can be recognized using an oracle for $0'$.

In identifying such a property, there is a natural trade-off between how easy the property is to describe and the computational power required to check if a string has the property. In the previous subsection, we just needed a property which can be recognized using a complete Σ^1_2 set and this allowed us to use a very straightforward property of σ. Now, however, we want a property that can be recognized by $0'$, which forces us to use a property that has a more intricate description. The key definition is the following.

Definition 6.6. A finite set of strings F is *k-safe* if there is some finite[4] partition $X_1, \ldots, X_n$ of $\mathbb{N}$ and some number m such that for all $i \leq n$ and all finite subsets $s \subseteq X_i$,

$$|s| \geq m \implies |\{\tau \mid C^s(\tau) < k\} \cup F| \leq 2^k.$$

The idea is that F is k-safe as witnessed by a partition $X_1, \ldots, X_n$ if for any X_i and any infinite subset B of X_i, we may safely assume that B will assign all strings in F complexity less than k. More specifically, it may not actually be the case that B assigns a complexity less than k to each string in F, but if we assume that it does then we will never see a contradiction of the form "B assigns complexity less than k to too many strings."

Essentially the key property of σ is that it is a member of every maximal k-safe set. More precisely, we will show that any maximal k-safe set has size at most 2^k and contains σ and that there is some maximal k-safe set which can be enumerated by $0'$, uniformly in k. The key facts are

(1) Every k-safe set has size at most 2^k.
(2) If F is k-safe then so is $F \cup \{\sigma\}$.
(3) The collection of k-safe sets is c.e. relative to $0'$ (uniformly in k).

Before proving these three facts, let's see how they can be used to finish the proof. Consider the following procedure for using $0'$ to enumerate a maximal k-safe set.

```
1. Set $F = \varnothing$
2. While true:
3.   Search for a string $\tau$ such that $F \cup \{\tau\}$ is $k$-safe
4.   If such a $\tau$ is found, enumerate $\tau$ and set $F = F \cup \{\tau\}$
```

It is easy to show by induction that at every step of the above algorithm, F is k-safe. Since no k-safe set has size more than 2^k, the above algorithm can add at most 2^k strings to F. Thus after some point, no new strings will be added to F. We claim that at this point, σ must be an element of F. If not, then since $F \cup \{\sigma\}$ is k-safe, eventually the algorithm will discover this fact and add σ to F, contradicting our assumption that no new elements are added to F. All this implies that the algorithm must eventually enumerate σ and also that the algorithm will enumerate at most 2^k strings.

To finish, observe that the above procedure was uniform in k. In other words, there is a single program E such that for each l, $E^{0'}(l)$ carries out the above algorithm with $k = l$. As in the previous subsection, for every number l and string τ enumerated by $E^{0'}(l)$, $C^{0'}(\tau) \leq l + O(1)$. Since σ is enumerated by $E^{0'}(k)$, this shows that $C^{0'}(\sigma) \leq k + O(1)$, as desired.

[4] We could take $n = 2^{|F|}$ but this is not needed.

We will now prove each of the three facts above.

Lemma 6.7. *Every k-safe set has size at most 2^k.*

Proof. Suppose F is a k-safe set. Let $X_1, \ldots, X_n$ be a partition witnessing that F is k-safe. Since $X_1, \ldots, X_n$ is a finite partition of $\mathbb{N}$, some X_i must be infinite. It is straightforward to check that since X_i is infinite, we must have

$$|\{\tau \mid C^{X_i}(\tau) < k\} \cup F| \leq 2^k$$

which implies that $|F| \leq 2^k$. □

Lemma 6.8. *If F is a k-safe set then so is $F \cup \{\sigma\}$.*

Proof. Suppose that F is k-safe as witnessed by $X_1, \ldots, X_n$ and m. Recall that the string σ and set A have the property that for all infinite subsets B of A or $\overline{A}$, $C^B(\sigma) < k$. The main idea of this proof is that the fact that $F \cup \{\sigma\}$ is k-safe can be witnessed by the partition

$$X_1 \cap A, X_1 \cap \overline{A}, \ldots, X_n \cap A, X_n \cap \overline{A}.$$

More precisely, let m' be a number larger than both the size of any finite piece of this partition and m. We will show that $F \cup \{\sigma\}$ is k-safe as witnessed by m' and $X_1 \cap A, X_1 \cap \overline{A}, \ldots, X_n \cap A, X_n \cap \overline{A}$.

To prove this claim, suppose s is a finite subset of some set in this partition and $|s| \geq m'$. Without loss of generality, let's assume that $s \subseteq X_1 \cap A$. By our choice of m' and the assumption that $|s| \geq m'$, $X_1 \cap A$ must be infinite. Thus we can find some subset $B \subseteq X_1 \cap A$ which agrees with s below $\max(s)$ and which is infinite. It is straightforward to check the following facts

(1) Since B is an infinite subset of X_1, $|\{\tau \mid C^B(\tau) < k\} \cup F| \leq 2^k$.
(2) Since s is an initial segment of B, $\{\tau \mid C^s(\tau) < k\} \subseteq \{\tau \mid C^B(\tau) < k\}$.
(3) Since B is an infinite subset of A, $C^B(\sigma) < k$ and thus $\sigma \in \{\tau \mid C^B(\tau) < k\}$.

