ENTROPIES REALIZABLE BY BLOCK GLUING
$\mathbb{Z}^d$ SHIFTS OF FINITE TYPE

RONNIE PAVLOV AND MICHAEL SCHRAUDNER

ABSTRACT. In [9], Hochman and Meyerovitch gave a complete characterization of the set of topological entropies of $\mathbb{Z}^d$ shifts of finite type (SFTs) via a recursion-theoretic criterion. However, the $\mathbb{Z}^d$ SFTs they construct in the proof are relatively degenerate and in particular lack any form of topological mixing, leaving open the question of which entropies can be realized within $\mathbb{Z}^d$ SFTs with (uniform) mixing properties. In this paper, we describe some progress on this question. We show that in order for $\alpha \in \mathbb{R}_0^+$ to be the topological entropy of a block gluing $\mathbb{Z}^2$ SFT, it cannot be too poorly computable; in fact it must be possible to compute approximations to α within arbitrary tolerance ϵ in time $2^{O(\epsilon^{-2})}$. As a partial converse, we present a new technique to realize a large class of computable real numbers as entropies of block gluing $\mathbb{Z}^d$ SFTs for any $d > 2$. Also as a corollary of our methods, we construct, for any $N > 1$, a block gluing $\mathbb{Z}^d$ SFT ($d > 2$) with entropy $\log N$ but without a full N-shift factor, strengthening previous work [6] by Boyle and the second author.

1. INTRODUCTION AND MAIN RESULTS

This paper studies the class of real numbers which appear as topological entropies of $\mathbb{Z}^d$ shifts of finite type (SFTs) in the presence of a uniform mixing property called "block gluing" which was introduced in [5] (see also Section 2.1).

Topological entropy is the most important numerical invariant of a topological dynamical system. For subshifts over finite alphabets, topological entropy is given by the exponential growth rate of (globally) admissible patterns for the subshift on larger and larger shapes. It is thus defined as a limit (for a formal definition, again see Section 2.1) which, in the case of a non-empty subshift, yields a non-negative real number. Since any non-negative real number is realizable as the entropy of some $\mathbb{Z}$ subshift, this naturally leads to the question of which numbers actually appear as entropies within certain subclasses of $\mathbb{Z}$ (or $\mathbb{Z}^d$) subshifts.

For the fundamental class of (mixing) $\mathbb{Z}$ shifts of finite type this question was completely answered by Lind [14] in 1984. Recall that any $\mathbb{Z}$ SFT $X = \mathsf{X}_A$ can be represented by a finite directed graph – therefore this class of symbolic systems is countable (up to topological conjugacy) and so can not realize all non-negative real numbers – whose adjacency matrix A contains all of the information about admissible patterns (words). (Specifically, the number of legal words of a given

length is just the sum of the entries of the corresponding power of A.) Applying the well-known Perron-Frobenius theorem immediately implies that the entropy has the simple closed form $h_{top}(\mathsf{X}_A) = \log \lambda_A$, where $\lambda_A \in \mathbb{R}^+$ is the Perron eigenvalue of the non-negative integer matrix A. In particular, this shows that only logarithms of certain algebraic integers can appear as entropies of general (or mixing) $\mathbb{Z}$ SFTs, and Lind showed that all of those numbers are in fact realizable:

Theorem 1.1 ([14])**.** *The class of topological entropies of $\mathbb{Z}$ shifts of finite type coincides with the class of non-negative rational multiples of logarithms of Perron numbers[1], and the class of topological entropies of mixing $\mathbb{Z}$ shifts of finite type coincides with the class of logarithms of Perron numbers.*

For $\mathbb{Z}^d$ subshifts, the situation is much more complicated: even for the fundamental (again countable) class of $\mathbb{Z}^d$ SFTs, in general there is neither a closed form to represent nor a general algorithm to compute the exact value of the topological entropy. In fact, only for very few isolated non-trivial examples are there methods to compute the exact value of the entropy [2, 12, 13], while for other even more basic examples, like the $\mathbb{Z}^2$ golden mean shift, only approximations to $h_{top}(X)$ are known [3, 7, 19]. Despite the intrinsic difficulties in finding a closed form for topological entropy, in 2007 Hochman and Meyerovitch [9] completely classified the family of numbers realizable as entropies of $\mathbb{Z}^d$ SFTs. Surprisingly, this characterization, rather than involving algebraic conditions as in the $\mathbb{Z}$ setting, is given entirely in recursion-theoretic terms.

Theorem 1.2 ([9])**.** *For any $d \geqslant 2$, the class of topological entropies of $\mathbb{Z}^d$ shifts of finite type coincides with the class of right-recursively enumerable non-negative real numbers.*

Here a real number $r \in \mathbb{R}$ is called "right-recursively enumerable" if there exists a Turing machine which, for any input $n \in \mathbb{N}$, computes a rational approximation $r_n \in \mathbb{Q}$ such that the sequence $(r_n)_{n\in\mathbb{N}}$ is decreasing and converges to r. It is easy to check that the class of right-recursively enumerable numbers strictly contains the class of algebraic numbers as well as the class of computable numbers. (A real number $r \in \mathbb{R}$ is called "computable" if there exists a Turing machine which, for any input $n \in \mathbb{N}$, computes a rational approximation $r_n \in \mathbb{Q}$ such that $|r - r_n| < \frac{1}{n}$.)

The construction technique used by Hochman-Meyerovitch to realize an arbitrary right-recursively enumerable number as the entropy of a $\mathbb{Z}^d$ SFT however is very rigid. Unsurprisingly, it involves simulation of a Turing machine within the SFT itself, and the deterministic nature of the Turing machine precludes any form of topological mixing. Their classification therefore does not extend to any subclass of mixing $\mathbb{Z}^d$ SFTs as in Lind's result. In addition, the Hochman-Meyerovitch construction involves first constructing a zero-entropy $\mathbb{Z}^d$ SFT in which certain symbols appear with frequencies only up to a certain value, and then creating the desired entropy by introducing independent copies of those specific symbols. Hence the resulting $\mathbb{Z}^d$ SFT always has this initial $\mathbb{Z}^d$ SFT as a non-trivial zero-entropy factor, thus precluding in particular uniform mixing conditions such as block gluing (see Theorem B.2 from [5]).

This raises the question of whether general right-recursively enumerable numbers, for which no estimates on the rate of convergence of the sequence $(r_n)_{n\in\mathbb{N}}$ to

---

[1]A Perron number is a real algebraic integer larger than or equal to 1 which in modulus exceeds all its algebraic conjugates.

r are available, can appear as entropies only for degenerate $\mathbb{Z}^d$ SFTs or whether poorly computable numbers persist as entropies within subclasses of (uniformly) mixing $\mathbb{Z}^d$ SFTs. It was already shown in [9] that in the presence of the strongest uniform mixing condition, strong irreducibility, the entropies of $\mathbb{Z}^d$ SFTs must satisfy the strictly stronger recursion-theoretic condition of computability.

Hochman-Meyerovitch then asked the question of which computable numbers appear as entropies for strongly irreducible SFTs, but neither necessary nor sufficient conditions for realizability of a computable number as the entropy of such a $\mathbb{Z}^d$ SFT are known. We will show in this paper that any uniform mixing property necessarily forces an upper bound on the computation time of the rational approximations $(r_n)_{n \in \mathbb{N}}$ (i.e. very badly-computable numbers cannot be realized as the entropy of any such SFT), and that within the remaining class of computable numbers, many can actually be realized as entropies within the block gluing subclass.

Our main results are stated in the remainder of this section. The first theorem shows that not all computable non-negative real numbers can appear as topological entropies of block gluing, let alone strongly irreducible, $\mathbb{Z}^2$ SFTs.

Definition 1.3. Let $(t_n \in \mathbb{N})_{n \in \mathbb{N}}$ be a non-decreasing sequence of natural numbers. A real number $r \in \mathbb{R}$ is **computable with rate** $(t_n)_{n \in \mathbb{N}}$ if there exists a deterministic Turing machine which, for any $n \in \mathbb{N}$, calculates in at most t_n steps a rational approximation $r_n \in \mathbb{Q}$ of r such that $|r - r_n| \leqslant \frac{1}{n}$.

Theorem 1.4. *For any block gluing $\mathbb{Z}^2$ shift of finite type there exists some constant $C \in \mathbb{R}$ such that its topological entropy is computable with rate $\left(2^{C \cdot n^2}\right)_{n \in \mathbb{N}}$.*

Unfortunately our proof – given in Section 3 – only works for $d = 2$. The argument however does extend to a version for $\mathbb{Z}^d$ SFTs with $d \geqslant 3$, but it requires additional hypotheses, namely the uniform filling property and existence of a point with finite orbit, to imply computability of the entropy with rate $\left(2^{C \cdot n^d}\right)_{n \in \mathbb{N}}$.

Our next result gives a sufficient condition for realizability; we will show that any non-negative real number which is well-approximable in a certain sense (given by a condition on the computation time of its continued fraction expansion, defined below) is indeed realizable as the entropy of some block gluing $\mathbb{Z}^d$ SFT.

The concept of a Turing machine (TM) – whose connection to multidimensional symbolic dynamics was established by the work of Berger [4] – has recently seen several important applications [8, 9, 1]. In these works, the relevant property of the TM is its ability to recursively enumerate an effective set of forbidden words by only locally modifying the content of a tape in the neighborhood of the read-write head's current position. This makes it possible to embed such computations row-by-row into specifically marked regions, containing consecutively all instantaneous descriptions of the TM's discrete time evolution, within points of $\mathbb{Z}^2$ SFTs. However, in such simulations, the capabilities of standard TMs do not exploit all of the possibilities given by local rules. Hence, first we define an extension of the concept of a TM which seems better adapted to use within $\mathbb{Z}^d$ SFTs and captures exactly the computational capabilities of local rules. We call this new concept an **accelerated Turing machine** (for a formal definition, see Section 4) due to the fact that it can compute the same class of objects as a standard TM, but usually in considerably less time.

Definition 1.5. A real number $\alpha \in \mathbb{R}$ **satisfies computability condition (C)** if it allows a representation $\alpha = \alpha' \cdot \log M$ for a natural number $1 < M \in \mathbb{N}$ and some $\alpha' \in \mathbb{R}$ which is either rational or which has an infinite continued fraction expansion $\alpha' = [a_0; a_1, a_2, a_3, \ldots]$ with the following property: for the sequence $(t_n \in \mathbb{N})_{n \in \mathbb{N}_0}$ recursively defined by $t_0 := 1$, $t_1 := a_1$ and $t_n := a_n \cdot t_{n-1} + t_{n-2}$ for all $n \geqslant 2$, there exists an **(accelerated) Turing machine**, which recursively enumerates the sequence of partial quotients $(a_n)_{n \in \mathbb{N}}$ in such a way that for every $N \in \mathbb{N}$, a (binary) representation of all of the first N partial quotients $a_1, a_2, a_3, \ldots, a_N$ has been produced within the first $a_N \cdot t_{N-1}$ steps.

Before we state our second main theorem (Theorem 1.7) we give some families of real numbers that satisfy computability condition **(C)**:

Examples 1.6. All (non-negative) real numbers $\alpha = \alpha' \cdot \log M$ $(1 < M \in \mathbb{N})$ with
- $\alpha' \in \mathbb{Q}$ an arbitrary rational number or
- $\alpha' \in \mathbb{R} \backslash \mathbb{Q}$ an irrational number with an
  (1) **eventually constant** continued fraction expansion (e.g. numbers like $\frac{\sqrt{5}-1}{2} = [0; 1, 1, 1, 1, \ldots]$ and $\frac{\sqrt{2}}{2} = [0; 1, 2, 2, 2, \ldots]$ etc.)
  (2) **eventually periodic** continued fraction expansion (i.e. all irrational quadratic algebraic numbers)
  (3) **eventually affine** continued fraction expansion, meaning that there exist constants $C \in \mathbb{N}$ and $D \in \mathbb{Z}$ and a starting point $n' \in \mathbb{N}$ such that for all $n \in \mathbb{N}$, $n \geqslant n'$, the partial quotients are of the form $a_n = C \cdot n + D$
  (4) **eventually affine-periodic** continued fraction expansion, meaning that there exist constants $C_0, C_1, \ldots, C_{p-1} \in \mathbb{N}$ and $D_0, D_1, \ldots, D_{p-1} \in \mathbb{Z}$ $(p \in \mathbb{N})$ and a starting point $n' \in \mathbb{N}$ such that for all $n \in \mathbb{N}$, $n \geqslant n'$, the partial quotients are of the form $a_n = C_{n \ (\mathrm{mod}\ p)} \cdot n + D_{n \ (\mathrm{mod}\ p)}$

satisfy computability condition **(C)**.

Note that classes (1) to (4) form a strictly increasing diamond lattice and that classes (3) and (4) in particular include well known (transcendental) numbers like

$$
\begin{aligned}
\tanh\left(\tfrac{1}{k}\right) &= [0; k, 3k, 5k, 7k, \ldots] \quad \text{with } a_n := 2k \cdot n - k \\
\tan(1) &= [1; 1, 1, 3, 1, 5, 1, 7, 1, \ldots] \\
\tan\left(\tfrac{1}{k}\right) &= [0; k-1, 1, 3k-2, 1, 5k-2, 1, 7k-2, 1, \ldots] \quad (k > 1) \\
\exp(1) &= [2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, \ldots] \quad \text{or} \\
\exp\left(\tfrac{1}{k}\right) &= [1; k-1, 1, 1, 3k-1, 1, 1, 5k-1, 1, 1, 7k-1, 1, 1, \ldots] \quad (k > 1)
\end{aligned}
$$

for arbitrary $k \in \mathbb{N}$.

We give a short explanation why numbers in class (3) satisfy the computability condition **(C)** in Section 4. The argument for (the more general) class (4) is similar.

We remark that many other non-negative real numbers satisfy computability condition **(C)** as well, e.g. numbers α' with any other fast enough computable "regularity" in their partial quotients like $a_n := \begin{cases} 2 & \text{if } n \text{ is a perfect square} \\ 1 & \text{otherwise} \end{cases}$ but also – as we will prove in Section 6 – numbers like $\alpha' := \frac{\log L}{\log M}$ with $1 < L < M \in \mathbb{N}$ coprime, for which no regularity in their continued fraction expansion is known.

Theorem 1.7. *Suppose $d \geqslant 3$. For any non-negative real number $\alpha \in \mathbb{R}_0^+$ which satisfies computability condition (C), there exists a block gluing $\mathbb{Z}^d$ shift of finite type X with topological entropy $\mathrm{h}_{\mathrm{top}}(X) = \alpha$.*

As an application of the construction used to prove Theorem 1.7 (given in Section 5), we are also able to prove a new result about non-existence of certain entropy-preserving factor maps. In a previous work [6] by Mike Boyle and the second author, the following result was proven:

Theorem 1.8 ([6, Theorem 2.1]). *Suppose $d \geqslant 2$ and $1 < L \in \mathbb{N}$. Then there exists a $\mathbb{Z}^d$ shift of finite type X with topological entropy $\mathrm{h}_{\mathrm{top}}(X) = \log L$ which does not factor onto the full $\mathbb{Z}^d$ shift on L symbols.*

Adapting the argument from [6] to our newly constructed family of block gluing $\mathbb{Z}^d$ SFTs, we are able to show that the same result still holds true (for $d \geqslant 3$) even if we require X to be block gluing.

Theorem 1.9. *Suppose $d \geqslant 3$ and $1 < L \in \mathbb{N}$. Then there exists a block gluing $\mathbb{Z}^d$ shift of finite type X with topological entropy $\mathrm{h}_{\mathrm{top}}(X) = \log L$ which does not factor onto the full $\mathbb{Z}^d$ shift on L symbols.*

We point out that this new theorem stands in stark contrast to another result of the authors (again with Mike Boyle) contained in [5] which, in the presence of the block gluing property, guarantees the existence of strictly lower-entropy full shift factors even for general (non-SFT) shifts.

Theorem 1.10 ([5, Theorem 3.2]). *Suppose $d \geqslant 1$ and $L \in \mathbb{N}$. Any block gluing $\mathbb{Z}^d$ shift X with topological entropy $\mathrm{h}_{\mathrm{top}}(X) > \log L$ factors onto the full $\mathbb{Z}^d$ shift on L symbols.*

Theorems 1.9 and 1.10 thus demonstrate a conceptual difference between entropy-preserving and entropy-decreasing factor maps in the block gluing regime.

The remainder of this paper is organized as follows: In Section 2, we recall some basic definitions from (multi-dimensional) symbolic dynamics, as well as some facts about continued fraction expansions and Sturmian sequences necessary for the construction proving Theorems 1.7 and 1.9. In Section 3, we prove the necessity of the rate-of-computability hypothesis from Theorem 1.4. **Accelerated Turing machines** are formally introduced in Section 4. Section 5 contains a detailed description of the construction of our family of block-gluing $\mathbb{Z}^3$ SFTs realizing certain topological entropies as stated in Theorem 1.7. This construction has two distinct steps: In the first, we construct a $\mathbb{Z}^3$ SFT with the desired topological entropy having a property we call **upgradability**. The second step is given by Theorem 5.11, which demonstrates a general procedure to add symbols to any upgradable $\mathbb{Z}^d$ SFT ($d \geqslant 2$) which make it block gluing without increasing its entropy; a result which might be of independent interest. Finally, Section 6 describes the modifications of the argument from [6] necessary to prove Theorem 1.9.

2. Preliminaries

2.1. **Symbolic dynamics.** We assume a basic familiarity with symbolic dynamics (for additional background refer to [15]), and so only recall a few definitions and notations.

For $d \in \mathbb{N}$ let us consider $\mathbb{Z}^d$ as a metric space w.r.t. the maximum metric δ_∞, i.e. for every pair $\vec{u}, \vec{w} \in \mathbb{Z}^d$ we define $\delta_\infty(\vec{u}, \vec{w}) := \|\vec{u} - \vec{w}\|_\infty = \max_{1 \leqslant k \leqslant d} |\vec{u}_k - \vec{w}_k|$. We can also extend δ_∞ in the natural way to obtain a non-negative symmetric function $\delta_\infty(U, W) := \min_{\vec{u} \in U, \vec{w} \in W} \delta_\infty(\vec{u}, \vec{w})$ representing the separation between non-empty subsets $U, W \subseteq \mathbb{Z}^d$.

For $\vec{u}, \vec{w} \in \mathbb{Z}^d$ we define the partial order given by coordinatewise dominance: $\vec{u} \preceq \vec{w}$ denotes that $\vec{u}_k \leqslant \vec{w}_k$ for every $1 \leqslant k \leqslant d$. $B := \{ \vec{v} \in \mathbb{Z}^d \mid \vec{u} \preceq \vec{v} \preceq \vec{w} \}$ will then be called a non-empty (finite) **(rectangular/cuboid) solid block** in $\mathbb{Z}^d$ defined by $\vec{u} \preceq \vec{w} \in \mathbb{Z}^d$.

As we are mainly dealing with the case $\mathbb{Z}^2$ and $\mathbb{Z}^3$, we will refer to the (cardinal) $\vec{e}_1$, $\vec{e}_2$ directions as horizontal and vertical and an $\vec{e}_1\vec{e}_2$-cross section of $\mathbb{Z}^3$ will be a set $\mathbb{Z}^2 \times \{m\} := \{ \vec{u} \in \mathbb{Z}^3 \mid \vec{u}_3 = m \}$ for some fixed $m \in \mathbb{Z}$.

Every finite (discrete) alphabet $\mathcal{A}$ gives rise to a d-**dimensional full shift** $\mathcal{A}^{\mathbb{Z}^d}$ for any $d \in \mathbb{N}$, and when equipped with the product topology, this compact space supports a natural expansive and continuous $\mathbb{Z}^d$ (shift) action $\sigma : \mathbb{Z}^d \times \mathcal{A}^{\mathbb{Z}^d} \to \mathcal{A}^{\mathbb{Z}^d}$ given by translations as $(\sigma^{\vec{u}}(x))_{\vec{v}} = (\sigma(\vec{u}, x))_{\vec{v}} := x_{\vec{u}+\vec{v}}$ for all $\vec{u}, \vec{v} \in \mathbb{Z}^d$, $x \in \mathcal{A}^{\mathbb{Z}^d}$.

Any closed shift-invariant subset of $\mathcal{A}^{\mathbb{Z}^d}$ is called a $\mathbb{Z}^d$ **(sub)shift**, and a **subsystem** $Y \subseteq X$ of a $\mathbb{Z}^d$ subshift X is itself a closed shift-invariant subset of X, together with the restriction $\sigma|_{\mathbb{Z}^d \times Y}$ of the $\mathbb{Z}^d$ shift action to this set.

Let $\mathcal{A}^{*,d} := \bigcup_{F \subsetneq \mathbb{Z}^d \text{ finite}} \mathcal{A}^F$ denote the countable set of all patterns made by assigning letters of $\mathcal{A}$ on finite subsets of $\mathbb{Z}^d$ and let $\mathcal{A}^* := \mathcal{A}^{*,1}$. Every $\mathbb{Z}^d$ subshift on $\mathcal{A}$ can also be defined by specifying a set of **forbidden patterns** $\mathcal{F} \subseteq \mathcal{A}^{*,d}$ and defining $\mathsf{X}(\mathcal{F}) := \{ x \in \mathcal{A}^{\mathbb{Z}^d} \mid \forall F \subsetneq \mathbb{Z}^d \text{ finite} : x|_F \notin \mathcal{F} \}$ to be the subshift of all points of the full $\mathbb{Z}^d$ shift on $\mathcal{A}$ which do not contain any patterns from $\mathcal{F}$. If $X = \mathsf{X}(\mathcal{F})$ for some finite $\mathcal{F}$, then X is called a (**d-dimensional**) **shift of finite type** ($\mathbb{Z}^d$ SFT), and we may assume without loss of generality that $\mathcal{F} \subseteq \mathcal{A}^F$ is a set of excluded patterns (local rules) for a single finite non-empty shape $F \subsetneq \mathbb{Z}^d$.