Putting these together, we have

$$\begin{aligned}\{\tau \mid C^s(\tau) < k\} \cup F \cup \{\sigma\} &\subseteq \{\tau \mid C^B(\tau) < k\} \cup F \cup \{\sigma\}\\ &= \{\tau \mid C^B(\tau) < k\} \cup F\end{aligned}$$

and thus

$$|\{\tau \mid C^s(\tau) < k\} \cup F \cup \{\sigma\}| \leq |\{\tau \mid C^B(\tau) < k\} \cup F| \leq 2^k$$

as desired. □

Lemma 6.9. *The collection of k-safe sets is c.e. relative to $0'$, uniformly in k.*

Proof. Given a number k, a finite set of strings F, a number m and sets $X_1, \ldots, X_n \subseteq \mathbb{N}$, the statement that m and $X_1, \ldots, X_n$ witness that F is k-safe is uniformly Π^0_1. In other words, there is some computable tree $T(k, F, m, n)$ such that $X_1 \oplus \ldots \oplus X_n$ is a path through $T(k, F, m, n)$ if and only if m and $X_1, \ldots, X_n$ witness that F is k-safe and, furthermore, $T(k, F, m, n)$ is uniformly computable in k, F, m and n.

By König's lemma, $T(k, F, m, n)$ has an infinite path if and only if $T(k, F, m, n)$ is itself infinite. Thus to check if F is k-safe, it suffices to check if there are m and n such that $T(k, F, m, n)$ is infinite. This is a Σ^0_2 property (uniformly in k and F) and thus c.e. relative to $0'$. □

6.3. **Kolmogorov complexity and dense sets.** We will now prove Theorem 1.7. Actually, we will instead prove the following apparently weaker version and then derive Theorem 1.7 as a corollary.

Theorem 6.10. *For any string σ and set $A \subseteq \mathbb{N}$ which is δ-dense for $\delta \in (0,1]$,*

$$C(\sigma \mid [A]^\omega) \geq C^{0'}(\sigma) - \log(1/\delta) - O(\log\log(1/\delta))$$

where the constant hidden by the $O(\cdot)$ notation does not depend on σ or A.

Note that this theorem differs from Theorem 1.7 in that the set A is required to be δ-dense rather than to have lower density δ (essentially A is required to be dense everywhere rather than just dense at all sufficiently large points).

We will begin by showing how to use this theorem to prove Theorem 1.7.

Proof of Theorem 1.7 using Theorem 6.10. The basic point is that for any string σ and set A of lower density δ, there is a set $\widetilde{A}$ which is $\delta/4$-dense such that

$$C(\sigma \mid [\widetilde{A}]^\omega) \leq C(\sigma \mid [A]^\omega) + O(1)$$

and thus Theorem 6.10 implies that

$$\begin{aligned} C(\sigma \mid [A]^\omega) &\geq C(\sigma \mid [\widetilde{A}]^\omega) - O(1) \\ &\geq C^{0'}(\sigma) - \log(4/\delta) - O(\log\log(4/\delta)) \\ &= C^{0'}(\sigma) - \log(1/\delta) - O(\log\log(1/\delta)). \end{aligned}$$

The set $\widetilde{A}$ is simple to describe. Let N be large enough that for all $n \geq N$, A has density at least $\delta/2$ at n. Then define

$$\widetilde{A} = \{n \mid n \text{ is even and } n \leq 2N\} \cup \{2n+1 \mid n \in A\}.$$

In other words, $\widetilde{A}$ contains a copy of A on the odd numbers, together with enough even numbers to make sure $\widetilde{A}$ is relatively dense everywhere.

To finish, note that any infinite subset of $\widetilde{A}$ must contain an infinite number of odd numbers and thus can uniformly compute an infinite subset of A and so we have

$$C(\sigma \mid [\widetilde{A}]^\omega) \leq C(\sigma \mid [A]^\omega) + O(1)$$

as desired. □

We will now prove Theorem 6.10. The proof closely follows the proof from the previous subsection, but with a few additional difficulties. We will begin by modifying the definition of a k-safe set.

Definition 6.11. For any $k \in \mathbb{N}$ and rational $\delta \in (0,1]$, a finite set of strings $F = \{\tau_1, \ldots, \tau_n\}$ is *k-safe at density* δ if there is a number m and sets $X_1, \ldots, X_n \subseteq \mathbb{N}$ of density at least δ such that for all $i_1 < i_2 < \ldots < i_l$ and all finite subsets $s \subseteq X_{i_1} \cap \ldots \cap X_{i_l}$,

$$|s| \geq m \implies |\{\tau \mid C^s(\tau) < k\} \cup \{\tau_{i_1}, \ldots, \tau_{i_l}\}| \leq 2^k.$$

As in the previous subsection, we have the following claims.