Given $X = \mathsf{X}(\mathcal{F}) \subseteq \mathcal{A}^{\mathbb{Z}^d}$, a (finite) pattern $P \in \mathcal{A}^{*,d}$ is called **locally admissible** in X if it contains no element from $\mathcal{F}$ as a subpattern, whereas P is called **globally admissible** in X if it actually shows up in a point of X, i.e. if P can be extended to a valid configuration on all of $\mathbb{Z}^d$ which does not contain any element from $\mathcal{F}$. The set of locally admissible patterns $\mathcal{L}^{\mathrm{loc}}(\mathsf{X}(\mathcal{F})) := \{ P \in \mathcal{A}^{*,d} \mid \forall Q \sqsubseteq P : Q \notin \mathcal{F} \}$ in general depends on the choice of $\mathcal{F}$ and strictly contains the set of globally admissible patterns $\mathcal{L}(X) := \{ x|_F \mid x \in X \land F \subsetneq \mathbb{Z}^d \text{ finite} \}$; the latter is also known as the **language** of X. Each of these sets can be written as the union over all finite shapes $F \subsetneq \mathbb{Z}^d$ of all sets of (locally/globally) admissible patterns with this shape F; the latter sets are denoted by $\mathcal{L}_F^{\mathrm{loc}}(\mathsf{X}(\mathcal{F})) := \{ P \in \mathcal{A}^F \mid \forall Q \sqsubseteq P : Q \notin \mathcal{F} \}$ resp. $\mathcal{L}_F(X) := \{ x|_F \mid x \in X \} \subsetneq \mathcal{L}(X)$. When the shape is a rectangular (cuboid) solid block in $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) with side-lengths $i, j \in \mathbb{N}$ (resp. $k, l, m \in \mathbb{N}$), we will also use $\mathcal{L}_{i,j}^{\mathrm{loc}}(X)$ (resp. $\mathcal{L}_{k,l,m}^{\mathrm{loc}}(X)$) and $\mathcal{L}_{i,j}(X)$ (resp. $\mathcal{L}_{k,l,m}(X)$) to denote these sets.

One important invariant associated with a $\mathbb{Z}^d$ subshift X is its **topological entropy**, a non-negative real number measuring the exponential growth rate of the number of globally admissible patterns on $C_l := \{ \vec{u} \in \mathbb{Z}^d \mid \| \vec{u} \|_\infty \leqslant l \}$, defined as

$$ \mathrm{h}_{\mathrm{top}}(X) := \lim_{l \to \infty} \frac{\log |\mathcal{L}_{C_l}(X)|}{|C_l|} \ . $$

A surjective continuous map between two $\mathbb{Z}^d$ subshifts commuting with the respective shift actions is called a **(topological) factor map** and the image of a $\mathbb{Z}^d$ subshift X under such map is referred to as a **factor** of X. Note that a factor map $\phi : X \to Y$ between $\mathbb{Z}^d$ subshifts can never increase topological entropy. In the case where $\mathrm{h}_{\mathrm{top}}(X) = \mathrm{h}_{\mathrm{top}}(Y)$ we say ϕ is **entropy-preserving**, and in the remaining case, i.e. when $\mathrm{h}_{\mathrm{top}}(X) > \mathrm{h}_{\mathrm{top}}(Y)$, ϕ is called **entropy-decreasing**.

Another highly important concept in the theory of $\mathbb{Z}^d$ subshifts (for $d > 1$) is the notion of topological mixing with a uniform distance, usually called **uniform mixing**. We give a few examples of uniform mixing conditions here.

Definition 2.1. A $\mathbb{Z}^d$ subshift X

(1) is called **block gluing with gap size** $g \in \mathbb{N}_0$ if for any pair of two solid blocks $B_1, B_2 \subsetneq \mathbb{Z}^d$ with separation $\delta_\infty(B_1, B_2) > g$ and any pair of valid points $y, z \in X$ there exists a valid point $x \in X$ such that $x|_{B_1} = y|_{B_1}$ and $x|_{B_2} = z|_{B_2}$.

(2) has the **uniform filling property (UFP) with filling length** $l \in \mathbb{N}_0$ if for any solid block $B \subsetneq \mathbb{Z}^d$ and any pair of valid points $y, z \in X$ there exists a valid point $x \in X$ with $x|_B = y|_B$ and $x|_{\mathbb{Z}^d \setminus (B + C_l)} = z|_{\mathbb{Z}^d \setminus (B + C_l)}$ for $C_l := \{\vec{u} \in \mathbb{Z}^d \mid \|\vec{u}\|_\infty \leqslant l\}$ as above.

(3) is called **strongly irreducible with gap size** $g \in \mathbb{N}_0$ if for any pair of non-empty (disjoint) finite subsets $U, W \subsetneq \mathbb{Z}^d$ with separation $\delta_\infty(U, W) > g$ and any pair of valid points $y, z \in X$ there exists a valid point $x \in X$ such that $x|_U = y|_U$ and $x|_W = z|_W$.

A $\mathbb{Z}^d$ subshift X is called **block gluing** (resp. **UFP**, etc.) if it is block gluing at gap g (resp. has the UFP with filling length g, etc.) for some $g \in \mathbb{N}_0$.

We mention that strong irreducibility clearly implies the uniform filling property, which in turn implies block gluingness (see the Appendices of [5], which provide a more detailed background on those uniform mixing conditions).

2.2. **Continued fractions.** We will give only the briefest of introductions to continued fractions; for more information, see the book of Hardy and Wright [11].

Definition 2.2 ([11, Chapter X]). Let $\alpha' \in \mathbb{R} \setminus \mathbb{Q}$ be an irrational real number. The **infinite continued fraction expansion** of α' is given as

$$\alpha' = [a_0; a_1, a_2, a_3, \ldots] := a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

where $a_0 \in \mathbb{Z}$ and $a_n \in \mathbb{N}$ for all $n \in \mathbb{N}$.

We remark that the continued fraction expansion of an irrational α' is unique, i.e. the value a_0 and the sequence $(a_n)_{n \in \mathbb{N}}$ of **partial quotients** is completely determined by the value of α'. For reasons which will become clear in Section 5, we are mostly interested in irrational numbers $\alpha' \in [0, 1] \setminus \mathbb{Q}$ for which we get an infinite expansion $\alpha' = [0; a_1, a_2, a_3, \ldots]$ starting with $a_0 := 0$.

Truncating the continued fraction expansion after a finite number of steps yields a sequence $(\frac{e_n}{f_n})_{n \in \mathbb{N}_0}$ of rational numbers called the **approximants** to α'; the nth approximant is thus defined as

$$\frac{e_n}{f_n} := a_0 + \cfrac{1}{a_1 + \cfrac{\ddots}{a_{n-1} + \frac{1}{a_n}}} \in \mathbb{Q} \ .$$

The integers e_n, f_n with $n \geqslant 0$, can also be defined recursively by

(i) $\quad \begin{aligned} &e_{-1} := 1, &&e_0 := a_0, &&e_n := a_n \cdot e_{n-1} + e_{n-2} \\ &f_{-1} := 0, &&f_0 := 1, &&f_n := a_n \cdot f_{n-1} + f_{n-2} \end{aligned} \qquad \forall n \in \mathbb{N} \ .$

The following lemma and its corollary will be useful in Section 6.

Lemma 2.3. *For any $\beta \in \mathbb{R}^+$ and any $n \in \mathbb{N}$,*

$$a_0 + \cfrac{1}{a_1 + \cfrac{\ddots}{a_{n-1} + \frac{1}{\beta}}} = \frac{\beta e_{n-1} + e_{n-2}}{\beta f_{n-1} + f_{n-2}} \ .$$

Proof. We use induction on the depth of the truncated continued fraction expansion. For $n = 1$, using the initial values in (i), the claim simply becomes $a_0 + \frac{1}{\beta} = \frac{\beta e_0 + e_{-1}}{\beta f_0 + f_{-1}} = \frac{\beta a_0 + 1}{\beta}$, which is obviously true. Now assume the claim has been established for depth $n = N$ and define $\beta' := a_N + \frac{1}{\beta} \in \mathbb{R}^+$. Again using the recursion (i), this immediately yields the desired equality for $n = N + 1$:

$$a_0 + \cfrac{1}{a_1 + \cfrac{\ddots}{a_N + \frac{1}{\beta}}} = a_0 + \cfrac{1}{a_1 + \cfrac{\ddots}{a_{N-1} + \frac{1}{\beta'}}} = \frac{\beta' e_{N-1} + e_{N-2}}{\beta' f_{N-1} + f_{N-2}} =$$

$$= \frac{\frac{1}{\beta} e_{N-1} + a_N e_{N-1} + e_{N-2}}{\frac{1}{\beta} f_{N-1} + a_N f_{N-1} + f_{N-2}} \overset{\text{(i)}}{=} \frac{\frac{1}{\beta} e_{N-1} + e_N}{\frac{1}{\beta} f_{N-1} + f_N} = \frac{\beta e_N + e_{N-1}}{\beta f_N + f_{N-1}},$$

finishing the proof. $\qquad\square$

As a corollary, we have the following alternate characterization of the partial quotients a_n ($n \in \mathbb{N}$) in terms of α' and the previous two approximants.

Corollary 2.4. *For every $n \in \mathbb{N}$ the nth partial quotient a_n is the unique integer for which α' is between $\frac{a_n e_{n-1} + e_{n-2}}{a_n f_{n-1} + f_{n-2}}$ and $\frac{(a_n+1) e_{n-1} + e_{n-2}}{(a_n+1) f_{n-1} + f_{n-2}}$.*

Proof. For $\beta := [a_n; a_{n+1}, a_{n+2}, \ldots] \in \mathbb{R}^+$, by Lemma 2.3 $\alpha' = \frac{\beta e_{n-1} + e_{n-2}}{\beta f_{n-1} + f_{n-2}}$. The corollary now follows directly by noting that $\beta \in (a_n, a_n + 1)$ and that the function $g(x) := \frac{x e_{n-1} + e_{n-2}}{x f_{n-1} + f_{n-2}}$ is strictly monotone in x. $\qquad\square$

2.3. **Sturmian sequences.** The family of Sturmian sequences is defined as the subset of one-sided infinite binary sequences over the alphabet $\{0, 1\}$ which satisfy any (both) of the equivalent conditions stated in the following theorem.

Theorem 2.5 ([16, Chapter 2]). *Let $y = (y_i)_{i \in \mathbb{N}_0} \in \{0,1\}^{\mathbb{N}_0}$ be an infinite sequence of 0s and 1s. The following are equivalent:*

(1) *y is the sequence obtained from coding the (forward) orbit of a point $x \in \mathbb{R}/\mathbb{Z}$ in the circle under the action of an irrational rotation $R_{\alpha'} : \mathbb{R}/\mathbb{Z} \to \mathbb{R}/\mathbb{Z}$, $x \mapsto x + \alpha' \pmod 1$ by some fixed irrational angle $\alpha' \in [0,1] \backslash \mathbb{Q}$ where the coding $y_i := \begin{cases} 0 & \text{if } R_{\alpha'}^i(x) \in [0, 1 - \alpha') \\ 1 & \text{if } R_{\alpha'}^i(x) \in [1 - \alpha', 1) \end{cases}$ for every $i \in \mathbb{N}_0$ takes place with respect to the intervals $I_0 := [0, 1 - \alpha')$ and $I_1 := [1 - \alpha', 1)$.*

(2) *y is aperiodic (i.e. $\forall p \in \mathbb{N} \, \exists i \in \mathbb{N}_0 : \, y_i \neq y_{i+p}$), balanced (i.e. $\forall u, v \sqsubseteq y$ finite subwords of y with $|u| = |v|$: $|\#_1(u) - \#_1(v)| \leqslant 1$)[2], and has limiting frequency of ones $\lim_{i \to \infty} \frac{\#_1(y_{[0, i-1]})}{i}$ equal to α'.*

For any given $\alpha' \in [0, 1] \backslash \mathbb{Q}$, there is a particular Sturmian sequence $c_{\alpha'} \in \{0,1\}^{\mathbb{N}_0}$ with limiting frequency of ones α', called the characteristic sequence for α'. This sequence $c_{\alpha'}$ has the following useful algorithmic description:

---

[2]Here $\#_1(u) := |\{i \in \mathbb{N} \mid u_i = 1\}|$ denotes the number of 1 symbols seen in the word $u \in \mathcal{A}^*$.

Facts 2.6 ([16, Chapter 2]). Let $y \in \{0,1\}^{\mathbb{N}_0}$ and let $\alpha' \in [0,1]\backslash\mathbb{Q}$.

(1) A Sturmian sequence y (of frequency α') is called **characteristic** if and only if $0y$ and $1y$ are both Sturmian sequences. Such $y = c_{\alpha'}$ corresponds to the coding of the (forward) orbit of the point α' under the action of $R_{\alpha'}$, i.e. $(c_{\alpha'})_i := \begin{cases} 0 & \text{if } R_{\alpha'}^i(\alpha') \in I_0 \\ 1 & \text{if } R_{\alpha'}^i(\alpha') \in I_1 \end{cases}$ for all $i \in \mathbb{N}_0$.

(2) If α' has continued fraction expansion $\alpha' = [0; a_1, a_2, a_3, \ldots]$ then there is a recursively defined sequence of finite words $(w_n \in \{0,1\}^*)_{n\in\mathbb{N}_0}$ with[3]

$$w_0 := 0 \,, \quad w_1 := 0^{a_1-1}\,1 \,, \quad w_n := w_{n-1}{}^{a_n}\,w_{n-2} \quad \forall\, n \geqslant 2$$

such that $c_{\alpha'} = \lim_{n\to\infty} w_n$ (in fact, $(c_{\alpha'})_{[0,|w_n|-1]} = w_n$ for all $n \in \mathbb{N}$).

3. A NECESSARY CONDITION – PROOF OF THEOREM 1.4

Suppose that $X \subseteq \mathcal{A}^{\mathbb{Z}^2}$ is a block gluing $\mathbb{Z}^2$ SFT with alphabet $\mathcal{A}$. Since it is not hard to check that any higher-block recoding of a block gluing SFT is still block gluing (with increased gap size), we may without loss of generality assume X to be nearest-neighbor. Define $g \in \mathbb{N}_0$ to be a gap size for which X is block gluing. We begin with a lemma relating local and global admissibility of patterns for X:

Lemma 3.1. *Let $X \subseteq \mathcal{A}^{\mathbb{Z}^2}$ be a block gluing nearest neighbor $\mathbb{Z}^2$ shift of finite type with gap size $g \in \mathbb{N}_0$ and let $m \in \mathbb{N}$. An $m \times m$ square pattern $P \in \mathcal{A}^{m \times m}$ on $\mathcal{A}$ is in $\mathcal{L}(X)$ if and only if there exists a rectangular locally admissible pattern $Q \in \mathcal{L}_{i,j}^{\mathrm{loc}}(X)$ with $i \leqslant |\mathcal{A}|^{2g+1}(g+m)+1$, $j \leqslant 2g+m+2$, which has identical left and right edges and identical top and bottom edges, and which contains P as a subpattern.*

Proof. "$\Longleftarrow$": We first prove the "if" direction. Suppose that P is a subpattern of a pattern $Q \in \mathcal{L}_{i,j}^{\mathrm{loc}}(X)$ as described in the lemma. Since X is a nearest neighbor shift of finite type, Q can clearly be used to (periodically) tile the plane yielding a valid point of X. But then P is a subpattern of this point, and so $P \in \mathcal{L}(X)$.

"$\Longrightarrow$": We now prove the "only if" direction. Let $K := |\mathcal{A}|^{2g+1}$. Choose any globally admissible square pattern $P \in \mathcal{L}_{m,m}(X)$ and any $R \in \mathcal{L}_{(K+1)(g+m),1}(X)$ of width $(|\mathcal{A}|^{2g+1}+1)(g+m)$ and height 1. By block gluing of X, there exists a still globally admissible rectangular pattern $Q' \in \mathcal{L}_{(K+1)(g+m),2g+m+2}(X)$ such that its top and bottom edge equal R, i.e.

$$Q'|_{[1,(K+1)(g+m)]\times\{1\}} = Q'|_{[1,(K+1)(g+m)]\times\{2g+m+2\}} = R$$

and, for any $0 \leqslant k \leqslant K$, Q' contains a copy of P at

$$Q'|_{[1+k(g+m),m+k(g+m)]\times[g+2,g+m+1]} = P \,.$$

For $0 \leqslant k \leqslant K$ consider the subpatterns $Q'_k := Q'|_{\{1+k(g+m)\}\times[1,2g+m+2]}$ which appear as equispaced columns of height $2g+m+2$ and width 1 inside Q' (see Figure 1). All of them contain as their middle part $Q'_k|_{\{1\}\times[g+2,g+m+1]}$ the leftmost column of P which is surrounded above and below by an arbitrary pattern of symbols from $\mathcal{A}$ of size $1 \times (g+1)$ where the top and bottom symbol of Q'_k have to coincide. Note that there are $|\mathcal{A}|^{2g+1}+1$ of these columns Q'_k, which have only $2g+1$ non-forced sites, so by the pigeonhole principle, there must exist two indices $0 \leqslant k <$

---

[3]Note that the sequence $(t_n \in \mathbb{N})_{n\in\mathbb{N}_0}$ used in Definition 1.5 satisfies $t_n = |w_n|$ for all $n \in \mathbb{N}_0$.
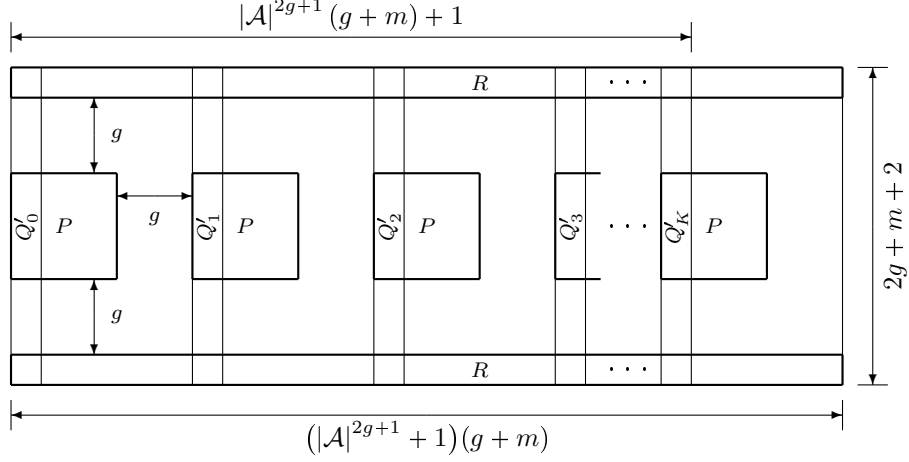
FIGURE 1. Q' and its subpatterns P, R, and Q'_k $(0 \leqslant k \leqslant K)$.

$k' \leqslant K$ such that $Q'_k = Q'_{k'}$. Then taking $Q := Q'|_{[1+k(g+m),1+k'(g+m)] \times [1,2g+m+2]}$ completes the proof. □

The following lemma is standard; see [18] for a proof. (The version in [18] has the hypothesis that X satisfies a stronger uniform mixing condition (strong irreducibility), but goes through with no changes in the block gluing case.)

Lemma 3.2. *If X is a block gluing nearest neighbor $\mathbb{Z}^d$ shift of finite type with gap size $g \in \mathbb{N}_0$, then for any $m \in \mathbb{N}$,*

$$\frac{\log \left| \mathcal{L}_{[1,m]^d}(X) \right|}{(m+g)^d} \leqslant \mathrm{h}_{\mathrm{top}}(X) \leqslant \frac{\log \left| \mathcal{L}_{[1,m]^d}(X) \right|}{m^d} \ .$$

The first inequality in Lemma 3.2 comes from the possibility to generate valid points in X by independently placing arbitrary globally admissible cuboid patterns of shape $[1,m]^d \subsetneq \mathbb{Z}^d$ with their lower left corners on the grid $(m+g)\mathbb{Z}^d \leqslant \mathbb{Z}^d$ and filling the remaining sites using block gluing of X. The second inequality is obvious from the definition of topological entropy.

We can now prove Theorem 1.4.

Proof of Theorem 1.4. Consider an arbitrary block gluing $\mathbb{Z}^2$ SFT X. It follows from Lemma 3.2 that for large $m \in \mathbb{N}$ the quantity $\frac{\log|\mathcal{L}_{m,m}(X)|}{m^2}$ is a close approximation of $\mathrm{h}_{\mathrm{top}}(X)$. In particular, for every $m \in \mathbb{N}$, we get a bound linear in $\frac{1}{m}$:

$$\left| \mathrm{h}_{\mathrm{top}}(X) - \frac{\log |\mathcal{L}_{m,m}(X)|}{m^2} \right| \leqslant \log |\mathcal{L}_{m,m}(X)| \cdot \left(\frac{1}{m^2} - \frac{1}{(m+g)^2} \right)$$

$$\leqslant \frac{2g \log |\mathcal{L}_{m,m}(X)|}{m^2 \cdot (m+g)} \leqslant \frac{2g \log |\mathcal{L}_{m,m}(X)|}{m+g} \leqslant \frac{2g \log |\mathcal{A}|}{m} \ .$$

(In the second to last inequality, the trivial estimate $|\mathcal{L}_{m,m}(X)| \leqslant |\mathcal{A}|^{m^2}$ was used.) So, to get an upper bound on the time it takes to calculate an approximation of $\mathrm{h}_{\mathrm{top}}(X)$ to within a certain error linear in $\frac{1}{m}$, it suffices to determine how many

steps are needed to actually compute the number of globally admissible $m \times m$ square patterns, i.e. to determine the cardinality $|\mathcal{L}_{m,m}(X)|$.

For any fixed $m \in \mathbb{N}$, we can do this with the following algorithm:

(1) Initialize Counter $:= 0$.

(2) Enumerate all locally admissible $m \times m$ square patterns $P \in \mathcal{L}_{m,m}^{\mathrm{loc}}(X)$. To do this generate – one at a time – all $|\mathcal{A}|^{m^2}$ possibilities to fill a $m \times m$ square with symbols from $\mathcal{A}$ and, when placing each new symbol, check with its already placed neighbors to ensure that none of the nearest neighbor rules defining X is violated. This takes a constant number of steps per symbol and thus the whole enumeration gives at most a multiplicative factor of $\mathcal{O}(|\mathcal{A}|^{m^2})$ for the run-time of Steps (3) and (4).

(3) For each locally admissible pattern P constructed in Step (2), enumerate all patterns $Q' \in \mathcal{A}^{(|\mathcal{A}|^{2g+1}+1)(g+m) \times (2g+m+2)}$ which respect the restrictions given in the proof of Lemma 3.1, namely the top and bottom row of Q' have to be equal and the horizontal strip of height m in the center of Q' has to contain $|\mathcal{A}|^{2g+1} + 1$ copies of the pattern P, separated from each other by the gap size g. Note that each such pattern Q' has only

$$\left(|\mathcal{A}|^{2g+1} + 1\right)(g+m) \cdot (2g+m+1) - \left(|\mathcal{A}|^{2g+1} + 1\right) \cdot m^2$$
$$\leqslant \left(|\mathcal{A}|^{2g+1} + 1\right)(3g+1)(g+m) \in \mathcal{O}(m)$$

a priori undetermined sites, which can be successively filled with symbols from $\mathcal{A}$. Validity of the nearest neighbor rules is again checked during the construction of all possible Q' (rejecting those Q' which are not locally admissible for X) and only increases the run-time by a fixed multiplicative constant. So enumerating all such patterns Q' takes at most $|\mathcal{A}|^{\mathcal{O}(m)}$ steps.

(4) If any (the first) complete pattern Q' is constructed for P, the Counter variable is increased by one (Counter $:=$ Counter $+ 1$) and the algorithm continues at Step (2) with the next pattern P. This step takes a constant amount of time which is uniform over all P.

(5) After all patterns P have been tested, the algorithm outputs the actual value of Counter and terminates.

As was shown in Lemma 3.1, it is possible to finish the construction of a locally admissible pattern Q' in Step (3) iff Q' has a subpattern Q containing P which periodically tiles the plane. Hence for every increase of Counter, the corresponding pattern P is globally admissible, and at the end of the algorithm Counter $= |\mathcal{L}_{m,m}(X)|$.

Analyzing the total run-time of the algorithm yields the existence of constants $C_1, C_2 \in \mathbb{R}$ (depending only on the actual implementation of the algorithm) such that the maximal number of steps is bounded from above by $C_1 \cdot |\mathcal{A}|^{m^2} \cdot |\mathcal{A}|^{C_2 \cdot m}$, which is asymptotically less than $|\mathcal{A}|^{C_3 \cdot m^2}$ for a slightly bigger constant $C_3 \in \mathbb{R}$. To get an approximation of $\mathrm{h}_{\mathrm{top}}(X)$ with error at most $\frac{1}{n}$ ($n \in \mathbb{N}$), we therefore need no more than $|\mathcal{A}|^{C_3 \cdot (2g \cdot n \cdot \log |\mathcal{A}|)^2} \leqslant 2^{C \cdot n^2}$ steps, where $C \in \mathbb{R}$ is again another sufficiently large explicit constant. This establishes our claim and finishes the proof. $\qquad \square$

We quickly point out that for any $d > 2$, if X is assumed to have the uniform filling property and to contain at least one periodic point $x^* \in X$, then it is clear that a version of Lemma 3.1 is true for Q with all dimensions bounded from above

by $m + 2g + p$ for $p \in \mathbb{N}$ the largest of the cardinal periods of x^*. Then the same algorithm will generate an approximation to $\mathrm{h}_{\mathrm{top}}(X)$ with error less than $\frac{1}{n}$ in runtime less than $2^{C \cdot n^d}$ for some constant $C \in \mathbb{R}$.

4. Accelerated Turing machines

Assuming the reader is familiar with the standard notion of a Turing machine (TM) we will not repeat its formal definition, but rather refer to the classic textbook by Hopcroft and Ullman [10] for further details. Here instead of using a standard TM, we introduce the novel concept of an accelerated Turing machine. The main additional property of an accelerated TM is that it can instantly execute certain predefined operations affecting the currently used (finite) segment of tape-space in a single step, while an ordinary TM would need to execute a corresponding (finite) subroutine. The idea behind this notion is to have a model for computations given by local rules which is better adapted to the capabilities of simulating computations inside a multilayer $\mathbb{Z}^2$ SFT than standard TMs. We point out that our new model does not increase the computational power which is still that of a standard TM. However, it does decrease computation time considerably (instant operations do NOT just provide a linear speed-up of computation time).

Definition 4.1. The term **accelerated Turing machine** will denote a deterministic (or non-deterministic) multi-tape Turing machine (with its usual finite set of states and an arbitrary but fixed number of tapes whose cells contain symbols from a finite alphabet) starting off with each read-write head on the leftmost cell of the corresponding one-sided infinite tape which is initially filled entirely with an otherwise unused blank symbol. Computation now follows the rules of a standard TM – in each step, all but finitely many cells on each tape are still filled with the blank symbol and each head only modifies the symbol at its current position and either stays at this position or moves one cell left/right – except when certain designated machine states are reached, in which case an additional predefined instant operation (still defined cell-wise only using local rules) is performed on a part or on all of a tape's content first and then the usual standard TM step is carried out.

Examples of such instant operations[4] acting on whole tapes include the following (in each operation, if a "bounded" constant is referenced, this is with respect to a fixed uniform upper bound for the entire accelerated Turing machine):

Fill: Cell-wise fill the entire currently used segment of a tape with a single symbol (or a fixed periodic sequence of bounded period length). This includes filling the tape with blanks (= deleting its content) and can also be used to check the currently used tape segment has length a multiple of some constant.

Copy: Cell-wise copy the content of one tape to another tape starting from the leftmost cell. This can also be combined with a fill/delete operation.

Shift: Cell-wise shift the content of one tape by a bounded number of cells to the right or left (this requires filling in resp. discarding some symbols).

Multiply/divide: Multiply or divide the content of one tape (viewed as the binary representation of a natural number) by a fixed constant. Shifting one step

---

[4]Those operations are actually more in the spirit of a register machine than a TM which, instead of tapes, has a finite number of registers capable of holding any natural number (or finite word over some finite alphabet) and a reduced set of steering and arithmetical operations that act instantaneously on the content of a single (or a group of) register(s).

to the right/left would correspond to multiplication/division by 2, shifting and adding (with running carry on an auxiliary tape) corresponds to multiplication by 3 etc. More generally a tape's content can be modified by cell-wise applying any fixed local (cellular automaton) replacement rule.

Add/subtract: Cell-wise add or subtract the content of one tape to/from another (both contents are viewed as binary representations of natural numbers) starting at the leftmost cell using an auxiliary tape to store a running carry of adding each corresponding pair of cells from the other tapes.

Compare: Cell-wise compare the content of two (or more) tapes, starting from the leftmost/rightmost cell. By using an auxiliary tape for sending a corresponding signal, information about the outcome of the comparison can be transferred back to the read-write head, to update its state in the following step.

Send signal: Send the read-write head's current symbol or the tape's rightmost/leftmost symbol to all currently used cells, using an auxiliary tape which gets filled with this symbol. The information about this symbol is then available everywhere (e.g. at the read-write heads' positions of other tapes).

Move head: Transfer the read-write head's position, along with its current state, to the leftmost/rightmost currently used cell or the previous/next occurrence of some symbol. Again, this can be combined with some other instant operations.

An even more powerful set of instant operations can be obtained by using markers, i.e. special symbols which can be placed in and removed from cells by the read-write head in the course of the standard TM steps (or in an instant fill/replace operation). This allows us to have any of the above operations (and their combinations) act on a thereby marked segment of the tape, e.g. fill/compare/add only cells between a pair of such markers, check markers are spaced by a multiple of some fixed length, send information from a marked cell to the head's current position, let the head jump to the previous/next marker instead of the end of the tape, etc.

All this models a (non-)deterministic algorithm evolving in discrete time steps, modifying the content of a bounded (sufficiently large) number of one-sided infinite tapes[5] according to a finite set of local rules. The time evolution of this algorithm thus can be embedded into (consecutive rows within points of) a $\mathbb{Z}^2$ SFT.

Note, however, that the notion of computability in this new framework coincides with the usual one. Our accelerated Turing machines have the same computational power as standard (deterministic, one-tape) TMs; the only difference is in computation time. One can think of their enhanced capabilities as calling a subroutine, given by another deterministic standard TM, which performs the corresponding instant operation. The running time of this subroutine would be linear in the currently used space on the machine's tapes, whereas the accelerated TM performs the whole subroutine in a single step. Hence our accelerated TMs are not just standard TMs using a constant linear speed-up. In fact the overall decrease in computation time of an algorithm depends on the number of instant operations each of which lets us gain a number of steps comparable to the length of the currently used tape segment, which might grow unbounded during the algorithm's execution.

Now that we have defined the concept of an accelerated TM, let us prove that numbers with an eventually affine continued fraction expansion (i.e. numbers in class 1.6.(3)) satisfy the computability condition **(C)**:

---

[5]The content of those tapes will fill the rows on different superimposed layers of our $\mathbb{Z}^2$ SFTs.

Let $\alpha' = [a_0; a_1, a_2, \ldots, a_{n'-1}, a_{n'}, a_{n'+1}, \ldots]$ with $a_n = C \cdot n + D$ for all $n \geqslant n'$ ($C \in \mathbb{N}$, $D \in \mathbb{Z}$ fixed) and, by increasing n' if necessary, suppose that $a_{n'} \geqslant C$. Clearly there exists a (standard) Turing machine that has a hard-wired algorithm which – starting from a blank tape – writes down the binary representation of a single partial quotient a_n with $n < n'$ one bit at a time. Here and in the following, our convention for such binary representations is to have the least significant bit at the left end of the tape and all following bits consecutively to its right. By putting those TMs as subroutines into an accelerated TM, we can ensure that the tape contains such binary representations of the first N partial quotients after $\sum_{n=1}^{N}\big(\lfloor \log_2 a_n \rfloor + 1\big)$ steps, which is surely sufficient for the bound in condition **(C)**. Recall that in an accelerated TM, whenever a binary representation has been finished, independently of its length, it can be instantly stored on another tape and be deleted by immediately filling the work tape with blanks in one step; this works due to the enhanced copying and filling capabilities of an accelerated TM.

Once the first $n' - 1$ partial quotients are finished, the accelerated TM switches to a second part of its program, now using three tapes, defined as follows: starting from blank tapes as before, the TM now puts down at the same time a binary representation of $a_{n'} = C \cdot n' + D$ on one tape and a binary representation of the parameter C on the second tape (both starting at the left end of their respective tape). Since $C \leqslant a_{n'}$, this takes not more than $\lfloor \log_2 a_{n'} \rfloor + 1$ steps and already produces $a_{n'}$. Starting from here, the accelerated TM now is capable of producing all remaining partial quotients in real time, i.e. one per step, by simply adding the content of the second tape, i.e. C, to the content of the first tape over and over again, using the third tape to propagate the carry produced by adding each pair of bits from the left to the right. Hence the bound from condition **(C)** is met (for $N \in \mathbb{N}$, $N \geqslant n'$, a_N is ready after at most $\sum_{n=1}^{n'}\big(\lfloor \log_2 a_n \rfloor + 1\big) + (N - n')$ steps).

Roughly speaking, for numbers in class 1.6.(4), we would need a similar accelerated TM, using $2p+1$ tapes instead of 3, which are then used to store and add the coefficients $C_0, C_1, \ldots, C_{p-1}$ to their corresponding partial quotients $a_{ip}, a_{ip+1}, \ldots, a_{ip+p-1}$ for $ip \geqslant n'$ ($i \in \mathbb{N}$), while keeping track of which coefficient is being worked on via a p-cycle in the machine's states. We leave the details to the interested reader.

Let us be a little more specific about how the computations of such accelerated TMs can be implemented inside a $\mathbb{N}^2$-quarter-plane, using only local rules. (Those local rules will then be used to define a $\mathbb{Z}^2$ SFT as usual.) Assuming that the accelerated TM has $I \in \mathbb{N}$ one-sided infinite tapes, each containing a single read-write head, we will use $I + 2$ superimposed layers to encode the corresponding computation on $\mathbb{N}^2$ (and thus later in a $\mathbb{Z}^2$ SFT). Rows on the first I of those layers will be used to hold the content of the respective tape, i.e. a one-sided infinite sequence over some finite alphabet eventually ending in all blank symbols, in the respective step of the computation. Moreover, the current position of the read-write head will be marked by a "$\wedge$" (decorating the usual symbol contained in a cell) and, to ensure that on each tape there is (at most) one such position, we force all symbols to the right of the head's position to be decorated by "$\rightarrow$", while all symbols to its left must carry a "$\leftarrow$". To summarize, the alphabet for the ith layer ($1 \leqslant i \leqslant I$) is defined to be the product $\Gamma_i \times \big\{ \leftarrow, \, \wedge, \, \rightarrow \big\}$, where Γ_i is the ith tape's alphabet. Then, a nearest neighbor rule only allows horizontal transitions of the form $\overleftarrow{c_1}\,\overleftarrow{c_2}$, $\overleftarrow{c_1}\,\hat{c_2}$, $\hat{c_1}\,\overrightarrow{c_2}$ or $\overrightarrow{c_1}\,\overrightarrow{c_2}$ for $c_1, c_2 \in \Gamma_i$. Now, layer $I + 1$ will be used to actually control the computation, while layer $I + 2$ acts as an auxiliary layer, whose rows may be used in

some of the instant operations. Each of the rows on layer $I + 1$ contains a constant sequence t^∞ of some symbol $t \in \delta \subseteq Q \times \prod_{i=1}^{I} \Gamma_i \times Q \times \prod_{i=1}^{I} \Gamma_i \times \{L, N, R\}^I$, which codes a legal transition of the TM (here Q denotes its set of states, $\{L, N, R\}$ is the set of possible head movements, and δ is the usual transition function/relation of a (non-)deterministic multi-tape TM). This way, all of the following is available at all cells: the machine's current and next state, as well as the symbols seen at all read-write heads' positions, their replacements in the actual step, and the heads' movements. Consistency of the computation is assured by the following rules, all of which have a simple and obviously local character (in each of the following, $t = \big(q, (c_1, c_2, \ldots, c_I), q', (c'_1, c'_2, \ldots, c'_I), (m_1, m_2, \ldots, m_I)\big) \in \delta$ represents the transition tuple used in row $j \in \mathbb{N}$ of layer $I + 1$):

• The symbol marked by "$\wedge$" on layer i ($1 \leqslant i \leqslant I$) in row j is forced to be $\hat{c}_i$. (In the exceptional case caused by compactness of the $\mathbb{Z}^2$ SFT where there is no symbol marked by "$\wedge$" in row j on layer i, the corresponding c_i in t is unconstrained.)

• Whenever state q' is one of the ordinary TM states not triggering an instant operation, all symbols $c \in \Gamma_i$ in row j of each layer i at positions not marked by "$\wedge$" are kept unchanged, i.e. reappear at the same position in row $j + 1$, whereas the remaining symbol at the position of the "$\wedge$" in row j is forced to be a decorated version of c'_i in row $j+1$ (again if there is no such position, nothing is implied here). In addition, on each layer i the position of the marking "$\wedge$" in row $j+1$ has shifted with respect to its position in row j according to the value of m_i (either it stays unchanged "N" or it moves one step to the left "L" or right "R"); furthermore row $j + 1$ on the auxiliary layer $I + 2$ is forced to be entirely filled with blanks.

• If state q' is one of the TM states designated to perform an instant operation defined by local rules, this operation is executed by first modifying the content of some (all) tape(s) and possibly the heads' positions, thus producing the symbols and decorations for row $j + 1$ on layers 1 to I (including row $j + 1$ on the auxiliary layer $I + 2$ which can either be used in the operation, e.g. to store a running carry or to send back-and-forth some information, or, if not needed, is again to be filled with blanks). The content of tapes not affected by the instant operation is kept unchanged and the replacements of symbols at positions marked by "$\wedge$" given by $(c'_1, c'_2, \ldots, c'_I)$ as well as the shifts of the decorations according to $(m_1, m_2, \ldots, m_I)$ then take place as before.

• In both cases, state q' has to reappear as the first component of the transition tuple $t' \in \delta$ used in row $j + 1$ on layer $I + 1$, so that the consecutive machine states match up forming a legal computation.

5. A SUFFICIENT CONDITION – PROOF OF THEOREM 1.7

The goal of this section is to construct, for any non-negative real number $\alpha \in \mathbb{R}_0^+$ which satisfies computability condition **(C)**, a block gluing $\mathbb{Z}^3$ SFT X of topological entropy $\mathrm{h}_{\mathrm{top}}(X) = \alpha$. The portion of Theorem 1.7 for $d > 3$ then follows from taking full $\mathbb{Z}^{d-3}$-extensions of those $\mathbb{Z}^3$ shifts.

Suppose $\alpha = \alpha' \cdot \log M$ is the representation given by Definition 1.5. Note that we can basically restrict ourselves to the case where $\alpha' \in (0, 1)$; for $\alpha' = 0$ the trivial one-point SFT is block gluing of entropy zero, for $\alpha' = 1$ there is a full $\mathbb{Z}^3$ shift on M symbols that is clearly block gluing of entropy $\alpha = \log M$, and whenever $\alpha' > 1$, we can use the product of a full $\mathbb{Z}^3$ shift on $M^{\lfloor \alpha' \rfloor}$ symbols with a block gluing $\mathbb{Z}^3$ shift

of finite type (to be constructed below) of entropy $\left(\alpha' - \lfloor\alpha'\rfloor\right)\log M$, which is clearly block gluing of topological entropy $\alpha = \alpha'\log M = \left(\alpha' - \lfloor\alpha'\rfloor\right)\log M + \log M^{\lfloor\alpha'\rfloor}$.

Our main construction proceeds in several steps: The first is to use methods similar to [9] to construct an auxiliary zero-entropy $\mathbb{Z}^3$ SFT $X' \subseteq \mathcal{A}'^{\mathbb{Z}^3}$ in which the alphabet $\mathcal{A}'$ can be partitioned into two sets $\mathcal{A}' = \mathcal{A}'_0 \,\dot\cup\, \mathcal{A}'_1$ such that the frequency of symbols from $\mathcal{A}'_1$ in every point of X' is very close to $\alpha' \in (0,1)$. The second step is to create a new $\mathbb{Z}^3$ SFT $X'' \subseteq \mathcal{A}''^{\mathbb{Z}^3}$ with alphabet $\mathcal{A}'' := \mathcal{A}'_0 \,\dot\cup\, \left(\mathcal{A}'_1 \times \{1, 2, \ldots, M\}\right)$ by splitting each symbol in $\mathcal{A}'_1$ into M independent copies, which, as in [9], will cause X'' to have entropy $\alpha'\log M$. Thus, we will think of α' as a frequency of certain symbols in our $\mathbb{Z}^3$ SFTs X' and X'' which, in the latter shift, come with multiplicity M. Unfortunately, the construction in these first two steps only yields a degenerate and highly non-mixing $\mathbb{Z}^3$ SFT X'', which is why we need the third and final step. Here, we create X from X'' by introducing some new symbols to $\mathcal{A}''$, called "wall symbols", which will cause X to finally be block gluing. The fundamental difficulty is to ensure that the entropy is not increased by this addition, since if it were, we would have no real technique for controlling the exact increase.

We remark that if we instead wanted X to have the uniform filling property (or to be strongly irreducible) rather than being just block gluing, this last step could not possibly work. It is known (see [22, Lemma 2.7]) that a $\mathbb{Z}^d$ SFT satisfying the uniform filling property is entropy minimal, meaning that it cannot contain a proper subsystem of equal entropy, thus making an entropy-preserving construction of $X \supsetneq X''$ impossible. Luckily, for some $\mathbb{Z}^d$ SFTs it is possible to introduce additional symbols so as to make (or keep) those shifts block gluing without increasing their entropy. This was first done in [22] and in [5] for a full shift. It turns out that in order for the entropy to not increase under our method, a subshift must satisfy a very restrictive technical property which we state in the following definition:

Definition 5.1. A $\mathbb{Z}^3$ shift of finite type Z is **upgradable**[6] if there is a non-negative real constant $C \in \mathbb{R}_0^+$ so that $\left|\mathcal{L}^{\mathrm{loc}}_{k,l,m}(Z)\right| \leqslant e^{\mathrm{h_{top}}(Z)klm + C(kl + km + lm)}$ holds for all $k, l, m \in \mathbb{N}$.