(1) If F is k-safe at density δ then $|F| \leq 2^k/\delta$.
(2) Suppose we have a set A and string σ such that A is δ dense and $C(\sigma \mid [A]^\omega) < k$. If F is k-safe at density δ then so is $F \cup \{\sigma\}$.

(3) The collection of sets which are k-safe at density δ is c.e. relative to $0'$, uniformly in k and δ.

Before proving these claims, let's see how to use them to finish the proof. In a similar fashion to the previous subsection, we can find a program E such that for any k and n, $E^{0'}(k, n)$ enumerates a maximal set which is k-safe at density 2^{-n}. Furthermore, $E^{0'}(k, n)$ enumerates at most $2^k \cdot 2^n$ strings. By standard facts about Kolmogorov complexity, this implies that for any τ which is enumerated by $E^{0'}(k, n)$, $C^{0'}(\tau \mid n) \leq k + n + O(1)$. Thus for such a τ we have

$$C^{0'}(\tau) \leq C^{0'}(\tau \mid n) + O(C^{0'}(n)) \leq k + n + O(C^{0'}(n))$$

and so

$$C^{0'}(\tau) \leq k + n + O(\log(n)).$$

Now fix a string σ and a set A of density at least $\delta \in (0, 1/2]$. Let $k = C(\sigma \mid [A]^\omega) + 1$ and n be such that $2^{-n} < \delta \leq 2^{-n+1}$. Then by the facts above, we have that σ is enumerated by $E^{0'}(k, n)$ and thus

$$C^{0'}(\sigma) \leq k + n + O(\log(n)) = k + \log(1/\delta) + O(\log\log(1/\delta)).$$

Note that in the calculation above we used that $n - 1 \leq \log(1/\delta) < n$.

We will now prove the claims above.

Lemma 6.12. *Suppose F is k-safe at density δ. Then $|F| \leq 2^k/\delta$.*

Proof. Suppose $F = \{\tau_1, \ldots, \tau_n\}$ and that the sets $X_1, \ldots, X_n$ witness that F is k-safe at density δ.

We first claim that there is some collection of at least δn many X_i's whose intersection is infinite. To see why, suppose for contradiction that each such collection is finite. Then there is some number N large enough that every $x \geq N$ is in fewer than δn of the X_i's. Fix some N' much larger than N and consider a random point x from the interval $[N, N')$. We will derive a contradiction by calculating the expected number of X_i's which contain x.

First consider a single fixed X_i. Since X_i is δ dense at N', X_i has at least $\delta N'$ elements less than N' and thus has at least $\delta N' - N$ elements in the interval $[N, N')$. So the probability that x is in X_i is at least

$$\frac{\delta N' - N}{N'} = \delta - \frac{N}{N'}.$$

By linearity of expectation, this implies that the expected number of X_i's which contain x is at least

$$\delta n - \frac{Nn}{N'}.$$

Thus there is some x in the interval $[N, N')$ which is contained in at least $\delta n - \frac{Nn}{N'}$ many X_i's. Since the number of X_i's containing x must be an integer, this x must actually be contained in

$$\left\lceil \delta n - \frac{Nn}{N'} \right\rceil$$

many X_i's. If N' is large enough then this is simply equal to $\lceil \delta n \rceil$. In other words, x is contained in at least δn many X_i's, contradicting our assumption.

We have now shown that there is some collection of at least δn many X_i's whose intersection is infinite. Suppose for convenience that these X_i's are simply

$X_1, \ldots, X_l$ where $l = \lceil \delta n \rceil$ and let $B = X_1 \cap \ldots \cap X_l$. Note that by definition of "k-safe at density δ," we must have

$$|\{\tau \mid C^B(\tau) < k\} \cup \{\tau_1, \ldots, \tau_l\}| \leq 2^k,$$

which is impossible unless $l \leq 2^k$. To summarize, we have $\delta n \leq 2^k$, which implies that $n \leq 2^k/\delta$, as desired. □

Lemma 6.13. *Suppose A is δ dense, $C(\sigma \mid [A]^\omega) < k$ and F is k-safe at density δ. Then $F \cup \{\sigma\}$ is also k-safe at density δ.*

Proof. The proof is essentially the same as the proof of Lemma 6.8. Namely, let $F = \{\tau_1, \ldots, \tau_n\}$ and fix sets $X_1, \ldots, X_n$ and a number m witnessing that F is k-safe at density δ. Let $m' > m$ be large enough that for any intersection of finitely many of $X_1, \ldots, X_n, A$, if that intersection happens to be finite, then it has at most m' elements. By copying the proof of Lemma 6.8, one can show that $F \cup \{\sigma\}$ is k-safe at density δ, as witnessed by $X_1, \ldots, X_n, A$ and m'. □

Lemma 6.14. *The collection of sets which are k-safe at density δ is c.e. relative to $0'$, uniformly in k and δ.*

Proof. The proof is essentially the same as the proof of Lemma 6.9. The key point is that the statement "$\{\tau_1, \ldots, \tau_n\}$ is k-safe at density δ, as witnessed by $X_1, \ldots, X_n$ and m" is Π^0_1. □

6.4. **Open questions.** The presence of the $0'$ oracle in Theorems 6.5 and 1.7 is somewhat unsatisfying. In one sense it is necessary: Propositions 6.4 and 1.6 show that it cannot be removed. However, there is another sense in which it may be possible to strengthen the two theorems.

For the sake of concreteness, we will focus on a possible strengthening of Theorem 6.5. In that theorem, we showed that for any set A and string σ, $C(\sigma \mid \mathsf{RT}^1_2(A)) \geq C^{0'}(\sigma) - O(1)$. By Vereshchagin's theorem, this is equivalent to showing that for each string σ, there is some number N such that $C(\sigma \mid \mathsf{RT}^1_2(A)) \geq C(\sigma \mid \geq N) - O(1)$. However, this allows the possibility that even for a fixed set A, there is a different such N for each σ. It seems natural to wonder whether this is necessary, or whether, for each set A, there is a single N which works for all σ.

Conjecture 6.15. *For any set $A \subseteq \mathbb{N}$, there is a number N such that for any string σ,*

$$C(\sigma \mid \mathsf{RT}^1_2(A)) \geq C(\sigma \mid \geq N) - O(1).$$

Note that the set A we constructed to show that the bound in Theorem 6.5 is tight is not a counterexample to this conjecture: for that set A there is a single number N such that for all strings σ, $C(\sigma \mid \mathsf{RT}^1_2(A)) = C(\sigma \mid \geq N) \pm O(1)$.

One could also ask for a similar strengthening of Theorem 1.7.

Conjecture 6.16. *For any set $A \subseteq \mathbb{N}$ of lower density δ, there is a number N and a string τ of length at most $\log(1/\delta)$ such that for any string σ,*

$$C(\sigma \mid [A]^\omega) \geq C(\sigma \mid \geq N, \tau) - O(\log\log(1/\delta)).$$

There is another conjecture, also related to Theorem 1.1 and Kolmogorov complexity, but quite distinct from the conjectures above, that we would like to mention here. Namely, it might be possible to strengthen Theorem 1.1 to say that given A

of positive lower density, and X uncomputable, not only is there a subset of A that does not compute X, but there is a subset of A which does not help to compress initial segments of X.

Conjecture 6.17. *Let X be any set. For any set $A \subseteq \mathbb{N}$ of lower density δ there is an infinite subset $B \subseteq A$ such that for infinitely many n,*

$$C^B(X \upharpoonright n) \geq C(X \upharpoonright n) - O(1).$$

Here we take inspiration from a theorem due to Chaitin (see [4], Theorem 3.4.4) that X is computable if and only if $C(X \upharpoonright n) \leq C(n) + O(1)$ for all n. If X is uncomputable, there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that $\limsup_{n\to\infty} f(n) = \infty$ and for all n, $C(X \upharpoonright n) \geq C(n) + f(n) - O(1)$. If $B \subseteq A$ is an infinite set which preserves the complexity of infinitely many initial segments of X, then for infinitely many n,

$$C^B(X \upharpoonright n) \geq C(X \upharpoonright n) - O(1) \geq C(n) + f(n) - O(1) \geq C^B(n) + f(n) - O(1),$$

and hence X is not computable relative to B, which is exactly the conclusion of Theorem 1.1.

Note that in this last conjecture we must only ask that $C^B(X \upharpoonright n) \geq C(X \upharpoonright n) - O(1)$ for infinitely many n, rather than for all n, as whenever $B \subseteq A$ is uncomputable it will given shorter descriptions for infinitely many n, and thus B can further compress even some initial segments of the empty set $X = \varnothing$.

7. The relationship between the main theorem and Seetapun's theorem

It seems clear that Theorem 1.1 and Seetapun's theorem are closely related. Both give limitations on coding information into all infinite subsets of a set and we used Seetapun's theorem in our proof of Theorem 1.1. However, the two theorems are even more closely related than this suggests. In particular, Theorem 1.1 directly implies Seetapun's theorem.

To see why, let X be uncomputable and $A \subseteq \mathbb{N}$ be arbitrary. Consider the following set $B \subseteq \mathbb{N}$ of lower density $1/2$:

$$B = \{2n \mid n \in A\} \cup \{2n+1 \mid n \notin A\}.$$

Given any subset $C \subseteq B$, we can read off a subset C_{even} of A by looking at the even elements of C and a subset C_{odd} of $\overline{A}$ by looking at the odd elements of C. Moreover, if C is infinite then at least one of C_{even} and C_{odd} must be infinite as well. By Theorem 1.1, B must have some infinite subset C that does not compute X. But then either C_{even} is an infinite subset of A which does not compute X or C_{odd} is an infinite subset of $\overline{A}$ which does not compute X.