Note that a full shift on any alphabet $\mathcal{A}$ obviously satisfies Definition 5.1, as $\left|\mathcal{L}^{\mathrm{loc}}_{k,l,m}\left(\mathcal{A}^{\mathbb{Z}^3}\right)\right| = \left|\mathcal{L}_{k,l,m}\left(\mathcal{A}^{\mathbb{Z}^3}\right)\right| = e^{\log|\mathcal{A}|\cdot klm}$; this is the reason behind similar arguments used in [5, 22]. Nevertheless, being upgradable is a quite restrictive condition, and thus our construction of X'' is by necessity fairly intricate in order to achieve upgradability.

We begin our construction with any non-negative real number $\alpha \in \mathbb{R}_0^+$ satisfying computability condition **(C)**. As we will see, this restriction on α is an intrinsic consequence of our method and in view of Theorem 1.4, some condition on the computability of α seems unavoidable. By Definition 1.5 and the remarks above we may assume $\alpha = \alpha' \cdot \log M$ with $1 < M \in \mathbb{N}$ a natural number and $\alpha' \in (0,1)$. Depending on the value of α' we distinguish two principal constructions:

First, suppose $\alpha' = \frac{r}{s} \in \mathbb{Q} \cap (0,1)$. In this case we may simply define $X' \subseteq \{0,1\}^{\mathbb{Z}^3}$ to be the full $\mathbb{Z}^2$-extension of the $\mathbb{Z}$ shift of finite type $Y' := \mathrm{Orb}\left\{(0^{s-r}\,1^r)^\infty\right\}$, i.e. $X' := \left\{x \in \{0,1\}^{\mathbb{Z}^3} \mid \forall \vec{\imath} \in \mathbb{Z}^3 : \; x|_{\vec{\imath} + \mathbb{Z}\vec{e}_1} \in Y'\right\}$. It is apparent that Y' can be

---

[6]Similarly we could define upgradability for general $\mathbb{Z}^d$ shifts of finite type ($d \neq 3$), e.g. requiring $\left|\mathcal{L}^{\mathrm{loc}}_{k,l}(Z)\right| \leqslant e^{\mathrm{h_{top}}(Z)kl + C(k+l)}$ for all $k, l \in \mathbb{N}$ in the 2-dimensional setting. However in our construction we will be focusing on the case $d = 3$.

defined by local (1-dimensional) rules, hence X' is a $\mathbb{Z}^3$ SFT. As each row of a point in X' is actually an element of Y' it is easy to see that the frequency of 1s – which is just the frequency of 1s in the periodic point $(0^{s-r}\, 1^r)^\infty$ – is equal to $\frac{r}{s} = \alpha'$ and constant across all elements of X'. Moreover the entropy of X' is zero and thus splitting 1s into M independent copies immediately results in a $\mathbb{Z}^3$ SFT X'' of entropy $\mathrm{h_{top}}(X'') = \alpha' \cdot \log M = \alpha$. Theorem 5.11, proved later in this section, allows us to upgrade X'' to a block gluing version X with equal entropy as long as X'' satisfies the condition stated in Definition 5.1. Denote by $Y'' := \left\{ x|_{\mathbb{Z}\vec{e}_1} \mid x \in X'' \right\}$ the horizontal projective subdynamics of X'' (i.e. the $\mathbb{Z}$ subshift consisting of the rows of X''), which is basically Y' with the 1s split into M independent copies, and note that then $\left| \mathcal{L}_k^{\mathrm{loc}}(Y'') \right| \leqslant s \cdot M^{\alpha' \cdot k + r}$ for every $k \in \mathbb{N}$. This is because there are s ways to pick a starting point in the orbit of $(0^{s-r}\, 1^r)^\infty$, and the maximal number of 1s in a block of length k is bounded by k times the frequency $\alpha' = \frac{r}{s}$, plus a possible overage less than or equal to r, from the cases where k is not a multiple of s. As X'' is (still) a full $\mathbb{Z}^2$-extension of Y'', this yields

$$\left| \mathcal{L}_{k,l,m}^{\mathrm{loc}}(X'') \right| = \left| \mathcal{L}_k^{\mathrm{loc}}(Y'') \right|^{lm} \leqslant \left(s \cdot M^{\alpha' \cdot k + r} \right)^{lm} = (s \cdot M^r)^{lm} \cdot M^{\alpha' \cdot klm}$$
$$= e^{(\log s + r \log M) lm} \cdot e^{\mathrm{h_{top}}(X'') klm}$$

which clearly implies upgradability of X'' for $C := \log s + r \log M \in \mathbb{R}_0^+$. Theorem 5.11 then concludes the proof of our main result for $\alpha' \in \mathbb{Q} \cap (0,1)$. (The construction in this case also works for $d = 2$.)

The complementary case of $\alpha' \in (0,1)\backslash\mathbb{Q}$ is less straightforward and will take up most of the remainder of this section.

Step 1 (Constructing X'). This time our $\mathbb{Z}^3$ SFT X' will consist of various superimposed layers as defined in [9]. The admissible configurations on each of those layers can be entirely described by a set of local rules which use information from nearby symbols both in this layer as well as in some (or all) of the other layers.

The first layer, which we from now on refer to as the **base layer**, consists of 0s and 1s, and is forced by simple nearest neighbor rules to be constant in the directions of the cardinal vectors $\vec{e}_2$ and $\vec{e}_3$. Therefore, the base layer of any point in X' will be entirely determined by its restriction to a single row in the $\vec{e}_1$-direction. We will use the rules on the other layers to restrict further these 0/1-configurations, controlling what rows can actually show up on the base layer.

The next layer, called the **board layer**, will be used to, for any point $x' \in X'$, partition each $\vec{e}_1\vec{e}_2$-cross section $x'|_{\mathbb{Z}^2 \times \{m\}}$ ($m \in \mathbb{Z}$ fixed) into square "boards" on which accelerated Turing machines – running on another layer called the construction layer (to be defined below) – will eventually perform some computation. Rather than using the more general construction of Mozes (introduced in [17]) as was done in [9], we use the Robinson tilings from [21]. This is because in light of Definition 5.1, we need very good control on the number of locally admissible patterns on large rectangular prisms to show upgradability of X'', and the use of the simpler and more explicit Robinson system makes such technical conditions easier to verify.

For an easy-to-read synopsis of the Robinson system and its relevant properties, see [20], pp. 39-47. In Figure 2 we recall the 56 square tiles forming its alphabet. The nearest-neighbor rules for putting together these decorated squares to produce a valid tiling of $\mathbb{Z}^2$ are that across every edge shared by a pair of adjacent tiles, arrows of each type (thick or thin) must match head to tail, and the parity check
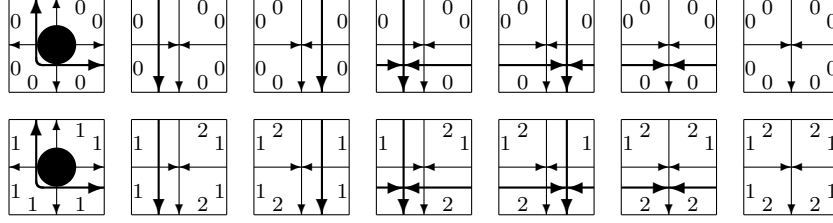
FIGURE 2. The alphabet used in the Robinson tilings (displayed tiles can still be rotated giving a total of $4 \times 14 = 56$ symbols).

symbols (0, 1 or 2) on both sides of the shared edge must sum to 2. This forces parity check symbols in any row or column either to be constantly 1 or to strictly alternate between 0s and 2s and those two types of rows/columns again must strictly alternate in each tiling. In our construction of the board layer, we will not use all of the information contained in these tiles, but will be primarily interested in the locations of the two tiles from the leftmost column of Figure 2 – usually called **crosses**, as in their interior the thinline arrows spread out in all 4 directions – in a valid tiling. So, to distinguish them from the rest of the tiles, we mark these two cross symbols with a central "black dot". The remaining non-cross symbols – also called **arms** in [20, 21] – can be thought of as carrying a "white dot" (not shown). With this simplification, points of the Robinson system then almost correspond to points of the substitution system defined in Section 6 of [9]. There are, however, some exceptional points in the Robinson system which contain a "fault line" consisting entirely of white dots (arms), in which the two half-planes adjacent to the fault line have a non-zero offset; such points do not exist in the substitution system. Figure 3 shows the position of the black dots (together with the information about the thickline arrows) induced by a typical point of the Robinson system.

Now we define the board layer, which will actually consist of three sublayers: Two of them, called **Robinson sublayers**, are independently formed from valid Robinson tilings, and the third one, called the **synchronization sublayer**, is used to relate the two Robinson sublayers to force the final structure of square boards mentioned above. More specifically: the first sublayer is constant along the $\vec{e}_1$-direction, and must contain a legal point of the Robinson system on each (identical) $\vec{e}_2\vec{e}_3$-cross section which is enforced by the standard Robinson rules. The second sublayer is constant along the $\vec{e}_2$-direction, and must contain a legal Robinson tiling on each (identical) $\vec{e}_1\vec{e}_3$-cross section. A priori any pair of points of the Robinson system can be used in the respective cross sections to induce these first two sublayers, while the synchronization sublayer will impose an additional relation between them, breaking their independence. Note that since the Robinson system is defined by local rules, and it is trivial to force constancy along a cardinal direction with nearest neighbor rules, the SFTness property extends to the two Robinson sublayers. As each Robinson sublayer is constant in one of the cardinal directions, this means that each sublayer can be thought of as consisting of lines in $\mathbb{Z}^3$ traced out by the black dots. The first sublayer induces such lines in the $\vec{e}_1$-direction, the
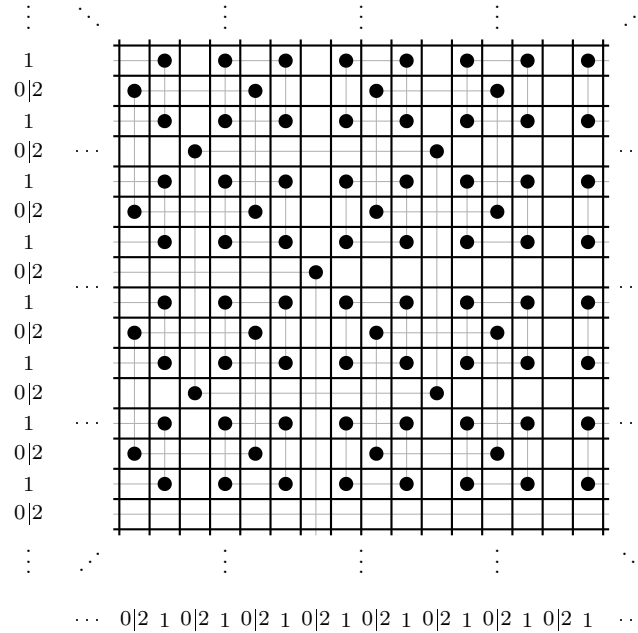
FIGURE 3. The black and white dots induced by a (typical) point of the Robinson system (thickline arrows shown as gray lines; parity of rows/columns recorded along the left and bottom edges).

second induces lines in the $\vec{e}_2$-direction; thus their union will create a rectangulation of each $\vec{e}_1\vec{e}_2$-cross section of $\mathbb{Z}^3$ (possibly including infinite rectangles).

For our construction, we would like these rectangulations on (most) $\vec{e}_1\vec{e}_2$-cross sections to be regular grids given by aligned copies of congruent squares rather than general rectangles. This is basically (aside from some possible exceptional cross-sections) accomplished by superimposing the synchronization sublayer, which has an extremely simple SFT structure. Its alphabet consists of two symbols, □ (blank) and ◻ (diagonal), and there are two types of local rules. The first is that a blank symbol □ is not allowed to coexist at the same site with two black dots (crosses) on the two Robinson sublayers (at least one of those symbols has to be a white dot), and a diagonal symbol ◻ on the synchronization layer can only coexist with either two black dots (crosses) or two white dots (arms) present at the same site on each of the two Robinson sublayers. The second local rule is that the presence of a diagonal symbol ◻ at any site $\vec{\imath} \in \mathbb{Z}^3$ forces the presence of diagonal symbols at sites $\vec{\imath} + \vec{e}_1 + \vec{e}_2$ and $\vec{\imath} - \vec{e}_1 - \vec{e}_2$. This way diagonal symbols have to propagate infinitely along the $(\vec{e}_1 + \vec{e}_2)$-direction and have to cross through all corners, forcing complete rectangles to be squares. This finishes the board layer's description.

In order to show the claimed upgradability of our $\mathbb{Z}^3$ SFT X' we have to control the structure of locally admissible cuboid patterns in the board layer. For this we first need to recall some facts about locally admissible patterns in the Robinson system. Here we essentially follow Sections 3 and 8 of [21]. First recall that the parity check symbols in the tiles of Figure 2 force the presence of crosses (with

parity symbols 1s along their 4 edges) at alternating sites in every other row (and column) of any locally admissible pattern P with rectangular shape $R = [1, k] \times [1, l]$ ($4 \leqslant k, l \in \mathbb{N}$). This then forces arms (with parity symbols 1s and 2s) adjacent to these crosses. Moreover any two of those crosses with horizontal (or vertical) separation 2 either face each other, meaning that the tips of their thickline arrows point towards each other, or are back-to-back, i.e. thickline arrows point away from each other. If $k, l \geqslant 6$ our locally admissible rectangular pattern P contains what is called a 3-square in [21], having as its corners 4 crosses at distance 2 pairwise facing each other. Note that the central tile in such a 3-square again has to be a cross (with parity symbols 0s) and that the 3 tiles adjacent to either side of the 3-square have to be arms. In particular the orientation of its central cross forces the orientation of the arms adjacent to one of the 3-square's corners and also forces another cross (with parity symbols 0s) diagonally adjacent to this corner. This behavior then repeats on bigger scales forcing four 3-squares to be aligned into a 7-square, four 7-squares to form a 15-square etc. as long as those $(2^{i+1} - 1)$-squares ($i \in \mathbb{N}$) are completely contained inside the rectangle R. This process also excludes, for $k, l \geqslant 2^{i+3}$ ($i \in \mathbb{N}$), the presence of more than one horizontal resp. vertical fault line having non-aligned complete $(2^{i+1} - 1)$-squares on either side; a fact which will be key in the proof of the next lemma. So looking only at the complete $(2^{i+1} - 1)$-squares inside R the pattern P has to fall into one of the following 3 categories:

(1) all complete $(2^{i+1} - 1)$-squares inside R are aligned;
(2) there is a unique vertical fault line with a non-zero vertical offset of even size between the complete $(2^{i+1} - 1)$-squares to its left and its right;
(3) there is a unique horizontal fault line with a non-zero horizontal offset of even size between the complete $(2^{i+1} - 1)$-squares above and below it.
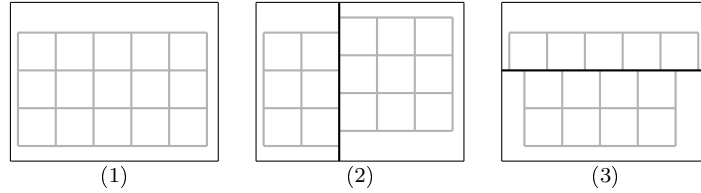

FIGURE 4. Types (1)–(3) of locally admissible patterns on rectangles $R = [1, k] \times [1, l]$ ($k, l \geqslant 2^{i+3}$) in the Robinson system (displaying the possible alignment of complete $(2^{i+1} - 1)$-squares).

Figure 4 illustrates categories (1)–(3) where we explicitly leave some room near the boundary of R possibly containing incomplete and not aligned $(2^{j+1} - 1)$-squares ($j \leqslant i$) which we can not fully control with the local rules of the Robinson system.

Lemma 5.2. *For any $k, l \geqslant 2^{i+3}$ ($i \in \mathbb{N}$) and any pattern P locally admissible for the Robinson system and with rectangular shape $R = [1, k] \times [1, l]$ the following holds: If for some $r \in \mathbb{N} \cap [1, l]$ row r of P contains either zero crosses or a single cross, then for any $s \in \mathbb{N} \cap [1, l]$ such that $2^i \nmid (r - s)$, row s of P contains two consecutive crosses with separation $2^{\nu_2(r-s)+1}$. (Here, for any integer $n \in \mathbb{Z} \backslash \{0\}$, $\nu_2(n) := \max\{m \in \mathbb{N}_0 \mid 2^m \text{ divides } n\}$ denotes the 2-adic valuation.)*

Proof. The proof of this claim in the case where P is a full $(2^i - 1)$-square for any $i \in \mathbb{N}$ is easily proved by induction, and we leave it to the reader. It remains to use this to prove the lemma for general P.

First note that the hypothesis $k, l \geqslant 2^{i+3}$ forces P to contain at least 9 complete $(2^{i+1} - 1)$-squares arranged in a (skew) 3×3 configuration and that thus – no matter which category (1)–(3) P falls into – a rectangular portion of width at least $\lceil \frac{k-1}{2} \rceil \geqslant 2^{i+2}$ and height at least $\lceil \frac{l-1}{2} \rceil \geqslant 2^{i+2}$ containing 4 of those $(2^{i+1} - 1)$-squares forming a complete $(2^{i+2} - 1)$-square will be on one side of a possibly existing horizontal resp. vertical fault line. Call this complete $(2^{i+2} - 1)$-square $Q \subseteq R$ and let $c \in \mathbb{N} \cap [1, l]$ be the number of its central row seen as a row in P.

Suppose row r of P contains at most one cross. As a consequence of the inductive definition of $(2^{j+1} - 1)$-squares $(j \in \mathbb{N})$, see [21], all rows $s \in \mathbb{N} \cap [1, l]$ intersecting Q, except its central row, i.e. row c, contain at least two consecutive crosses with separation $2^{\nu_2(c-s)+1}$. This immediately implies that row r of P either has to contain the central row of Q (thus having a single cross) or otherwise must not intersect Q at all. We will show that in either case the difference $r - c$ has to be divisible by 2^i which allows us to conclude that $\nu_2(r-s) = \nu_2(c-s)$ and will finally prove the lemma. Let us look at the rows above Q; the argument for rows below Q being identical. If the distance from the top row of Q to the top row of R is at least 2^{i+1} the following situation has to occur: First there is a (possibly empty) pile of pairs of horizontally adjacent complete $(2^{i+1} - 1)$-squares sitting right above Q, horizontally exactly aligned with Q and separated from Q as well as one from the next by a single row. Note that all the rows in between those pairs of $(2^{i+1} - 1)$-squares have distance from row c a multiple of 2^i (in fact 2^{i+1}). Hence the lemma actually does not make any claim about them as long as we can prove $2^i \mid (r - c)$. Clearly all rows in P intersecting those pairs of $(2^{i+1} - 1)$-squares contain at least two crosses and are thus excluded from appearing as row r. Moreover the separation of consecutive crosses in any such row $s \in \mathbb{N} \cap [1, l]$ is still given as $2^{\nu_2(c-s)+1}$. Above this pile of exactly aligned pairs of $(2^{i+1} - 1)$-squares there might be another pile of complete $(2^{i+1} - 1)$-squares showing a non-zero horizontal offset with respect to Q. However, since P has to fall into categories (1)–(3) above, the uniqueness of a horizontal fault line then forces this misaligned pile of complete $(2^{i+1} - 1)$-squares to extend up to a distance less than 2^{i+1} from the top row of R without allowing any further horizontal offset. Again all rows separating the $(2^{i+1} - 1)$-squares in this second pile as well as their central rows have distance to row c a multiple of 2^i, while all other rows $s \in \mathbb{N} \cap [1, l]$ inside the pile contain at least two consecutive crosses with separation $2^{\nu_2(c-s)+1}$. This leaves us with either $2^i \mid (r - c)$ or row r sitting above the last complete $(2^{i+1} - 1)$-square of the second pile.

Finally we repeat the same argument starting from the top of Q looking at $(2^i - 1)$-squares instead. There has to be a (possibly empty) pile of exactly aligned groups of 4 horizontally adjacent complete $(2^i - 1)$-squares followed by a (possibly empty) pile of misaligned groups of 3 horizontally adjacent complete $(2^i - 1)$-squares extending all the way up to within a distance of at most $2^i - 1$ from the top row of R. All rows intersecting these $(2^i - 1)$-squares contain at least three consecutive crosses at the correct separation and thus can not appear as r. The rows in between have distance a multiple of 2^i from row c (and, except possibly the very last one, have already been analyzed as part of the $(2^{i+1} - 1)$-squares above). We continue, dealing with $(2^j - 1)$-squares for all values of $j < i$, and the same argument goes

through virtually unchanged, excluding all remaining rows near the top of R, forcing the distance between rows r and c to be a multiple of 2^i as claimed. (The only change is that for $j < i$ there are even more than 3 forced horizontally adjacent squares.) $\qquad\square$

Next we describe the structure of locally admissible patterns in the board layer:

Lemma 5.3. *For any $k, l, m \geqslant 2^{i+4}$ ($i \in \mathbb{N}$) and any pattern P locally admissible for the board layer with cuboid shape $[1, k] \times [1, l] \times [1, m]$, there exists $t \in \mathbb{N} \cap [1, m]$ so that for any $s \in \mathbb{N} \cap [1, m]$ with $2^{i-1} \nmid (t - s)$, the $\vec{e}_1 \vec{e}_2$-cross section $P|_{[1,k] \times [1,l] \times \{s\}}$ is a subpattern of a gridding of $\mathbb{Z}^2$ by aligned squares of size $2^{\nu_2(t-s)+1}$.*

In other words, as we increase the $\vec{e}_3$-coordinate, the $\vec{e}_1 \vec{e}_2$-cross sections of P can be labeled by a subword of the periodic sequence $(\underbrace{* 1\,2\,1\,3\,1\,2\,1\,4\,1\,2\,1\,3\,1\,2\,1 \ldots 1\,2\,1}_{\text{length } 2^{i-1}})^\infty$,

where the label $j \in \mathbb{N}$ represents that the corresponding cross section is part of a rectangulation of $\mathbb{Z}^2$ by aligned congruent squares of size 2^j (hereafter called **boards**), and the label $*$ represents that no information is given about that particular cross section.

Proof. Clearly any such locally admissible pattern P is induced by two patterns P_1, P_2 filling the cross sections of the Robinson sublayers, both locally admissible for the standard Robinson system and with rectangular shapes $R_1 = [1, k] \times [1, m]$ and $R_2 = [1, l] \times [1, m]$ respectively.

First let $r \in \mathbb{N} \cap [1, m]$ be such that row r has at least two crosses (black dots) in one and at least one cross in the other of the two patterns P_1 or P_2. The corresponding $\vec{e}_1 \vec{e}_2$-cross section in the synchronization layer then has to contain diagonals spreading from the (at least) 2 crossings of the corresponding lines traced out by the black dots in P_1 and P_2. As each of those diagonals continue, they hit the other line(s) and thus force both rows r in P_1 and P_2 to have consecutive crosses with the same constant separation appearing periodically throughout all of row r.

Now we define intervals $K \subseteq [1, k]$ and $L \subseteq [1, l]$ both of length 2^{i+3} such that the restrictions $P_1' := P_1|_{K \times [1,m]}$ and $P_2' := P_2|_{L \times [1,m]}$ each contain a complete $(2^{i+3} - 1)$-square; this is clearly possible, since $k, l, m \geqslant 2^{i+4}$, no matter which category (1)–(3) that P_1 and P_2 lie in. Let $c_1, c_2 \in \mathbb{N} \cap [1, m]$ be the central rows of such complete $(2^{i+3} - 1)$-squares (thus containing a single cross) in P_1' and P_2' respectively. Without loss of generality we may assume that $c_1 \leqslant c_2$ by switching the role of P_1' and P_2' if necessary. We claim that $c_1 \equiv c_2 \pmod{2^{i-1}}$. Suppose for a contradiction that this is not true. By Lemma 5.2, row $(c_2 + 2^{i-1})$ of P_2' must contain two consecutive crosses with separation $2^{\nu_2(2^{i-1})+1} = 2^i$. However, since $c_2 + 2^{i-1} = c_1 + (c_2 - c_1) + 2^{i-1}$, and since $(c_2 - c_1) + 2^{i-1}$ is not divisible by 2^{i-1}, hence neither by 2^i, Lemma 5.2 again shows that the same row in P_1' must contain two consecutive crosses with separation 2^j for $j := \nu_2((c_2 - c_1) + 2^{i-1}) + 1 < i$. By the argument above, the synchronization layer then clearly yields a contradiction (diagonals would force the same separation on row $(c_2 + 2^{i-1})$ in P_1' and P_2'). This means that for any $s \in \mathbb{N} \cap [1, m]$ not equal to c_1 (and therefore also not equal to c_2) modulo 2^{i-1}, any pair of consecutive crosses in row s of either P_1 or P_2 have to occur at the same constant separation $2^{\nu_2(c_1-s)+1}$, completing the proof. $\qquad\square$

Corollary 5.4. *For any $k, l, m \geqslant 2^{i+4}$ ($i \in \mathbb{N}$), any pattern P locally admissible for the board layer with cuboid shape $[1, k] \times [1, l] \times [1, m]$, and any $j \in \mathbb{N} \cap [1, i-1]$, there*

exists an $\vec{e}_1\vec{e}_2$-cross section of P which consists of a subpattern of a rectangulation of $\mathbb{Z}^2$ by aligned squares of size 2^j.

The function of the square boards is to provide the infrastructure for the remaining two layers, which we will now informally describe. The first of these, called the **construction layer**, uses accelerated Turing machines to construct prefixes of the characteristic Sturmian sequence $c_{\alpha'}$. Roughly speaking, the construction layer uses two main components, each of which, inside each finite (or infinite) board, simulates row-by-row the run of an accelerated Turing machine whose space and runtime is confined by the size of the boards. In other words, given an $\vec{e}_1\vec{e}_2$-cross section with square boards of size 2^j ($j \in \mathbb{N}$), computation is initialized in the starting state with empty tapes (seen as rows on the respective sublayers, all initially filled with blanks) of length 2^j and all read-write heads positioned on the leftmost symbol of each tape on the bottom row of each such square board. As described in Section 4, there are also control and auxiliary layers, which are initialized at the bottom of each board with the constant sequence of some initial transition tuple and the constant sequence of all blanks respectively. Each subsequent row, again the superposition of all tapes, the control and auxiliary sublayers, then contains the consecutive instantaneous description of the Turing machine run, determined by purely local rules, until the top of the board is reached after the simulation of exactly $2^j - 1$ steps, at which point the computation is stopped and reinitialized on the next board's bottom row. As we have seen, locally admissible cuboid patterns contain $\vec{e}_1\vec{e}_2$-cross sections with boards of varying sizes and thus both the length of the tapes of our Turing machines and the number of computational steps, though finite on each individual board size, are globally unbounded in points in X'.

The final layer, called the **pruning layer**, runs a simple checking procedure to match the prefixes of $c_{\alpha'}$ produced by the construction layer against subwords of the 0/1-sequences seen in the base layer to ensure that the latter have roughly the correct proportion α' of 1 symbols. This is slightly different than the procedure in [9] where the authors controlled the frequency α' of 1s by forcing the base sequences to have a regular (binary) Toeplitz structure – which is not enough for our purposes. The reason why we need a more restrictive construction is again the desired upgradability property of X''; a regular Toeplitz sequence on $\{0, 1\}$ with frequency α' of 1 symbols has the property that for any $j \in \mathbb{N}$, there exist subwords of length $m \in \mathbb{N}$ exponential in j containing $m\alpha' + j$ 1 symbols. This is unsatisfactory in our setting, as it would yield the existence of a locally (in fact globally!) admissible pattern in X' on a rectangular prism with shape $[1, m]^3$ with more than $m^3\alpha' + jm^2$ symbols 1 on the base layer, which would mean that $\left|\mathcal{L}^{\text{loc}}_{m,m,m}(X'')\right| \geqslant e^{\text{htop}(X'')m^3 + j \log M \cdot m^2}$, and since j was arbitrary, this would preclude upgradability of X''. This intrinsic and quintessential problem is obviated by instead checking the base against a Sturmian sequence, which is balanced: for any m, the number of 1 symbols within a subword of length m can take only two different values in $\mathbb{N}_0 \cap [m\alpha' - 1, m\alpha' + 1]$.

Local rules, as explained in detail in Section 4, are used to define both the construction and pruning layers uniformly on all $\vec{e}_1\vec{e}_2$-cross sections. These rules thus apply both to cross sections which see subpatterns of square griddings of $\mathbb{Z}^2$ on the board layer and to the (possible) exceptional cross sections containing incomplete and/or infinite boards where there is no left end of the tapes or no bottom row to initialize the computation. On such exceptional cross sections our rules may in fact allow the construction and pruning layers to contain strange, not well-controlled

configurations. Luckily, in any locally admissible pattern on a large rectangular prism, the combined volume (according to Lemma 5.3) of all such exceptional cross sections is so small that this does not affect the desired upgradability property of X' (and X'').

We begin with an informal description of the construction layer. Roughly speaking, it will be based on two accelerated Turing machines working together: The first TM can be more or less considered a black box consecutively producing the partial quotients $(a_n)_{n\in\mathbb{N}}$ of the continued fraction expansion of α', while the second TM, whose job is to construct longer and longer prefixes of the characteristic Sturmian sequence $c_{\alpha'}$, with lengths increasing by one each TM step, can be described in full detail. To do this, the second machine emulates the recursive construction from (2) of Facts 2.6; each time w_{n-1} is completed, it continues appending copies of w_{n-1} until reaching a total of a_n copies, after which it appends a copy of w_{n-2}, completing w_n. For this the second machine clearly needs access to each a_n in time to know that it has completed the desired number of copies of w_{n-1} and should append a copy of w_{n-2}, which must be begun in step $a_n \cdot t_{n-1} + 1$. Note that Definition 1.5 precisely implies that a_n can be computed by this time by the first TM.

Unfortunately, there could be a slight complication in this process if the sequence of "computation times" of the partial quotients is very irregular. For example, suppose that for some strictly increasing sequence $(n_i)_{i\in\mathbb{N}}$ of positive integers, our accelerated black-box TM can generate the successive partial quotients $a_{n_i}, a_{n_i+1}, \ldots, a_{n_{i+1}-2}$ very fast, say almost one per time step, while each one of the sparse partial quotients $a_{n_{i+1}-1}$ $(i \in \mathbb{N})$ takes a very long time to compute, always finishing just before time step $a_{n_{i+1}-1} \cdot t_{n_{i+1}-2} + 1$. For such a sequence $(a_n)_{n\in\mathbb{N}}$ the bound in computability condition $(\mathbf{C})$ would still be satisfied. However, the "requests" for $a_{n_i}, a_{n_i+1}, \ldots, a_{n_{i+1}-2}$ from the second TM would yet come at a very regular rate, i.e. for every $N \in \mathbb{N}$, a_N is asked for between step $a_{N-1} \cdot t_{N-2} + 1$ and $a_N \cdot t_{N-1}$. So at the time when (for instance) a_{n_i} is requested, the first TM might be already in the middle of the (long) computation of $a_{n_{i+1}+1}$, and no longer have access to a_{n_i}.

Note that in such a situation it is neither possible to simply hold the computation of the next partial quotient until the present one was requested nor to store all of the yet unrequested partial quotients on an additional tape, as both their size and their number (growing gaps between n_i's) might be unbounded, which would not allow us to copy them back in a single time step. Nevertheless, using the capabilities of an accelerated TM, it is still possible to store a single bit of information about every one of them (which thus can be transmitted over arbitrarily long tape distances in one time step). This allows the computation times of the first Turing machine to be "regularized", finally leading to a solution for our problem.

We now give a formal description of the **construction layer**. Its first component consists of layers implementing the run of an accelerated Turing machine T_1 successively generating the partial quotients $(a_n)_{n\in\mathbb{N}}$ of the continued fraction expansion of α', where for each $N \in \mathbb{N}$ all $a_1, a_2, \ldots, a_N$ are finished before time step $a_N \cdot t_{N-1} + 1$. The existence of T_1 is guaranteed by the assumption that α' satisfies computability condition (C). Every time an a_n is completed, the machine checks the status of a tape, called the request-tape, which is shared with the second component (see below). If its leftmost cell contains a symbol "?", signaling

a request, T_1 copies the binary representation of a_n onto this shared request-tape using an instant operation. If there is no request present on the shared tape, T_1 just examines the value of a_n and copies one bit of information onto an additional tape (which we call the $1/*$-**tape**): this bit will be either a 1 if $a_n = 1$, or a $*$ if $a_n > 1$. The transfer of each such bit is done via an instant operation sending the information to the right end of the already used part of the $1/*$-tape where the bit is put down by the read-write head which then moves one cell further right, thus always staying at the right end of the occupied tape segment. Therefore, at any point throughout the construction, the $1/*$-tape (initially starting off with all blanks and the read-write head on its leftmost cell) acts like a queue containing a simplified record of the already produced (but not yet used) partial quotients a_n.

Though a bit inelegant, for organization's sake, we need a few facts about the second component (which will be proved below) in order to completely describe the first component's behavior. We will for now take on faith (and soon verify) the following facts: Firstly, the second component will simulate an accelerated TM T_2 which creates prefixes of the characteristic sequence for α', at the rate of one symbol per time step. Secondly, T_2 will send "requests" to the first component asking for a partial quotient a_N, each request being sent at time step $a_{N-1} \cdot t_{N-2} + 1$. These "requests" are communicated via instant operations acting on a tape, called **request-tape**, shared by the two components, in a way which will be described in a moment. Also we add another tape, called **counter-tape**, to the second TM T_2, whose job is to store a binary representation of the index N of the currently requested partial quotient a_N. Starting from zero the value on this counter-tape is incremented by one every time a "request" is sent by T_2, i.e. at time steps $a_{N-1} \cdot t_{N-2} + 1$. Now each request triggers the following instant operations: First the leftmost symbol on the $1/*$-tape, corresponding to the bit of information stored about the partial quotient a_N, is examined. If it is a blank symbol (and thus the read-write head is positioned at the leftmost cell), then a_N has not yet been computed by T_1, and so nothing needs to be done; the request-tape stays unchanged and T_1 will compute a_N and copy it to the request-tape before time step $a_N \cdot t_{N-1} + 1$ by the definition of computability condition **(C)**, in time to be picked up by the second component. If the examined symbol of the $1/*$-tape is a 1 or a $*$, the complete content of the $1/*$-tape (including the read-write head's position) is shifted one cell to the left deleting the leftmost symbol. In case of a 1, an instant operation is then used to copy this symbol directly to the shared request-tape (overwriting the "?"), where it will be available when needed by the second component. If the examined symbol from the $1/*$-tape is a $*$, then we know that $a_N \geqslant 2$. This gives the machine T_1 enough time to invoke a subroutine given as another accelerated TM T_1', run in parallel on separate layers. This TM T_1' instantly deletes the request-tape and then simulates a version of T_1, enhanced by a linear speed-up by a factor of 2, to recompute all partial quotients up to a_N. (Such a speed-up is easily implemented by creating a version of T_1 which executes two steps of computation of the original TM T_1 in a single step by modifying its transition function or by using twice the number of tapes/sublayers.) Note that (for $N \geqslant 3$) the time difference between the request for a_N at step $a_{N-1} \cdot t_{N-2} + 1$ and the time by which it has to be ready, namely $a_N \cdot t_{N-1}$, satisfies: $\Delta t := a_N \cdot t_{N-1} - (a_{N-1} \cdot t_{N-2} + 1) \geqslant a_N \cdot t_{N-1} - (a_{N-1} \cdot t_{N-2} + t_{N-3}) = (a_N - 1)t_{N-1}$. Since $a_N \geqslant 2$, this yields $\frac{\Delta t}{a_N \cdot t_{N-1}} \geqslant \frac{a_N - 1}{a_N} = 1 - \frac{1}{a_N} \geqslant \frac{1}{2}$. Therefore the subroutine executed by T_1' can proceed as follows: it will copy the

current binary value on the counter-tape holding the index N to one of its own tapes using an instant operation and then will use the twice-as-fast version of T_1 to recompute $a_1, \ldots, a_N$. Whenever the twice-as-fast version of T_1 produces a partial quotient, it decrements the value of the copied counter. As long as this value is bigger than zero, the partial quotient is discarded and computation continues, while once it has reached zero the corresponding partial quotient, i.e. a_N, is instantly copied to the shared request-tape. Since the computation time of a_N in this twice-as-fast subroutine is at most $\frac{1}{2} a_N \cdot t_{N-1}$, this happens before time step $a_N \cdot t_{N-1}$ as claimed. In particular the computation of the subroutine finishes before the next request arrives, and thus the relevant tapes/layers of T_1' will be free if T_1' is required to restart for a future a_n.

To be clear, what we have shown so far is that if, for any $N \in \mathbb{N}$, the second component sends the request for a_N via the shared tape at step $a_{N-1} \cdot t_{N-2} + 1$, then the described accelerated TM(s) T_1 (and T_1') will in an instant operation copy the requested partial quotient a_N to the shared tape after the request, but not after step $a_N \cdot t_{N-1}$. It remains to describe the second component, and to show that if it receives a_N between time steps $a_{N-1} \cdot t_{N-2} + 1$ and $a_N \cdot t_{N-1}$, then it will send the next request at time step $a_N \cdot t_{N-1} + 1$ while producing longer and longer prefixes of the sequence $c_{\alpha'}$ (row $r \in \mathbb{N}$ of each big enough board will contain $c_{\alpha'}|_{[0,r-1]}$).

The second component starts from a single 0 or 1 in the lower left corner (the remainder of this row is filled with blanks). Each subsequent row repeats the finite word of 0s and 1s seen in the row below, and appends a single new symbol 0 or 1 to its right end according to certain rules, keeping the blanks thereafter. This process continues until the top row of a finite board is reached, which is completely filled with a prefix of $c_{\alpha'}$. To be a bit more specific, the accelerated TM T_2 has exactly 4 tapes and thus is implemented by local rules on 6 sublayers (the 4 tapes, along with a control sublayer and an auxiliary sublayer). It starts off (at the bottom of a finite board) with all 4 read-write heads at the left end and tapes 1 to 3 filled with a symbol 0 followed by all blanks while tape 4, which will be used to record the prefixes of $c_{\alpha'}$, begins with a single 0 or 1 (the first symbol of $c_{\alpha'}$, which is a 0 if $a_1 > 1$ and a 1 if $a_1 = 1$) followed by all blanks. Following a hard-wired procedure, the TM T_2 then runs for $a_1 - 1 \geqslant 0$ steps, putting down symbol-by-symbol the remainder of the word $w_1 = 0^{a_1-1}1$ on tape 4, moving the head one cell to the right in each step while the other tapes and head positions are not changed. Once finished with this part, T_2 enters an infinite loop, repeating for each value of $N \in \mathbb{N}$ ($N \geqslant 2$) the following steps to recursively produce the words w_N (see Figure 5):

Phase 1: A combined instant operation copies the complete content of tape 3 to tape 2 and the complete content of tape 4 to tape 3, placing the heads on both tapes 2 and 3 at their left end. The same instant operation fills tape 1 with a 1 followed by all blanks (i.e. puts a binary 1 while keeping its head at the left end) and then compares this value against the content of the shared request-tape, which either contains a "?" (request) or the binary value of the partial quotient a_N. The result of this comparison steers the behavior of T_2 during an inner loop: if the two quantities are unequal (this includes the case where the request tape contains a "?"), it moves to Phase 2; if they are equal, it moves to Phase 3.

Note that after the instant operation from Phase 1, the content of the tapes is as follows: tapes 3 and 4 both contain a copy of the word w_{N-1}, tape 2 contains a copy of the word w_{N-2}, and tape 1 contains the value 1 in binary (here $N \in \mathbb{N}$

starting from $N = 2$ increases with every completed cycle of the infinite loop).

Tape 4:	Tape 3:	Tape 2:	Tape 1:
⋮ ⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮
$0100100100101001001001010\overline{0}$	$\overline{0}100100100101\ldots10$	$\overline{0}1001\ldots01$	$\overline{1}$
$010010010010100100100101\overline{0}$ = w_4	$01\overline{0}$	$01\overline{0}$	$\overline{0}1$
$01001001001010010010010\overline{1}$	$0\overline{1}0$	$0\overline{1}0$	$\overline{0}1$
$0100100100101001001001\overline{0}$	$\overline{0}10$	$\overline{0}10$	$\overline{0}1$ = a_4
$010010010010100100100\overline{1}$	$0100100100\overline{1}$	$\overline{0}10$	$\overline{1}$
$01001001001010010010\overline{0}$	$010010010\overline{0}1$	$\overline{0}10$	$\overline{1}$
$0100100100101001001\overline{0}$	$01001001\overline{0}01$	$\overline{0}10$	$\overline{1}$
$010010010010100100\overline{1}$	$0100100\overline{1}001$	$\overline{0}10$	$\overline{1}$
$01001001001010010\overline{0}$	$010010\overline{0}1001$	$\overline{0}10$	$\overline{1}$
$0100100100101001\overline{0}$	$01001\overline{0}01001$	$\overline{0}10$	$\overline{1}$
$010010010010100\overline{1}$	$0100\overline{1}001001$	$\overline{0}10$	$\overline{1}$
$01001001001010\overline{0}$	$010\overline{0}1001001$	$\overline{0}10$	$\overline{1}$
$0100100100101\overline{0}$	$01\overline{0}01001001$	$\overline{0}10$	$\overline{1}$
$010010010010\overline{1}$	$0\overline{1}001001001$	$\overline{0}10$	$\overline{1}$
$01001001001\overline{0}$	$\overline{0}1001001001$	$\overline{0}10$	$\overline{1}$ ≠ a_4
$0100100100\overline{1}$ = w_3	$0\overline{1}0$	$0\overline{1}$	$\overline{1}1$
$010010010\overline{0}$	$\overline{0}10$	$\overline{0}1$	$\overline{1}1$ = a_3
$01001001\overline{0}$	$0\overline{1}0$	$\overline{0}1$	$\overline{0}1$
$0100100\overline{1}$	$\overline{0}10$	$\overline{0}1$	$\overline{0}1$
$010010\overline{0}$	$01\overline{0}$	$\overline{0}1$	$\overline{0}1$ ≠ a_3
$01001\overline{0}$	$0\overline{1}0$	$\overline{0}1$	$\overline{1}$
$0100\overline{1}$	$\overline{0}10$	$\overline{0}1$	$\overline{1}$
$010\overline{0}$	$0\overline{1}$	$\overline{0}$	$\overline{1}$ ≠ a_3
$01\overline{0}$ = w_2	$\overline{0}1$	$\overline{0}$	$\overline{1}$ = a_2
$0\overline{1}$ = w_1	$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$

FIGURE 5. First steps of the run of the accelerated Turing machine from the second component of the construction layer for an irrational $\alpha' = [0; 2, 1, 3, 2, 5, \ldots]$. (On all 4 tapes blanks are omitted and the read-write head's position is marked by an overline.)