The argument above does not constitute a new proof of Seetapun's theorem because we used Seetapun's theorem in our proof of Theorem 1.1. However, it might make one wonder whether there is a similar direct proof of Theorem 1.1 from Seetapun's theorem (in which case, the somewhat complicated proof of Theorem 1.1 that we gave in Section 3 would be pointless). In this section, we will show that this is not the case. However, we first have to make precise what sort of proof we are trying to rule out.

7.1. Strong omniscient computable reducibility.

The proof of Seetapun's theorem from Theorem 1.1 that we gave above can be seen as an example of a *strong computable reduction*, a notion closely related to Weihrauch reducibility and reverse math which was first introduced by Dzhafarov [5].

Definition 7.1. Given two partial functions $P, Q\colon 2^\omega \to \mathcal{P}(2^\omega)$, P is *strongly computably reducible to* Q, written $P \leq_{sc} Q$, if for any $x \in \operatorname{dom}(P)$,

(1) there is some $\widetilde{x} \leq_T x$ such that $\widetilde{x} \in \operatorname{dom}(Q)$
(2) and for any $\widetilde{y} \in Q(\widetilde{x})$, there is some $y \leq_T \widetilde{y}$ such that $y \in P(x)$.

Intuitively, a partial function $P\colon 2^\omega \to \mathcal{P}(2^\omega)$ can be seen as a problem: elements x of the domain of P are *instances* of the problem and elements y of $P(x)$ are *solutions* to the instance x. Under this interpretation, a problem P is strongly computably reducible to a problem Q if any instance x of the problem P can compute an instance $\widetilde{x}$ of Q such that any solution to $\widetilde{x}$ can compute a solution to the original problem x.

As we said, the proof of Seetapun's theorem from Theorem 1.1 that we gave above can be seen as an example of a strong computable reduction. First, we can define problems corresponding to Seetapun's theorem and Theorem 1.1:

- RT^1_2 is the problem in which an instance is a set $A \subseteq \mathbb{N}$ and a solution to that instance is an infinite subset of either A or $\overline{A}$.
- SD (which stands for "subset of a dense set") is the problem in which an instance is a set $A \subseteq \mathbb{N}$ of positive lower density and a solution is an infinite subset of A.

Next, we can translate the main idea of the proof into a strong computable reduction of RT^1_2 to SD:

(1) Given an instance A of RT^1_2, we can compute the instance $B = \{2n \mid n \in A\} \cup \{2n+1 \mid n \notin A\}$ of SD.
(2) Given a solution C to the instance B (i.e. an infinite subset $C \subseteq B$), we can compute a solution to the instance A of RT^1_2 (namely, either C_{even} or C_{odd}).

Furthermore, it is easy to check that if P and Q are problems such that $P \leq_{sc} Q$ and X is a set such that there is some instance P, all of whose solutions compute X then there is some instance of Q all of whose solutions compute X. Thus the fact that $\mathsf{RT}^1_2 \leq_{sc} \mathsf{SD}$ plus Theorem 1.1 implies Seetapun's theorem.

So to show that there is no proof of this sort of Theorem 1.1 from Seetapun's theorem, we can show that there is no strong computable reduction of SD to RT^1_2. In fact, we will prove a somewhat stronger statement: instead of strong computable reductions of SD to RT^1_2, we will consider *strong omniscient computable reductions.*

Strong omniscient computable reducibility is a natural weakening of strong computable reducibility, introduced by Monin and Patey [11], in which the transformation of instances is not required to be computable (though the transformation of solutions still is).

Definition 7.2. Given two partial functions $P, Q\colon 2^\omega \to \mathcal{P}(2^\omega)$, P is *strongly omnisciently computably reducible to* Q, written $P \leq_{soc} Q$, if for any $x \in \operatorname{dom}(P)$,

(1) there is some $\widetilde{x}$ such that $\widetilde{x} \in \operatorname{dom}(Q)$
(2) and for any $\widetilde{y} \in Q(\widetilde{x})$, there is some $y \leq_T \widetilde{y}$ such that $y \in P(x)$.

Note that the only difference from strong computable reducibility is that here, $\widetilde{x}$ is not required to be computable from x.

It is clear that $P \leq_{sc} Q$ implies $P \leq_{soc} Q$, but not always vice-versa, and hence showing that $\mathsf{SD} \not\leq_{soc} \mathsf{RT}^1_2$ is a stronger result than showing that $\mathsf{SD} \not\leq_{sc} \mathsf{RT}^1_2$.

7.2. **Theorem 1.1 is not reducible to Seetapun's theorem.** We will now prove that $\mathsf{SD} \not\leq_{soc} \mathsf{RT}^1_2$. In our proof, we will construct infinite sets A and G such that A has lower density at least $1/2$ and no infinite subset of G computes any infinite subset of A.

To see why constructing such an A and G is sufficient to prove $\mathsf{SD} \not\leq_{soc} \mathsf{RT}^1_2$, suppose for contradiction that such an A and G exist and that $\mathsf{SD} \leq_{soc} \mathsf{RT}^1_2$. Viewing A as an instance of SD, we get a set B (i.e. an instance of RT^1_2) such that every infinite subset of B and $\overline{B}$ (i.e. every solution to $\mathsf{RT}^1_2(B)$) computes an infinite subset of A (i.e. a solution to $\mathsf{SD}(A)$). However, since G is infinite, at least one of $G \cap B$ and $G \cap \overline{B}$ is also infinite and does not compute any infinite subset of A.

In our construction of A and G, we will use a corollary of the Galvin-Prikry theorem [7].

Theorem 7.3 (Galvin-Prikry theorem)**.** *For any Borel set $\mathcal{X} \subseteq \mathcal{P}(\mathbb{N})$, there is some infinite set $B \subseteq \mathbb{N}$ such that either all infinite subsets of B are in $\mathcal{X}$ or no infinite subsets of B are in $\mathcal{X}$.*

Corollary 7.4. *For any finite set k, Borel coloring $c\colon \mathcal{P}(\mathbb{N}) \to k$ and infinite set B, there is some infinite set $B' \subseteq B$ such that all infinite subsets of B' have the same color.*

Before proving the existence of A and G, we will try to motivate the construction and explain how the Galvin-Prikry theorem will be used. Our goal is to construct A and G such that for all Turing functionals Φ and all infinite subsets $C \subseteq G$, $\Phi(C)$ is not an infinite subset of A. Suppose that instead of having to handle all Turing functionals, we just had to handle a single Turing functional, Φ. By the Galvin-Prikry theorem, there is some infinite set B_0 such that either

(1) for all infinite subsets $C \subseteq B_0$, $\Phi(C, 0)\downarrow = 1$
(2) or for all infinite subsets $C \subseteq B_0$, $\Phi(C, 0) \neq 1$ (i.e. $\Phi(C, 0)$ either diverges or converges to something besides 1).

In the first case, we are done: we can take $G = B_0$ and make sure $0 \notin A$ (e.g. take $A = \mathbb{N} \setminus \{0\}$). The second case is more difficult. Taking $G = B_0$ is not enough because it does not give us much control over what is computed by infinite subsets of G—it just ensures that if C is an infinite subset of G then 0 is not an element of the set $\Phi(C)$.

However, requiring G to be a subset of B_0 does seem like *progress* towards our goal: if we could similarly ensure that for each $n \in \mathbb{N}$ and infinite set $C \subseteq G$, n is not an element of $\Phi(C)$, then we would be done (because this would imply that for each such C, $\Phi(C)$ is either not total or computes the empty set).

This suggests the following strategy: form a sequence of sets

$$B_0 \supseteq B_1 \supseteq B_2 \supseteq \ldots$$

where for each n, B_n is chosen using the Galvin-Prikry theorem so that either

(1) for all infinite subsets $C \subseteq B_n$, $\Phi(C, n)\downarrow = 1$
(2) or for all infinite subsets $C \subseteq B_0$, $\Phi(C, n) \neq 1$.

If the first case holds for some n, we can take $G = B_n$ and forbid n from being included in A. If the second case holds for every n, then we would like to take $G = \bigcap_{n\in\mathbb{N}} B_n$ since, as we noted above, this ensures that for each infinite $C \subseteq G$, $\Phi(C)$ is either not total or computes the empty set. However, there is one problem with this: $\bigcap_{n\in\mathbb{N}} B_n$ could be finite, or even empty.

To deal with this problem, we will construct G using a sequence of Mathias conditions, at each step restricting the reservoir in a manner similar to what we have just described, while also adding some elements to the stem to ensure G is infinite. We will now give the details of the proof. Because of the necessity of using Mathias conditions, of dealing with all Turing functionals (rather than just one), and of ensuring that A has high density, our proof is somewhat more elaborate than the sketch we have just given. However, the key ideas are the same.

Theorem 7.5. $\mathsf{SD} \not\leq_{soc} \mathsf{RT}^1_2$.

Proof. As explained above, we will show how to construct infinite sets A and G such that A has lower density at least $1/2$ and no infinite subset of G computes an infinite subset of A. To construct G, we will build a sequence of Mathias conditions

$$(\varnothing, \mathbb{N}) = (s_0, B_0) \geq (s_1, B_1) \geq (s_2, B_2) \geq \dots$$

and take $G = \bigcup_n s_n$. Along the way, we will forbid certain numbers from being included in A and then take A to be the set of all non-forbidden numbers. For convenience, we will ensure that each s_n has size exactly n.

For each e, we must satisfy the following requirement:

> **Requirement *e*:** for every infinite $C \subseteq G$, either there is some m such that $\Phi_e(C, m)\!\downarrow = 1$ and $m \notin A$ or for all but finitely many m, $\Phi_e(C, m) \neq 1$.