Phase 2: If the comparison from Phase 1 was unsuccessful, i.e. the content of tape 1 did not (yet) match the content of the request-tape, then at least one more copy of w_{N-1} is needed to complete a prefix of w_N. To accomplish this, the content of tape 3 (i.e. w_{N-1}) is appended to the end of the word present on tape 4, moving the head on both tapes to the right and sending one symbol at a time across (using the auxiliary sublayer). This takes $|w_{N-1}| = t_{N-1}$ (standard TM) steps.

When finished, another instant operation sends the read-write head on tape 3 back to the tape's left end and, to make note of the newly completed copy of w_{N-1}, the value on tape 1 is incremented by 1 (addition of 1 in binary) and again compared against the content of the shared request-tape. If the comparison is successful, the accelerated TM T_2 moves to Phase 3, otherwise it repeats Phase 2.

After a finite number of iterations (namely $a_N - 1$) of Phase 2, the value on tape 1 will have reached a_N. By this time, i.e. precisely at the beginning of step $a_N \cdot t_{N-1} + 1$, the shared request-tape already has to contain the binary representation of the partial quotient a_N produced by the first component's TM T_1, so the comparison will be successful and T_2 exits the inner loop and enters Phase 3. (Note that this inner loop can not finish earlier, as in all previous iterations the strictly increasing value on tape 1 has not yet reached a_N, always resulting in an unsuccessful comparison against a_N or "?".)

Phase 3: An instant operation deletes the content of the request-tape and replaces it with a "?" to request the next partial quotient a_{N+1}. Also the value on the shared counter-tape holding the binary representation of N is incremented by 1 in this instant operation. Note that after a successful comparison tape 4 exactly contains the word $w_{N-1}{}^{a_N}$. To complete it to w_N, all that remains to do is to append a copy of w_{N-2}, which is done by copying the content of tape 2 one symbol at a time with the same technique as above, which takes $|w_{N-2}| = t_{N-2}$ (standard TM) steps. This concludes the procedure generating w_N precisely in step t_N and starts the next cycle of the TM's infinite loop, i.e. returns it to Phase 1.

Finally, we note that with this procedure, the second component will indeed send the "requests" at the desired times, as long as provided with the partial quotients a_N between request times. This means that all components work together as desired, and in particular that the second component will fill the entire top row of any complete board with a prefix of the characteristic Sturmian sequence for α'.

We now superimpose a last layer, called the **pruning layer**, whose purpose is to compare the prefixes of the characteristic Sturmian sequence $c_{\alpha'}$ generated on the top row of each (finite) square board against subwords of the unique 0/1-sequence seen in all rows of the base layer. The pruning layer will ensure that the number of 1s in both such words and all their subwords having the same length differ by at most one, thus restraining the 0/1-base sequence to be "almost" Sturmian. This is achieved by using two sublayers with very simple local rules: The first sublayer, called the **prefix sublayer**, consists of 0s and 1s and is constant along the $\vec{e}_1 + \vec{e}_2$-direction. In addition on each $\vec{e}_1\vec{e}_2$-cross section containing finite boards its symbols are determined by the symbols from the construction layer seen on the top row of those boards, i.e. whenever a site is part of the top row of a square board, a local rule forces the symbols present at this site on the prefix sublayer and the sublayer containing tape 4 of the TM T_2 in the construction layer to coincide. Since all square boards in a $\vec{e}_1\vec{e}_2$-cross section have the same size, are aligned, and have the same 0/1-word on their top rows, this additional rule is consistently well defined. The second sublayer, called **checking sublayer**, uses the alphabet $\{-1, 0, +1\}$ and along every row implements a running total of the difference between the number of 1s in certain subwords of length 2^j ($j \in \mathbb{N}$) of the two 0/1-sequences seen in the prefix sublayer and the base layer. The local rules for the checking sublayer are specified as follows: At every site $\vec{\imath} \in \mathbb{Z}^3$ where the synchronization sublayer contains a diagonal symbol $\square$ the checking layer (re-)starts its count with a symbol $c_{\vec{\imath}} := b_{\vec{\imath}} - p_{\vec{\imath}} \in \{-1, 0, +1\}$, while on all other sites $\vec{\imath}$ the checking layer has to contain the symbol $c_{\vec{\imath}} \in \{-1, 0, +1\}$ corresponding to the result of the sum $c_{\vec{\imath}} := b_{\vec{\imath}} - p_{\vec{\imath}} + c_{\vec{\imath}-\vec{e}_1}$, where $b_{\vec{\imath}} \in \{0, 1\}$ is the symbol from the base layer, $p_{\vec{\imath}} \in \{0, 1\}$ is the symbol on the prefix sublayer and $c_{\vec{\imath}-\vec{e}_1} \in \{-1, 0, +1\}$ is its left neighbor on the checking sublayer. Since the only legal symbols in the checking sublayer are -1, 0, and $+1$, this implies that – whenever the pruning layer can be legally filled in all of a point in X' – the difference in the number of 1 symbols has to be within ± 1 for arbitrary lengths 2^j. In fact, the construction implies the same for any finite subword of length $n \in \mathbb{N}$ of the 0/1-sequence present in the base layer: its number of 1s has to be within ± 1 of the number of 1s of a length-n prefix of the characteristic Sturmian sequence $c_{\alpha'}$, hence has to fall in the range $\mathbb{N}_0 \cap [n\alpha' - 2, n\alpha' + 2]$.

This completes the description of X', and we now begin to prove the properties of X' which will eventually imply the upgradability of its extension X''.

Lemma 5.5. *There exists a constant $C' \in \mathbb{R}$ so that for any dimensions $k, l, m \in \mathbb{N}$, the number of locally admissible patterns satisfies $\left| \mathcal{L}^{\mathrm{loc}}_{k,l,m}(X') \right| \leqslant e^{C'(kl+km+lm)}$.*

Remark 5.6. We note that this clearly implies that $\mathrm{h}_{\mathrm{top}}(X') = 0$, and so this is just the upgradability condition for X' as given in Definition 5.1.

Proof. As the total number of locally admissible patterns on any fixed shape $F \subsetneq \mathbb{Z}^3$ in X' is bounded from above by the product of the numbers of locally admissible patterns of shape F on each layer, it suffices to show the claimed inequality separately for each layer. Also note that if a layer has the property that each locally admissible pattern on a rectangular prism is completely determined by the symbols on its boundary, this immediately implies the desired inequality for this layer.

Due to constantness along one direction, any locally admissible pattern with cuboid shape on the base layer, the board layer (consisting of the two Robinson sublayers and the synchronization sublayer) and the prefix sublayer is forced by its boundary symbols. Moreover, given the base, synchronization, and prefix (sub)layers, the checking sublayer is uniquely determined as well. Hence we are already done with the base, board and pruning layers of X'.

The only remaining layer is the construction layer, whose analysis is a bit more complicated, because although it is entirely deterministic on any complete (finite) square board, it is not obvious what happens on infinite and/or incomplete boards. Consider any dimensions $k, l, m \in \mathbb{N}$, choose $i \in \mathbb{N}_0$ such that $2^i \leqslant \min\{k, l, m\} < 2^{i+1}$ and let $P \in \mathcal{L}^{\mathrm{loc}}_{k,l,m}(X')$ be any locally admissible cuboid pattern in X'. W.l.o.g. we may assume i to be large ($i > 5$), as for small i the constant C' trivially exists.

Then by Lemma 5.3, there are at most $\left\lceil \frac{m}{2^{i-5}} \right\rceil \leqslant \frac{2^5 \cdot m + 2^i}{2^i} \leqslant \frac{33m}{2^i} < \frac{66m}{\min\{k,l,m\}}$ exceptional $\vec{e}_1\vec{e}_2$-cross sections of P whose board layers might not consist of subpatterns of square griddings of $\mathbb{Z}^2$, while all other $\vec{e}_1\vec{e}_2$-cross sections of P are regular, containing a subpattern of aligned finite square boards of a certain size. First consider any non-exceptional $\vec{e}_1\vec{e}_2$-cross section $P|_{[1,k] \times [1,l] \times \{s\}}$ ($s \in \mathbb{N} \cap [1, m]$) with a regular configuration on the board layer. As already mentioned, the construction layer is entirely determined on each complete finite square board which is filled by the instantaneous descriptions of the TM runs. We also claim that for any incomplete board, the construction layer's configuration within a partial board is completely determined by the symbols on the boundary of every such partial board. This is not difficult to see: the accelerated Turing machines defining the construction layer are mostly deterministic; as long as one has perfect information about a complete instantaneous description (row) and the TM's transitions, the whole future evolution of the TM run (rows above, i.e. along the vertical direction) within the board is forced. This is even true if the construction layer's first component T_1 is a non-deterministic accelerated TM. In this case we simply force constantness along rows in its control sublayer on all non-exceptional $\vec{e}_1\vec{e}_2$-cross sections containing diagonals ▱. Thus the non-deterministic evolution of the TM runs have to coincide across all horizontally aligned square boards and knowledge of the symbols on the boundary of the locally admissible pattern P again forces determinism. The difficulty with a partial board is that we might only see part of the (infinite) tape and that the read-write head may "disappear" at the right and/or left end of an incomplete row. However, due to the construction using tape, control and auxiliary sublayers, information about the TM's current state, movements of the read-write heads and possible instant operations is available everywhere along a row. If we

are given the symbols on the entire boundary of the incomplete board, the bottom most (partial) row gives us a (partial) instantaneous description to start from and via the symbols along the right- and leftmost columns of the partial board we then know all rows where the read-write head exits and re-enters, so again the behavior on the interior is determined uniquely. Finally, again through the complete distribution of information along rows in the control sublayer, given the boundary, we know in which rows (if any) instant operations are performed and how they act on the partial tape's instantaneous description.

This means that on any non-exceptional $\vec{e}_1\vec{e}_2$-cross section, the construction layer on the entire cross section is determined uniquely by knowledge of the symbols from the finite alphabet $\mathcal{A}_c$ of the construction layer on the union of the boundaries of all partial boards. Regardless of the board size, the union of those boundaries traces the perimeter of the entire $\vec{e}_1\vec{e}_2$-cross section at most four times, and so has size not more than $4 \cdot (2k + 2l)$. The total number of ways to choose these symbols in all of P, given a particular assignment of the base, board and pruning layers, is then bounded from above by $|\mathcal{A}_c|^{8km+8lm}$. We must only now account for the (possible) exceptional cross sections from Lemma 5.3, which have combined volume bounded from above by $\frac{66m}{\min\{k,l,m\}}kl$. By combining these bounds, we see that the number of ways to assign a construction layer to P, given the base, board and pruning layers, is bounded from above by $|\mathcal{A}_c|^{8km+8lm+66klm/\min\{k,l,m\}} \leqslant |\mathcal{A}_c|^{74(kl+km+lm)}$, completing the proof of the desired inequality for the construction layer and thus the whole proof of the lemma. $\qquad\square$

Lemma 5.7. *There exists a constant $D \in \mathbb{R}_0^+$ so that for any dimensions $k, l, m \in \mathbb{N}$ and any locally admissible pattern $P \in \mathcal{L}_{k,l,m}^{\mathrm{loc}}(X')$, the number of 1s seen in the base layer of P satisfies $\#_1(\pi_{\mathrm{base}}(P)) \leqslant \alpha'klm + D(kl + km + lm)$.*

Proof. Fix any $k, l, m \in \mathbb{N}$, and choose $i \in \mathbb{N}_0$ so that $2^i \leqslant \min\{k, l, m\} < 2^{i+1}$. Consider an arbitrary pattern $P \in \mathcal{L}_{k,l,m}^{\mathrm{loc}}(X')$. Since the base layer of P is constant along the $\vec{e}_3$-direction, we can define $Q := \pi_{\mathrm{base}}(P)|_{[1,k]\times[1,l]\times\{1\}} \in \{0,1\}^{[1,k]\times[1,l]}$, the pattern consisting of the common base layer of P on every $\vec{e}_1\vec{e}_2$-cross section.

We first note that since the dimensions of P are $k, l, m \geqslant 2^i$ (again assume $i > 5$ while using the trivial bound with $D := 2^{5+1} = 64$ for $i \leqslant 5$), by Corollary 5.4 for any $j \in \mathbb{N} \cap [1, i-5]$ there exists a (non-exceptional) $\vec{e}_1\vec{e}_2$-cross section of P whose collective board size is 2^j.

Consider an $\vec{e}_1\vec{e}_2$-cross section of the board layer of P which consists of a subpattern of a gridding of $\mathbb{Z}^2$ by square boards of size 2^{i-5}. Since $k, l \geqslant 2^i$, there are several complete square boards in this cross section. In particular the cross section contains a row $r \in \mathbb{N} \cap [1, l]$ sitting in between the top rows of two horizontal strips of aligned complete boards such that a diagonal from the synchronization sublayer hits the left border of P exactly in row r (see Figure 6). This means that the 2^{i-5}-letter prefixes of the characteristic Sturmian sequence $c_{\alpha'}$ produced in the top rows of those complete square boards, i.e. above and below row r, are propagated along the $\vec{e}_1 + \vec{e}_2$-direction in the prefix sublayer and are eventually checked against 2^{i-5}-letter base layer subpatterns (words) on row r. Our earlier analysis of the pruning layer assures that those words have at most $\alpha' \cdot 2^{i-5} + 2$ 1s. The same overage of at most 2 holds true for shorter words starting from a diagonal on their left (as the checking sublayer is guaranteed to start the running total there with a 0). This lets us conclude that the number of 1s in row r is bounded by α' times its length k plus
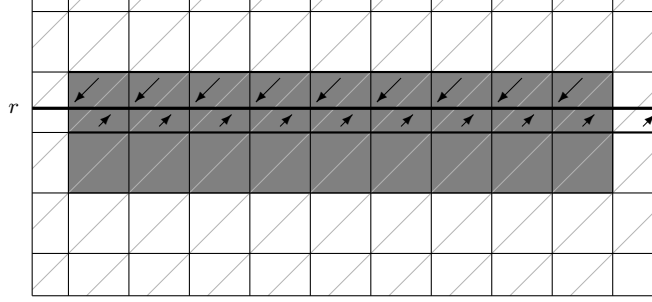
FIGURE 6. The pattern Q and its row r. (Two strips of complete boards of size 2^{i-5} in dark gray, diagonals from the synchronization sublayer in light gray.) Arrows indicate the propagation of the prefixes of $c_{\alpha'}$ produced on the top rows of (complete) boards which are then checked against subwords of the base layer in row r.

an overage of at most 2 for each subword seen between two consecutive diagonals: $\#_1(Q|_{[1,k] \times \{r\}}) \leqslant \alpha' \cdot k + 2 \lceil \frac{k}{2^{i-5}} \rceil$. As the rows in Q are constant and Q appears on every $\vec{e}_1 \vec{e}_2$-cross section in the base layer of P we get:

$$\#_1(\pi_{\text{base}}(P)) \leqslant \left(\alpha' k + 2 \lceil \tfrac{k}{2^{i-5}} \rceil \right) \cdot lm \leqslant \alpha' klm + 2^7 \cdot \tfrac{klm}{2^{i-1}} + 2lm$$
$$\leqslant \alpha' klm + 128(kl + km + lm) + 2lm$$

where we used $\min\{k, l, m\} < 2^{i+1}$. The lemma is thus proved for $D := 130$. $\qquad \square$

Step 2 (Constructing X''). We now wish to add entropy to X'. Define the partition $\mathcal{A}' =: \mathcal{A}'_0 \dot\cup \mathcal{A}'_1$ of its alphabet $\mathcal{A}'$, where $\mathcal{A}'_i$ is the set of elements of $\mathcal{A}'$ which see $i \in \{0, 1\}$ on the base layer. Then write $\mathcal{A}'_1 = \{b_1^{(1)}, \dots, b_I^{(1)}\}$ ($I := |\mathcal{A}'_1|$) and define disjoint copies $\mathcal{A}'_j := \{b_1^{(j)}, \dots, b_I^{(j)}\}$ for $2 \leqslant j \leqslant M$, to get the new alphabet $\mathcal{A}'' := \mathcal{A}'_0 \dot\cup \dot\bigcup_{j=1}^M \mathcal{A}'_j$. Finally, define a $\mathbb{Z}^3$ SFT $X'' \subseteq \mathcal{A}''^{\mathbb{Z}^3}$ on $\mathcal{A}''$ by the rule that a pattern on $\mathcal{A}''$ is in $\mathcal{L}(X'')$ if and only if replacing any symbol $b_i^{(j')} \in \dot\bigcup_{j=2}^M \mathcal{A}'_j$ by $b_i^{(1)} \in \mathcal{A}'_1$ ($1 \leqslant i \leqslant I$) yields a pattern in $\mathcal{L}(X')$, i.e. $X'' := \pi_\infty^{-1}(X')$ is the preimage of X' under the projection π_∞ induced by the 1-block map $\pi : \mathcal{A}'' \to \mathcal{A}'$, $b \mapsto \begin{cases} b & \text{if } b \in \mathcal{A}'_0 \\ b_i^{(1)} & \text{if } b = b_i^{(j)} \in \mathcal{A}'' \backslash \mathcal{A}'_0 \end{cases}$ collapsing back the newly introduced sets $\mathcal{A}'_j$ with $2 \leqslant j \leqslant M$ onto $\mathcal{A}'_1$. Informally, each symbol in $\mathcal{A}'_1$ is split into M independent copies to create points in X'' from a point in X' preserving SFTness.

Lemma 5.8. *There exists a non-negative real constant $C \in \mathbb{R}_0^+$ so that for any dimensions $k, l, m \in \mathbb{N}$, $\left| \mathcal{L}_{k,l,m}^{\text{loc}}(X'') \right| \leqslant e^{\alpha' \log M \cdot klm + C(kl + km + lm)}$.*

Proof. This follows from the definition of X'' together with Lemmata 5.5 and 5.7: the 1-block map π also induces a surjection $\pi_\infty^{\text{loc}} : \mathcal{L}_{k,l,m}^{\text{loc}}(X'') \to \mathcal{L}_{k,l,m}^{\text{loc}}(X')$ which for any $i \in \{1, 2, \dots, |\mathcal{A}'_1|\}$ changes all symbols $b_i^{(j)}$ with $j \in \{2, 3, \dots, M\}$, back to $b_i^{(1)}$. Then by Lemma 5.7, there exists D so that $\left| (\pi_\infty^{\text{loc}})^{-1}(P) \right| \leqslant M^{\alpha' klm + D(kl + km + lm)}$

holds for the fiber of any locally admissible pattern $P \in \mathcal{L}_{k,l,m}^{\mathrm{loc}}(X')$, and therefore

$$
\begin{aligned}
\left| \mathcal{L}_{k,l,m}^{\mathrm{loc}}(X'') \right| &\leqslant M^{\alpha' klm + D(kl+km+lm)} \left| \mathcal{L}_{k,l,m}^{\mathrm{loc}}(X') \right| \\
&\leqslant e^{\alpha' \log M \cdot klm + (D \log M + C')(kl+km+lm)}
\end{aligned}
$$

by Lemma 5.5. This proves the lemma for $C := D \log M + C' \in \mathbb{R}_0^+$. $\qquad\square$

Proposition 5.9. *The entropy of X'' satisfies $\mathrm{h}_{\mathrm{top}}(X'') = \alpha' \log M = \alpha$.*

Proof. It is clear from Lemma 5.8 that $\mathrm{h}_{\mathrm{top}}(X'') \leqslant \lim_{m \to \infty} \frac{\log \left| \mathcal{L}_{m,m,m}^{\mathrm{loc}}(X'') \right|}{m^3} = \alpha$. It remains to show that $\mathrm{h}_{\mathrm{top}}(X'') \geqslant \alpha$. To see this, consider any Sturmian sequence $y \in \{0,1\}^{\mathbb{Z}}$ obtained from coding the complete $\mathbb{Z}$-orbit of a point in the circle under the rotation $R_{\alpha'}$ by the fixed irrational α', and construct a point $x' \in X'$ by putting y in each row of the base layer of x' and filling the remaining layers as dictated by the SFT rules of X'. It is fairly easy to see that this can be done to create a valid point in X': the only layer which could cause a problem is the pruning layer, but this will always complete successfully; since Sturmian sequences are balanced, finite subpatterns of y (from the base layer) and the prefixes of $c_{\alpha'}$ (produced at the top of any complete board in the construction layer) which are compared against each other will have numbers of 1s within one of each other.

Then, since the overall frequency of symbols from $\mathcal{A}_1'$ in x' is equal to α', for any $m \in \mathbb{N}$ there exists a subpattern $P_m \in \mathcal{L}_{m,m,m}(X')$ of x' on a rectangular prism $[1,m]^3$ with a proportion of at least α' $\mathcal{A}_1'$-letters. The fiber of P_m under π_∞ then yields at least $M^{\alpha' m^3}$ patterns in $\mathcal{L}_{m,m,m}(X'')$ by independently changing each symbol $b_i^{(1)} \in \mathcal{A}_1'$ ($i \in \{1,2,\ldots,|\mathcal{A}_1'|\}$) in P_m into one of its M copies $b_i^{(j)} \in \mathcal{A}_j'$ ($j \in \{1,2,3,\ldots,M\}$), and so $\mathrm{h}_{\mathrm{top}}(X'') = \lim_{m \to \infty} \frac{\log \left| \mathcal{L}_{m,m,m}(X'') \right|}{m^3} \geqslant \alpha' \log M = \alpha$. $\qquad\square$

Lemma 5.8 and Proposition 5.9 obviously yield the following final corollary.

Corollary 5.10. *X'' is an upgradable $\mathbb{Z}^3$ shift of finite type with $\mathrm{h}_{\mathrm{top}}(X'') = \alpha > 0$.*

Step 3 (Constructing X)**.** The last step of our proof is based on the following general result, which, together with Steps 1 and 2 obviously implies Theorem 1.7.

Theorem 5.11. *For any upgradable $\mathbb{Z}^3$ shift of finite type Z with $\mathrm{h}_{\mathrm{top}}(Z) > 0$, exists a block gluing $\mathbb{Z}^3$ shift of finite type $\widetilde{Z}$ containing Z with $\mathrm{h}_{\mathrm{top}}(\widetilde{Z}) = \mathrm{h}_{\mathrm{top}}(Z)$.*

Remark 5.12. The same result also holds for $\mathbb{Z}^2$ SFTs (with a similar definition of upgradability for such SFTs). This basically follows from adopting our argument below to the wire symbols from [22] instead of the wall symbols introduced in Figure 7. In the following proof, the reader may thus first think of the wire shift as a two-dimensional "toy model", which might make it easier to follow the argument.

Proof. Suppose $Z \subseteq \mathcal{A}^{\mathbb{Z}^3}$ is a $\mathbb{Z}^3$ SFT with alphabet $\mathcal{A}$ which is upgradable, with constant $C \in \mathbb{R}_0^+$ such that $\left| \mathcal{L}_{k,l,m}^{\mathrm{loc}}(Z) \right| \leqslant e^{\mathrm{h}_{\mathrm{top}}(Z)klm + C(kl+km+lm)}$ for all $k,l,m \in \mathbb{N}$.

As a first step we define a new alphabet $\widetilde{\mathcal{A}} := \mathcal{A} \,\dot\cup\, \mathcal{A}_w$ where $\mathcal{A}_w$ is the alphabet of the 65 unit-cube symbols specified in Figure 7 – actually there are 39 "wall symbols" plus 26 additional symbols called "shadow symbols", but nevertheless we will refer to the whole set $\mathcal{A}_w$ as the wall alphabet[7]. The 26 distinct shadow

---

[7]Readers familiar with the wire shift [22] will recognize this as a three dimensional version of the alphabet consisting of the 6 wire symbols plus $26 = 3^3 - 1$ (in two dimensions only $8 = 3^2 - 1$) types of blank symbols in this context called shadow symbols.

symbols are used to fill in the collar left over when putting a locally admissible pattern $P \in \mathcal{L}^{\mathrm{loc}}_{k,l,m}(Z)$ $(k, l, m \in \mathbb{N})$ in the center of a larger rectangular prism being surrounded by wall symbols. This larger prism is subdivided into $27 = 3 \cdot 3 \cdot 3$ smaller prisms by the 6 planes supporting one of the 6 faces of the central prism and each of the 26 non-central subprisms is to be filled with shadow symbols of one particular type "$S_{\vec{c}}$" where $\vec{c} \in \{-1, 0, +1\}^3 \backslash \{(0,0,0)\}$ indicates the position of the subprism relative to the central one. For the sake of simplifying notation we sometimes use arrows "↑" pointing in the 6 principal orientations $\pm \vec{e}_i$ $(i \in \{1,2,3\})$ to denote the six shadow symbols "$S_{\pm \vec{e}_i}$" (see also Figure 7).
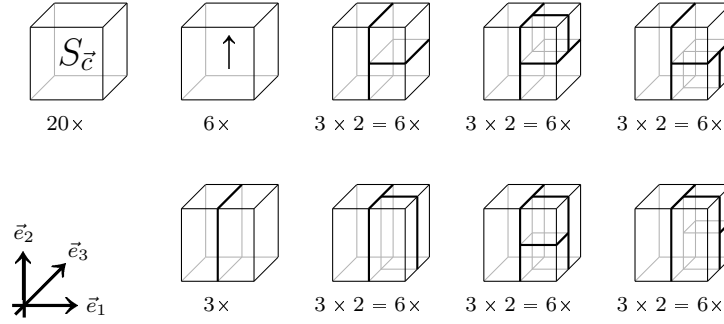

FIGURE 7. The 65 symbol wall alphabet $\mathcal{A}_w$ consisting of the 26 "shadow symbols" (marked either with the letter "$S_{\vec{c}}$" for $\vec{c} \in \{-1, 0, +1\}^3$ with $\|\vec{c}\|_1 = \sum_{i=1}^3 |\vec{c}_i| \geqslant 2$ or with an arrow having one of the six principal orientations) and the 39 "wall symbols" (counted with multiplicity: the principal, i.e. largest, wall segment may be oriented parallel to the $\vec{e}_1\vec{e}_2$-, $\vec{e}_1\vec{e}_3$- or $\vec{e}_2\vec{e}_3$-plane and all wall symbols having additional wall segments may then be reflected about this principal segment).

Now we define our $\mathbb{Z}^3$ SFT $\widetilde{Z} \subseteq \widetilde{\mathcal{A}}^{\mathbb{Z}^3}$ using the following local rules: symbols from the original alphabet $\mathcal{A}$ may either sit next to other symbols from $\mathcal{A}$ according to the SFT rules imposed on points in Z, or they may be placed adjacent to a shadow symbol containing an arrow pointing away from the symbol in $\mathcal{A}$, i.e. if site $\vec{\imath} \in \mathbb{Z}^3$ contains a symbol in $\mathcal{A}$ while site $\vec{\imath} \pm \vec{e}_i$ $(i \in \{1,2,3\})$ contains a shadow symbol, it has to be of type "$S_{\pm \vec{e}_i}$". In particular this forces connected patterns composed of symbols from $\mathcal{A}$ to have the shape of (possibly infinite) rectangular prisms, and forces such patterns to be locally admissible for Z. The 26 shadow symbols can a priori occur next to any other symbol in $\widetilde{\mathcal{A}}$. However using nearest-neighbor constraints, we forbid an increase by 2 and any decrease in the i-th coordinate as well as any change in the j-th coordinate $(i \neq j \in \{1,2,3\})$ of the index $\vec{c} \in \{-1, 0, +1\}^3 \backslash \{(0,0,0)\}$ when moving through consecutive shadow symbols along the $+\vec{e}_i$-direction, i.e. suppose sites $\vec{\imath} \in \mathbb{Z}^3$ and $\vec{\imath} + \vec{e}_i$ both contain shadow symbols "$S_{\vec{c}}$" and "$S_{\vec{c}'}$" respectively, then $0 \leqslant \vec{c}'_i - \vec{c}_i \leqslant 1$ and $\vec{c}'_j = \vec{c}_j$ for $j \neq i$. This implies in particular that in points of $\widetilde{Z}$, any finite "run" of adjacent shadow symbols "$S_{\pm \vec{e}_i}$" along the $\pm \vec{e}_i$-direction must begin with a symbol from $\mathcal{A}$ (sitting on the boundary of a rectangular prism filled with symbols from $\mathcal{A}$) and must continue until it hits a

wall symbol. Finally, a wall symbol may either appear next to a shadow symbol or next to other wall symbols; it may not appear next to any symbol from $\mathcal{A}$. We force the obvious nearest-neighbor constraints of having any wall segments intersecting (hitting) the boundary of any symbol (from the wall alphabet $\mathcal{A}_w$ as visualized in Figure 7) to "match up". More formally, whenever two symbols from $\mathcal{A}_w$ appear next to each other in either of the three cardinal directions they share a common square face, and the wall segments in either of the symbols touching this face from one side have to pass through the face and be continued inside the second symbol. In particular, a wall symbol can only occur adjacent to a shadow symbol if no wall segment hits the face the two symbols have in common. In addition to this, for each $i \in \{1, 2, 3\}$ we impose the following nearest-neighbor rule: If site $\vec{\imath} \in \mathbb{Z}^3$ contains a wall symbol while site $\vec{\imath} + \vec{e}_i$ (resp. $\vec{\imath} - \vec{e}_i$) contains a shadow symbol "$S_{\vec{c}}$", then $\vec{c}_i = -1$ (resp. $\vec{c}_i = +1$).

Let $s \in \mathbb{N}$ be a constant to be determined later which will represent the minimal width of the shadow cast by wall symbols in the following sense: if, in some point of $\widetilde{Z}$, a wall symbol appears at site $\vec{\imath} \in \mathbb{Z}^3$, then the whole cube $\vec{\imath} + C_s = \{\vec{\imath} + \vec{\jmath} \mid \vec{\jmath} \in \mathbb{Z}^3 \land \|\vec{\jmath}\|_\infty \leqslant s\} \subsetneq \mathbb{Z}^3$ of side length $2s + 1$ centered at $\vec{\imath}$ must contain only symbols from the wall alphabet $\mathcal{A}_w$ (i.e. at the sites inside this cube we forbid the occurrence of any symbol from $\mathcal{A}$).

Finally, we impose a local rule which forbids the existence of two parallel wall segments at a distance smaller than $2s + 2$ in any of the three cardinal directions, which has the effect of separating walls by a distance at least $2s + 2$ which always allows filling in some symbols from $\mathcal{A}$ between shadows cast by walls. For example, if we place the third symbol from the top row of Figure 7 at site $\vec{\imath} \in \mathbb{Z}^3$, then none of the wall symbols containing a wall segment parallel to the $\vec{e}_2\vec{e}_3$-plane can appear at any sites in $\{\vec{\imath} + k\vec{e}_1 \mid k \in \mathbb{Z} \land 1 \leqslant |k| \leqslant 2s + 1\} \subsetneq \mathbb{Z}^3$, and none of the wall symbols containing a wall segment parallel to the $\vec{e}_1\vec{e}_3$-plane in its right half can appear at any sites in $\{\vec{\imath} + l\vec{e}_2 \mid l \in \mathbb{Z} \land 1 \leqslant |l| \leqslant 2s + 1\} \subsetneq \mathbb{Z}^3$. Note that this symbol at $\vec{\imath}$ does not cause any constraint on segments of wall symbols parallel to the $\vec{e}_1\vec{e}_3$-plane in the left half of sites in $\{\vec{\imath} + l\vec{e}_2 \mid l \in \mathbb{Z} \land 1 \leqslant |l| \leqslant 2s + 1\} \subsetneq \mathbb{Z}^3$ nor any restrictions on segments of wall symbols parallel to the $\vec{e}_1\vec{e}_2$-plane.

As a consequence of all of these rules, points in $\widetilde{Z}$ are forced to have the global structure indicated in Figure 8: walls built from the wall symbols in $\mathcal{A}_w$ partition $\mathbb{Z}^3$ into cells that have the shape of (possibly infinite) rectangular prisms whose sides always have a length of at least $2s + 1$. Those rectangular prisms are filled by a collar of width at least s (from the prism's boundary) consisting entirely of shadow symbols, while the remaining central rectangular prism of side lengths at least 1 sitting inside this collar is filled with a locally admissible pattern from Z. The location of this central prism then completely forces the types of shadow symbols seen in the collar.

Since all those constraints are given by local rules of radius not bigger than the maximum between $2s + 2$ and the radius necessary to define Z, the $\mathbb{Z}^3$ subshift $\widetilde{Z} \subseteq \widetilde{\mathcal{A}}^{\mathbb{Z}^3}$ is again of finite type and clearly contains Z as a proper subsystem – obviously any point of Z (not containing any wall symbols) satisfies the constraint on local admissibility in all of $\mathbb{Z}^3$ and thus is globally admissible in $\widetilde{Z}$.

We now have to prove the two main properties of $\widetilde{Z}$ announced in the theorem.

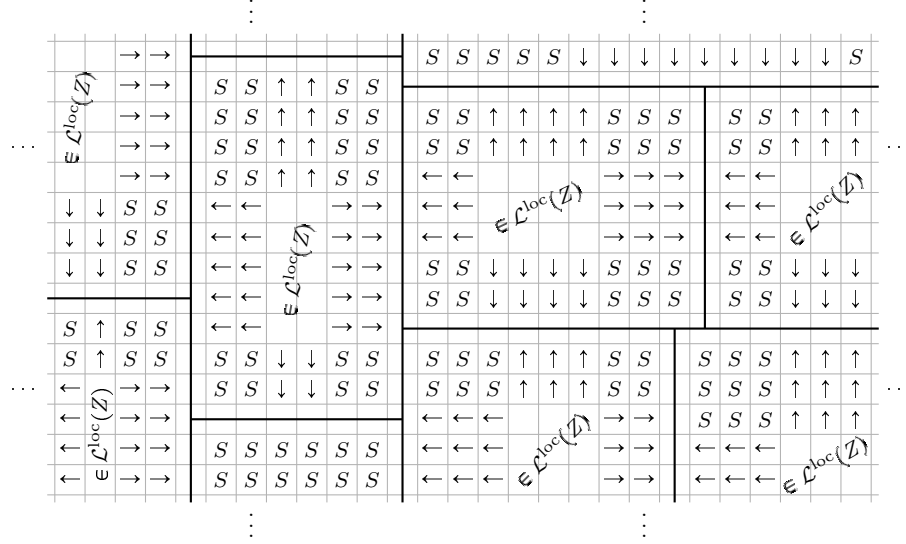**Claim 5.11.1.** $\mathrm{h_{top}}(\widetilde{Z}) = \mathrm{h_{top}}(Z)$.

FIGURE 8. A two dimensional cross section through a typical point in $\widetilde{Z}$ (in the picture we chose a minimal width of $s = 2$ for the shadow and suppressed the type $\vec{c}$ of shadow symbols "$S_{\vec{c}}$", except in the case $\vec{c} = \pm\vec{e}_i$.

Proof. Since $Z \subseteq \widetilde{Z}$, we immediately conclude that $\mathrm{h}_{\mathrm{top}}(Z) \leqslant \mathrm{h}_{\mathrm{top}}(\widetilde{Z})$ and thus it suffices to show the reversed inequality. The main step here is finding an upper bound on the cardinality $\left|\mathcal{L}_{n,n,n}(\widetilde{Z})\right|$ of cube-shaped globally admissible patterns for large sizes $n \in \mathbb{N}$. The core of the argument, which refines similar proofs used by the authors in [5, 19], is based on the fact that points of $\widetilde{Z}$ cannot contain a maximal connected component of wall symbols of finite size, i.e. any wall symbol within a globally admissible pattern $P \in \mathcal{L}_{n,n,n}(\widetilde{Z})$ has to be part of a connected component of walls which has to extend all the way to the boundary of P. This allows us to define an algorithm to produce (and estimate the number of) all possible ways to assign the wall symbols of a globally admissible pattern of shape $\mathbb{Z}^3 \cap [1, n]^3$, starting from the boundary. As this part is a bit tedious and similar to an argument in [5], we only give a brief summary, and refer to [5, C.4] for details.

Fix any integer $0 \leqslant q \leqslant n^3$ – which represents the number of wall symbols within a pattern $P \in \mathcal{L}_{n,n,n}(\widetilde{Z})$. We will prove that the number of globally admissible ways to place exactly this many wall symbols, while also determining the position and types of all shadow symbols, inside the cube $[1, n]^3$ is bounded from above by $28^{6n^2} \cdot 576^q$ by defining a deterministic algorithm which can generate any one of such patterns. The bound comes from counting the number of possible inputs of this algorithm where the input consists of two pieces: First there is a finite list $B \in \mathcal{B}^{\mathbb{Z}^3 \cap [1,n]^3 \setminus [2,n-1]^3}$ over the alphabet $\mathcal{B} := \{*, w\} \,\dot\cup\, \{S_{\vec{c}} \mid \vec{c} \in \{-1, 0, +1\}^3 \setminus \{(0,0,0)\}\}$ specifying the configuration of shadow symbols on the boundary (ordered e.g. lexicographically) of the cube $[1, n]^3$. Note that the cardinality of the boundary and thus the length of B is $\left|\mathbb{Z}^3 \cap [1,n]^3 \setminus [2, n-1]^3\right| = 6n^2 - 12n + 8$ and that elements in B tell us to fill each site of the boundary with one of the 26 shadow symbols "$S_{\vec{c}}$",

to use this site as the starting point of a connected component of wall symbols ("w")
or to simply skip filling this site ("$*$") as it may be used later or may be part of a
central prism reserved to be filled with a locally admissible pattern from Z. The sec-
ond piece of the input is a finite ordered list $\mathcal{I}$ of exactly q instructions – commands
to place a single wall symbol plus adjacent shadow symbols. Each instruction in
$\mathcal{I}$ is itself a 4-tuple (a_i, s_i, f_i, d_i) $(1 \leqslant i \leqslant q)$, where $a_i \in \mathcal{A}_w \backslash \{\text{shadow symbols}\}$
selects one of the 39 wall symbols to be placed next, $s_i \in \{S, \text{arrow}\} \, \dot\cup \, \{S, \text{arrow}\}^2$
coarsely specifies the type of the (up to two) shadow symbol(s) adjacent to the wall
symbol a_i, $f_i \in \{\text{revert}, \text{continue}\}$ is a simple flag used to steer the course of actions
taken by the algorithm, and $d_i \in \{1, 2, 3, 4\}$ is a number indicating – depending on
the present symbol a_i and the wall symbols already placed around it – the cardinal
direction in which the next wall symbol will be placed.

The algorithm now works like a Turing machine processing its input and putting
down symbols on an initially empty cube (tape) $\mathbb{Z}^3 \cap [1, n]^3$, starting by moving
its read-write head along all cells in the boundary of the cube $[1, n]^3$ placing there
the (partial) configuration specified by B. Sites for which B provides a symbol "$*$"
are skipped, while symbols "$S_{\vec{c}}$" are placed immediately. Whenever the algorithm
encounters a symbol "w" as part of B it consecutively executes instructions from
$\mathcal{I}$ as follows: Each 4-tuple (a_i, s_i, f_i, d_i) from the list $\mathcal{I}$ first causes the head to
write down the wall symbol a_i at its present location, and then to write down the
information about the type of adjacent shadow symbol(s) specified by s_i. There are
at most two of these, since at least four faces of any wall symbol contain segments
which force another wall symbol, and we may suppose that the six neighbors of
a site $\vec{\imath} \in \mathbb{Z}^3$ are considered in the order $\vec{\imath} + \vec{e}_1, \vec{\imath} - \vec{e}_1, \vec{\imath} + \vec{e}_2, \vec{\imath} - \vec{e}_2, \vec{\imath} + \vec{e}_3, \vec{\imath} - \vec{e}_3$
for consistency. After this the algorithm checks which of the 6 neighboring sites
of the current position are still inside the cube $\mathbb{Z}^3 \cap [1, n]^3$, share a square face
with the symbol a_i which is hit by a wall segment, and which have not yet been
filled in by another wall symbol. Note that for each wall symbol at most 4 of its
neighboring sites satisfy those conditions (at most 5 faces are hit by wall segments,
and of the 5 corresponding adjacent sites, one is either outside $[1, n]^3$ or has already
been assigned a wall symbol in the process of arriving at the present location).
Now, if the flag f_i signals "continue", the read-write head moves to the d_i-th of
those unassigned neighboring sites (again with respect to the previously described
ordering on the six neighbors) and continues with the next instruction. If the flag
f_i signals "revert", the read-write head instead backtracks along the way it came
until it finds the first site at which there are unassigned neighboring sites, moves
in the direction indicated by d_i from there, and again continues with the next
instruction from $\mathcal{I}$. If on its way back the read-write head does not find any sites
with unassigned neighbors, then the current connected component of wall symbols
inside the cube $[1, n]^3$ is completed, so d_i is discarded and the read-write head
instead moves to the next site in the boundary which is again filled according to
B. This process continues until the end of B (and $\mathcal{I}$) is reached, at which point
at most q wall symbols have been placed. (Exactly $q = |\mathcal{I}|$ wall symbols will have
been placed for correctly chosen B and $\mathcal{I}$.)

Note that for any element in $\mathcal{L}_{n,n,n}(\widetilde{Z})$, there exists an input for which this
algorithm generates all of its wall symbols. To see this, note that in each of the
finitely many connected components of wall symbols, one may choose a spanning
tree rooted at a symbol in the boundary of the cube which, in addition, has a

wall segment meeting this boundary and whose site is recorded by a symbol "w" in B. This tree then determines a possible list of instructions (4-tuples for $\mathcal{I}$) traversing the spanning tree depth-first, placing the corresponding symbols in its nodes while moving along its branches. On the other hand, while many of its outputs might not correspond to actual globally admissible patterns of $\widetilde{Z}$, for any possible input, the algorithm produces a unique pattern of wall and shadow symbols on $[1,n]^3$. Moreover given the shadow symbols in the boundary, the wall symbols and the coarse information about shadow symbols adjacent to walls provided by the instruction in $\mathcal{I}$ is enough to uniquely determine the position and types of all remaining shadow symbols within the cube $[1,n]^3$. Hence the number of different inputs with $|\mathcal{I}| = q$ gives a loose upper bound on the number of different patterns in $\mathcal{L}_{n,n,n}(\widetilde{Z})$ containing exactly q wall symbols (plus a particular configuration of shadow symbols). Since $|\mathcal{B}| = 28$, $\left| \mathbb{Z}^3 \cap [1,n]^3 \backslash [2, n-1]^3 \right| \leqslant 6n^2$ and (separating different cases) there are not more than $(3 \cdot 4 \cdot 3 + 36 \cdot 2 \cdot 4) + (3 \cdot 4 + 36 \cdot 2) \cdot 3 = 576$ distinct 4-tuples which we actually need to build $\mathcal{I}$, we recover the above stated upper bound $28^{6n^2} \cdot 576^q$.