To satisfy requirement e, we will take some action at each step $n > e$ of the construction. Roughly speaking, on step n we will try to satisfy the requirement for all m in the interval $[2^{e+n+3}, 2^{e+n+4})$.

More precisely, to satisfy requirement e, we will ensure that for each step $n > e$ and each $t \subseteq s_n$, either

(1) there is some $m \in [2^{e+n+3}, 2^{e+n+4})$ such that for all infinite sets C compatible with (t, B_n), $\Phi(C, m)\!\downarrow = 1$
(2) or for all $m \in [2^{e+n+3}, 2^{e+n+4})$ and infinite sets C compatible with (t, B_n), $\Phi(C, m) \neq 1$.

If the first case holds, we will also pick one such m to forbid from A.

How to pick s_{n+1} and B_{n+1}. For each n, form s_{n+1} by picking an arbitrary element of B_n to add to s_n. Then pick B_{n+1} as follows.

For each $e < n+1$, $t \subseteq s_{n+1}$ and $m \in [2^{e+n+1+3}, 2^{e+n+1+4})$, define a coloring $c_{e,t,m} \colon \mathcal{P}(\mathbb{N}) \to \{0,1\}$ by

$$c_{e,t,m}(B) = \begin{cases} 1 & \text{if } \Phi_e(t \cup B, m)\!\downarrow = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Next, define a coloring c on $\mathcal{P}(\mathbb{N})$ by setting $c(B)$ to be the sequence

$$c(B) = \langle c_{e,t,m}(B) \mid e < n+1, t \subseteq s_{n+1}, m \in [2^{e+n+1+3}, 2^{e+n+1+4})\rangle.$$

Note that c has finite range and can easily be seen to be Borel. Thus by the corollary to the Galvin-Prikry theorem, we can choose B_{n+1} to be an infinite subset of B_n such that c assigns the same value to all infinite subsets of B_{n+1}. It is straightforward to check that B_{n+1} has the properties desired.

Finally, recall that we must forbid some elements from A. For each $e < n+1$ and $t \subseteq s_{n+1}$, if there is any m in the interval $[2^{e+n+1+3}, 2^{e+n+1+4})$ such that $c_{e,t,m}$ has constant value 1 on B_{n+1} then pick the least such m and forbid it from A. Also recall that at the end of the construction, we take A to consist of all numbers not forbidden from A.

This concludes the construction of G and A. All that remains now is to check that they have the necessary properties.

The set G satisfies all requirements. Fix any e and we will show that requirement e is satisfied. Suppose C is an infinite subset of G. We will show that either $\Phi(C)$ is not a subset of A or it is not infinite.

For each n, define $t_n = C \cap s_n$ and note that since $G \subseteq s_n \cup B_n$, C is compatible with (t_n, B_n). We know that for each $n > e$, one of two possibilities holds:

(1) there is some $m \in [2^{e+n+3}, 2^{e+n+4})$ such that m is forbidden from A and for all infinite sets D compatible with (t_n, B_n), $\Phi(D, m)\downarrow = 1$
(2) or for every $m \in [2^{e+n+3}, 2^{e+n+4})$ and every infinite set D compatible with (t_n, B_n), $\Phi(D, m) \neq 1$.

On the one hand, if the first possibility holds for any n then there is some m such that $\Phi(C, m)\downarrow = 1$ and m is forbidden from A, which implies that $\Phi(C)$ is not a subset of A.

On the other hand, if the second possibility always holds then for every $n > e$ and every m in the interval $[2^{e+n+3}, 2^{e+n+4})$, $\Phi(C, m) \neq 1$. Since the union of these intervals consists of all numbers greater than or equal to 2^{2e+4}, this shows that either $\Phi(C)$ is not total or it only contains numbers less than 2^{2e+4} and thus is not infinite.

The set A has high density. For each e, let A_e denote the set of numbers forbidden from A on behalf of requirement e. Note that $\overline{A} = \bigcup_e A_e$. We will show that each A_e never has density greater than $1/2^{e+2}$ and thus that their union never has density more than

$$\sum_{e \in \mathbb{N}} \frac{1}{2^{e+2}} = \frac{1}{2}.$$

This implies that $\overline{A}$ never has density more than $1/2$ and thus that A is $1/2$-dense. Note, by the way, that when we say that each A_e never has density greater than $1/2^{e+2}$, we mean that for each n, A_e is at most $1/2^{e+2}$-dense at n (in the sense of Section 2.2); it is not good enough for it to just have upper density at most $1/2^{e+2}$.

So fix e and we will argue that A_e never has density more than $1/2^{e+2}$. First consider a single interval of the form $[2^{e+n+3}, 2^{e+n+4})$ for $n > e$. The only time numbers from this interval will be added to A_e is on step n of the construction and on that step, at most one number will be added to A_e per subset of s_n. Since s_n has size exactly n, this means at most 2^n such numbers will be added. Thus we have

$$|A_e \cap [2^{e+n+3}, 2^{e+n+4})| \leq 2^n = \frac{1}{2^{e+3}} \cdot |[2^{e+n+3}, 2^{e+n+4})|.$$

In other words, in the interval $[2^{e+n+3}, 2^{e+n+4})$, A_e has density at most $1/2^{e+3}$.

From this fact it can easily be shown that at each power of 2, A_e has density at most $1/2^{e+3}$, i.e. for each $m > 0$,

$$|A_e \cap [2^m - 1]| \leq \frac{2^m}{2^{e+3}}.$$

It remains to check that the density of A_e is high enough in-between powers of 2. But this follows easily from what we have already established. If $2^m \leq k < 2^{m+1}$

then we have

$$\begin{aligned}|A_e \cap [k]| &\leq |A_e \cap [2^{m+1} - 1]| \\ &\leq \frac{2^{m+1}}{2^{e+3}} = \frac{2^m}{2^{e+2}} \leq \frac{k+1}{2^{e+2}}\end{aligned}$$

and thus A_e is at most $1/2^{e+2}$-dense at k, as promised. □

The proof of Theorem 7.5 actually yields a somewhat stronger result. Let $\mathsf{RT}^1_{<\infty}$ denote the problem in which an instance is a finite partition $A_1, \ldots, A_n$ of $\mathbb{N}$ and a solution is an infinite set B which is a subset of some A_i. Roughly speaking, $\mathsf{RT}^1_{<\infty}$ is the problem corresponding to Corollary 2.2 of Seetapun's theorem and thus the corollary below shows that there is no direct proof of our main theorem from Corollary 2.2 in the same sense that the theorem above showed there is no direct proof of our main theorem from Seetapun's theorem.

Corollary 7.6. $\mathsf{SD} \not\leq_{soc} \mathsf{RT}^1_{<\infty}$.

Proof. Let A and G be as in the proof of Theorem 7.5 and suppose for contradiction that $\mathsf{SD} \leq_{soc} \mathsf{RT}^1_{<\infty}$. Then, thinking of A as an instance of SD, we get a partition $B_1, \ldots, B_n$ of $\mathbb{N}$ such that for any B_i and any infinite subset $C \subseteq B_i$, C computes an infinite subset of A. However, since G is infinite, there must be some B_i such that $G \cap B_i$ is infinite. Since $G \cap B_i$ is an infinite subset of G, it does not compute any infinite subset of A, which gives a contradiction. □

References

[1] Laurent Bienvenu, Adam R Day, and Rupert Hölzl. From bi-immunity to absolute undecidability. *The Journal of Symbolic Logic*, 78(4):1218–1228, 2009.

[2] Peter A Cholak, Carl G Jockusch, and Theodore A Slaman. On the strength of ramsey's theorem for pairs. *The Journal of Symbolic Logic*, 66(1):1–55, 2001.

[3] JCE Dekker and John Myhill. Retraceable sets. *Canadian journal of mathematics*, 10:357–373, 1958.

[4] Rodney G Downey and Denis R Hirschfeldt. *Algorithmic randomness and complexity*. Springer Science & Business Media, 2010.

[5] Damir D. Dzhafarov. Cohesive avoidance and strong reductions. *Proc. Amer. Math. Soc.*, 143(2):869–876, 2015.

[6] Damir D Dzhafarov and Carl G Jockusch. Ramsey's theorem and cone avoidance. *The Journal of Symbolic Logic*, 74(2):557–578, 2009.

[7] Fred Galvin and Karel Prikry. Borel sets and ramsey's theorem. *The Journal of Symbolic Logic*, 38(2):193–198, 1973.

[8] Denis R Hirschfeldt. *Slicing the truth: On the computable and reverse mathematics of combinatorial principles*. World Scientific, 2015.

[9] Carl G Jockusch. Uniformly introreducible sets. *The Journal of Symbolic Logic*, 33(4):521–536, 1969.

[10] Jiayi Liu. RT^2_2 does not imply WKL_0. *J. Symb. Log.*, 77(2):609–620, 2012.

[11] Benoit Monin and Ludovic Patey. Π^0_1-encodability and omniscient reductions. *Notre Dame Journal of Formal Logic*, 60(1):1–12, 2019.

[12] Benoit Monin and Ludovic Patey. Pigeons do not jump high. *Advances in Mathematics*, 352:1066–1095, 2019.

[13] Benoit Monin and Ludovic Patey. SRT^2_2 does not imply RT^2_2 in ω-models. *Adv. Math.*, 389:32, 2021. Id/No 107903.

[14] Benoit Monin and Ludovic Patey. The weakness of the pigeonhole principle under hyperarithmetical reductions. *J. Math. Log.*, 21(3):41, 2021. Id/No 2150013.

[15] David Seetapun and Theodore A Slaman. On the strength of ramsey's theorem. *Notre Dame Journal of Formal Logic*, 36(4):570–582, 1995.

[16] Nikolai K. Vereshchagin. Kolmogorov complexity conditional to large integers. *Theoretical Computer Science*, 271(1-2):59–67, 2002. Kolmogorov complexity.