Now to give an upper bound on $\left| \mathcal{L}_{n,n,n}(\widetilde{Z}) \right|$ we have to check how many distinct globally admissible patterns $P \in \mathcal{L}_{n,n,n}(\widetilde{Z})$ may be produced once we fix an arbitrary globally admissible way of placing exactly q wall plus all shadow symbols inside a cube of side length n. Note that the portion of $\mathbb{Z}^3 \cap [1,n]^3$ not occupied by wall symbols has to consist of disjoint rectangular prisms, which we will refer to as **cells**. Those each have a collar of width at least s filled by shadow symbols. The type (arrow or "$S_{\vec{c}}$") of the shadow symbols along the boundary of each such cell (including partial cells at the boundary) have already been assigned in the previous procedure by B and the second entry of each instruction in $\mathcal{I}$, and this information uniquely determines all shadow symbols inside the collar, leaving undetermined a (non-empty) central rectangular prism, which must be filled with a locally admissible pattern in Z. Since the fillings of these central rectangular prisms are independent of each other, the number of ways to extend the initially given pattern of wall and shadow symbols to the entire cube is thus the product over all cells of the numbers of locally admissible patterns in Z on the corresponding central prisms.

As stated in the beginning, the purpose of the upgradability condition is to bound those numbers from above. To wit, suppose there are $R \in \mathbb{N}_0$ such cells containing non-empty central rectangular prisms, and that these central prisms have dimensions $k_r \times l_r \times m_r$ $(k_r, l_r, m_r \in \mathbb{N})$ for each $1 \leqslant r \leqslant R$. Fixing the minimal width of the shadow to be $s := \left\lceil \frac{C}{2\,\mathrm{h}_{\mathrm{top}}(Z)} \right\rceil + 3 \left\lceil \frac{\log 576}{\mathrm{h}_{\mathrm{top}}(Z)} \right\rceil \in \mathbb{N}$ where $C \in \mathbb{R}_0^+$ is the upgradability constant of Z, we claim the following inequality:

(ii) $q + q \cdot \frac{\log 576}{\mathrm{h}_{\mathrm{top}}(Z)} + \sum_{r=1}^{R} \left(k_r + \frac{C}{\mathrm{h}_{\mathrm{top}}(Z)} \right) \left(l_r + \frac{C}{\mathrm{h}_{\mathrm{top}}(Z)} \right) \left(m_r + \frac{C}{\mathrm{h}_{\mathrm{top}}(Z)} \right) \leqslant (n + 2s)^3 \; .$

To see this, consider the cube of side length $n + 2s$ given by enlarging the original cube by s in each direction. Due to the forced collars of shadow symbols, this enlarged cube clearly contains the disjoint union of the central rectangular prisms in each cell extended by $\frac{C}{2\,\mathrm{h}_{\mathrm{top}}(Z)}$ in each direction. The union of all those enlarged central prisms has volume equal to the third term on the left-hand side of (ii). Moreover, even once this union is removed from the enlarged cube of volume $(n +$

$2s)^3$, each cell still contains a remaining collar of shadow symbols of width at least $3\lceil\frac{\log 576}{\mathrm{h}_{\mathrm{top}}(Z)}\rceil$. The total volume of those reduced collars exceeds the second summand on the left-hand side of (ii), as for each one of the q wall symbols there is at least one direction perpendicular to its principal wall segment where only shadow symbols occur throughout the whole reduced collar. Since each one of the symbols in the reduced collar can be part of this for up to three principal segments (there is some overlap near corners of a cell) each wall symbol in the original cube forces a volume of at least $\frac{1}{3}\cdot 3\lceil\frac{\log 576}{\mathrm{h}_{\mathrm{top}}(Z)}\rceil$ in the union of the reduced collars. Finally there are the q wall symbols themselves giving the additional q on the left and completing the proof of (ii).

We now apply (ii) to get

(iii)
$$
\begin{aligned}
\prod_{r=1}^{R}\big|\mathcal{L}_{k_r,l_r,m_r}^{\mathrm{loc}}(Z)\big| &\leqslant \prod_{r=1}^{R} e^{\mathrm{h}_{\mathrm{top}}(Z)k_r l_r m_r + C(k_r l_r + k_r m_r + l_r m_r)}\\
&\leqslant \prod_{r=1}^{R} e^{\mathrm{h}_{\mathrm{top}}(Z)\left(k_r+\frac{C}{\mathrm{h}_{\mathrm{top}}(Z)}\right)\left(l_r+\frac{C}{\mathrm{h}_{\mathrm{top}}(Z)}\right)\left(m_r+\frac{C}{\mathrm{h}_{\mathrm{top}}(Z)}\right)}\\
&= e^{\mathrm{h}_{\mathrm{top}}(Z)\sum_{r=1}^{R}\left(k_r+\frac{C}{\mathrm{h}_{\mathrm{top}}(Z)}\right)\left(l_r+\frac{C}{\mathrm{h}_{\mathrm{top}}(Z)}\right)\left(m_r+\frac{C}{\mathrm{h}_{\mathrm{top}}(Z)}\right)}\\
&\leqslant e^{\mathrm{h}_{\mathrm{top}}(Z)\cdot(n+2s)^3 - q\log 576 - q\,\mathrm{h}_{\mathrm{top}}(Z)} .
\end{aligned}
$$

In other words, for any fixed choice of wall and shadow symbols within a pattern of $\mathcal{L}_{n,n,n}(\widetilde{Z})$, the number of ways to legally fill the remaining central prisms with letters of $\mathcal{A}$ is less than or equal to $e^{\mathrm{h}_{\mathrm{top}}(Z)\cdot(n+2s)^3 - q\log 576 - q\,\mathrm{h}_{\mathrm{top}}(Z)}$. We can now multiply the upper bound on the number of distinct patterns with q wall (plus shadow) symbols obtained above by (iii) and sum over all possible values of q to get the desired bound for the cardinality of globally admissible patterns:

(iv)
$$
\begin{aligned}
\forall\, n\in\mathbb{N}:\qquad \big|\mathcal{L}_{n,n,n}(\widetilde{Z})\big| &\leqslant \sum_{q=0}^{n^3} 28^{6n^2}\cdot 576^q \cdot \prod_{r=1}^{R}\big|\mathcal{L}_{k_r,l_r,m_r}^{\mathrm{loc}}(Z)\big|\\
&\leqslant \sum_{q=0}^{n^3} 28^{6n^2}\cdot \cancel{576^q}\cdot e^{\mathrm{h}_{\mathrm{top}}(Z)\cdot(n+2s)^3 - \cancel{q\log 576} - q\,\mathrm{h}_{\mathrm{top}}(Z)}\\
&\leqslant (n^3+1)\cdot 28^{6n^2}\cdot e^{\mathrm{h}_{\mathrm{top}}(Z)\cdot(n+2s)^3}
\end{aligned}
$$

which finally yields – as s is a fixed natural number – the sought-after inequality $\mathrm{h}_{\mathrm{top}}(\widetilde{Z}) = \lim_{n\to\infty}\frac{\log|\mathcal{L}_{n,n,n}(\widetilde{Z})|}{n^3} \leqslant \mathrm{h}_{\mathrm{top}}(Z).$ $\qquad\square$

Claim 5.11.2. $\widetilde{Z}$ is block gluing with gap size $g = 6s + 6$.

Proof. By symmetry of the wall alphabet $\mathcal{A}_w$ and the rules defining $\widetilde{Z}$ it suffices to prove the claim for the cardinal direction $\vec{e}_1$, i.e. we show that for every pair $x, y \in \widetilde{Z}$ of points, it is always possible to find another valid point $z \in \widetilde{Z}$ such that $z|_{(-\infty,0]\times\mathbb{Z}^2} = x|_{(-\infty,0]\times\mathbb{Z}^2}$ and $z|_{[g,\infty)\times\mathbb{Z}^2} = y|_{[g,\infty)\times\mathbb{Z}^2}$ for some fixed universal $g \in \mathbb{N}_0$. The same argument will then work for the other cardinal directions as well.

Consider a point $x \in \widetilde{Z}$ and note that by definition of $\widetilde{Z}$, its $\mathbb{Z}^2$-cross section $x' := x|_{\{0\}\times\mathbb{Z}^2}$ yields a rectangulation of $\mathbb{Z}^2$ by disjoint (possibly infinite) rectangular cells – cross sections of the cells in x – induced by the wall symbols in x'. Now to define $z \in \widetilde{Z}$, we first wish to extend all rectangular cells appearing in the cross

section x' along the $\vec{e}_1$-direction for exactly $2s + 1$ steps. We begin with the wall symbols, which can be extended simply by continuing all wall segments parallel to the $\vec{e}_1\vec{e}_2$- or $\vec{e}_1\vec{e}_3$-plane, thus keeping the cell structure of the cross section x' unchanged over this distance. To fill each of those extended rectangular prisms defined by the wall symbols, we distinguish 4 different cases depending on what symbols we see inside each rectangular cell of the cross section x':

Case 1: If an entire cell in the cross section x' is filled with shadow symbols "$S_{\vec{c}}$" where $\vec{c}_1 = +1$, we are already in the right part of the collar and may just let each shadow symbol continue for another $2s + 1$ steps to the right.

Case 2: If we see symbols of the original alphabet $\mathcal{A}$ the cross section x' slices through a locally admissible pattern of Z sitting in the central prism of a cell. As such a pattern may not be extensible further along the $\vec{e}_1$-direction, we place immediately to the right of each symbol in $\mathcal{A}$ a shadow symbol with an arrow oriented in the $+\vec{e}_1$-direction and immediately to the right of all other shadow symbols in the cell's cross section a corresponding shadow symbol "$S_{\vec{c}}$" with $\vec{c}_1 = +1$. Now we are back to Case 1, and extend this pattern in a constant way for another $2s$ steps.

Case 3: If a cell of the cross section x' contains only shadow symbols and there is a site occupied by an arrow pointing in the $-\vec{e}_1$-direction, those arrows determine the rectangular "left" face of the central prism to be filled with a locally admissible pattern of Z. Hence we first extend the collar of shadow symbols for exactly s steps, fill in the $\{s + 1\} \times \mathbb{Z}^2$-cross section of the extended prism with an arbitrary rectangular locally admissible pattern of Z having the correct size (and position) and being surrounded by a corresponding collar of shadow symbols and finish as before with a stack of s layers of shadow symbols forming the right part of the collar.

Case 4: Finally if a rectangular cell in x' consists entirely of wall symbols with their principal segment parallel to the $\vec{e}_2\vec{e}_3$-plane (and all further segments to its left), the extended prism can be completely filled with an arbitrary pattern seen in a cell of the correct size in a point of $\tilde{Z}$.

Now, since each rectangular cell in the cross section x' fell into one of these cases, the s right-most cross sections (filling $\{s + 2, \ldots, 2s + 1\} \times \mathbb{Z}^2$) of this extension constructed from x have rectangular cells filled with shadow symbols of type "$S_{\vec{c}}$" with $\vec{c}_1 = +1$. We can therefore "top off" this entire extension with a cross section filling the plane $\{2s + 2\} \times \mathbb{Z}^2$ consisting entirely of wall symbols with their principal segments parallel to the $\vec{e}_2\vec{e}_3$-plane and any additional segments (forced by the extension) pointing to the left. The same strategy (with all $\vec{e}_1$ directions reversed) is used to close all rectangular cells in the cross section $y' := y|_{\{g\} \times \mathbb{Z}^2}$ within $2s + 2$ steps, now with a left-most cross section (filling $\{g - 2s - 2\} \times \mathbb{Z}^2$) consisting entirely of wall symbols with their principal segments parallel to the $\vec{e}_2\vec{e}_3$-plane and any additional segments (forced by the extension) pointing to the right.

If we choose the gap size $g := 6s + 6 \in \mathbb{N}_0$, then taking the union of these two extensions yields a point with a single undefined plane of width $2s+1$ sitting between the two cross sections $\{2s + 2\} \times \mathbb{Z}^2$ and $\{4s + 4\} \times \mathbb{Z}^2$ both consisting entirely of "outwards-pointing" wall symbols. Filling each $\vec{e}_2\vec{e}_3$-cross section $\{k\} \times \mathbb{Z}^2$ in this region completely with shadow symbols containing a symbol "$S_{(\vec{c}_1, +1, +1)}$" with $\vec{c}_1 := \delta_{k > 3s+3} - \delta_{k < 3s+3} \in \{-1, 0, +1\}$ ($2s + 3 \leqslant k \leqslant 4s + 3$) then finishes the construction of z, which clearly is an admissible point in $\tilde{Z}$. $\qquad\square$

$\square$

Remark 5.13. Although the construction presented in this section only works in three (or more) dimensions, it seems likely that a similar result holds in two dimensions. In fact, recent work of Nathalie Aubrun and Mathieu Sablik [1] proposes a constructive method to realize any effective $\mathbb{Z}$ shift as the one-dimensional subdynamics seen on one layer of a $\mathbb{Z}^2$ SFT consisting of multiple superimposed layers. Hence one could hope to apply this technique to realize $\mathbb{Z}$ Sturmian subdynamics of a certain frequency $\alpha' \in [0,1]\backslash\mathbb{Q}$ in a still upgradable $\mathbb{Z}^2$ SFT, though computability condition **(C)** might need to be modified. As the construction in Section 5 is already quite complicated and elaborate in three dimensions, we have here refrained from attempting an even more intricate construction for $d = 2$.

6. No equal entropy full shift factor – Proof of Theorem 1.9

In our proof of Theorem 1.9, we will apply the construction of Theorem 1.7 to $\alpha = \log L$ with $1 < L \in \mathbb{N}$. Obviously $\log L$ satisfies computability condition **(C)** since it can be decomposed as $1 \cdot \log L$, however for our argument we need to use the decomposition $\frac{\log L}{\log M} \cdot \log M$ with $L < M \in \mathbb{N}$ coprime natural numbers. It must then be verified that the partial quotients of the continued fraction expansion of $\alpha' = \frac{\log L}{\log M}$ can be computed at the rate proscribed in Definition 1.5.

Proposition 6.1. *For any pair of natural numbers $L < M \in \mathbb{N}$ with $\gcd(L,M) = 1$, the representation $\alpha = \frac{\log L}{\log M} \cdot \log M \in \mathbb{R}$ satisfies computability condition* **(C)**.

Proof. Following essentially a version of the Euclidean algorithm we describe an accelerated TM procedure to generate the partial quotients of the continued fraction expansion for the irrational $\alpha' = \frac{\log L}{\log M}$ and then analyze its runtime. Our algorithm involves two recursively defined auxiliary sequences $(e_n), (f_n)_{n\in\mathbb{N}_0}$, along with the generated continued fraction partial quotients $(a_n)_{n\in\mathbb{N}}$. Let the initial values $e_{-1} := 1$, $e_0 := 0$, $f_{-1} := 0$ and $f_0 := 1$ be hard-coded into an accelerated TM. Further values of these sequences are then computed recursively, starting from $n = 1$: if n is odd, a_n is defined to be the largest integer such that $L^{a_n f_{n-1}+f_{n-2}} < M^{a_n e_{n-1}+e_{n-2}}$, and if n is even, a_n is defined to be the largest integer such that $L^{a_n f_{n-1}+f_{n-2}} > M^{a_n e_{n-1}+e_{n-2}}$. In either case, we also define $e_n := a_n e_{n-1} + e_{n-2}$, $f_n := a_n f_{n-1} + f_{n-2}$, allowing computation of a_{n+1} and the continuation of the recursion.

We claim that $\left(\frac{e_n}{f_n}\right)_{n\in\mathbb{N}_0}$ is the sequence of continued fraction approximants to $\alpha' = \frac{\log L}{\log M}$ and that $(a_n)_{n\in\mathbb{N}}$ are the partial quotients. Obviously the recursions $e_n = a_n e_{n-1} + e_{n-2}$ and $f_n = a_n f_{n-1} + f_{n-2}$ (and their starting values) are correct. Also, for odd indices n, another way of stating the definition of a_n is that $L^{a_n f_{n-1}+f_{n-2}} < M^{a_n e_{n-1}+e_{n-2}}$ and $L^{(a_n+1)f_{n-1}+f_{n-2}} > M^{(a_n+1)e_{n-1}+e_{n-2}}$, or equivalently, $\frac{(a_n+1)e_{n-1}+e_{n-2}}{(a_n+1)f_{n-1}+f_{n-2}} < \frac{\log L}{\log M} < \frac{a_n e_{n-1}+e_{n-2}}{a_n f_{n-1}+f_{n-2}}$. Recall that by Corollary 2.4, this uniquely determines a_n. The reasoning for n even is similar.

All that remains is to explicitly describe the implementation of this algorithm in an accelerated Turing machine and to bound its runtime. Our TM will have 9 tapes, labeled by integers 1 through 9. We describe the computation of a_n for odd indices n, since the case for even indices n is trivially similar. In the first step of the computation of a_n, tape 1 and tape 2 both contain (a binary representation of) e_{n-1}, tape 3 contains $e_{n-1} + e_{n-2}$, tape 4 contains $M^{e_{n-1}+e_{n-2}}$, tape 5 and tape 6 both contain f_{n-1}, tape 7 contains $f_{n-1} + f_{n-2}$, tape 8 contains $L^{f_{n-1}+f_{n-2}}$, and

tape 9 contains the number 1, all in binary. (Clearly for $n = 1$, this information will have to be hard-coded into the TM as a starting condition for the 9 tapes.) The machine now runs a loop: in each of the next e_{n-1} steps, simultaneously, tapes 2 and 6 are decremented by 1, tapes 3 and 7 are incremented by 1, tape 4 is multiplied by M, and tape 8 is multiplied by L. As $1, L, M$ are constants those additions and multiplications each take only 1 step. Now, tape 2 will contain 0, which signals a new phase. For the next $f_{n-1} - e_{n-1}$ steps, simultaneously, tape 6 is decremented by 1, tape 7 is incremented by 1, and tape 8 is multiplied by L. After this, tapes 2 and 6 will both contain 0, which signals the end of the loop. At this point, the contents of tapes 4 and 8 are compared in an instant operation. Recall that $\gcd(L, M) = 1$, so the powers of M and L on tapes 4 respectively 8 can not be equal. If the number on tape 8 is larger, then the loop stops (we will describe the procedure for this momentarily). If the number on tape 4 is larger, then the loop repeats as follows: before the next step, tapes 2 and 6 are reset to e_{n-1} and f_{n-1} (this is done with an instant copy respectively from tapes 1 and 5, whose only role is to store the values of e_{n-1} and f_{n-1} for this resetting), and the value of tape 9 – i.e. the eventual value of a_n – is incremented by 1. Then the above loop can repeat just as before, with the phases corresponding to the first e_{n-1} steps (until tape 2 reaches 0) and the next $f_{n-1} - e_{n-1}$ steps (until tape 6 reaches 0).

As stated above, the loop stops at the first comparison in which the content of tape 8 is larger than that of tape 4, without increasing the value on tape 9 again. At this point, a signal is sent that the current content of tape 9 is the correct value of a_n. It should be clear that a_n is in fact the correct partial quotient of the continued fraction expansion, since, as was outlined above, it is the largest integer $a \in \mathbb{N}$ for which $L^{af_{n-1}+f_{n-2}} < M^{ae_{n-1}+e_{n-2}}$. We summarize the content of the rest of the tapes at this step: tape 1 contains e_{n-1}, tape 2 contains 0, tape 3 contains $e_{n-2} + (a_n + 1)e_{n-1} = e_n + e_{n-1}$, tape 4 contains $M^{e_{n-2}+(a_n+1)e_{n-1}} = M^{e_n+e_{n-1}}$, tape 5 contains f_{n-1}, tape 6 contains 0, tape 7 contains $f_{n-2} + (a_n + 1)f_{n-1} = f_n + f_{n-1}$, tape 8 contains $L^{f_{n-2}+(a_n+1)f_{n-1}} = L^{f_n+f_{n-1}}$, and tape 9 contains a_n. To initialize the machine for the computation of the next partial quotient a_{n+1}, tapes 1 and 2 are overwritten in an instant operation by tape 3 minus tape 1, i.e. e_n, tapes 5 and 6 are overwritten by tape 7 minus tape 5, i.e. f_n, and tape 9 is reset to 1. At this point, the content of the tapes is as initially described above with n replaced by $n + 1$, and so the accelerated TM can continue in a recursive fashion to compute the next partial quotient. (Note that now n is even and the condition for the comparison between tapes 4 and 8 has to be reversed.)

Since each iteration of the described loop takes f_{n-1} steps, and the loop is run a_n times, the number of TM steps that were used to compute a_n is exactly $a_n f_{n-1}$. The computation time for the first $N \in \mathbb{N}$ partial quotients $a_1, \ldots, a_N$ is thus:

$$\sum_{n=1}^{N} a_n f_{n-1} \leqslant a_N f_{N-1} + \sum_{n=1}^{N-1} f_n \leqslant (a_N + 1)f_{N-1} + \sum_{n=1}^{N-2} f_n$$
$$\leqslant (a_N + 3)f_{N-1} \leqslant 4a_N f_{N-1}$$

where we used the recursion for f_n and the fact that $\sum_{n=1}^{N-2} f_n \leqslant 2f_{N-1}$. Since the sequence $(t_n)_{n \in \mathbb{N}_0}$ used in Definition 1.5 coincides with the sequence $(f_n)_{n \in \mathbb{N}_0}$, by implementing a linear speed-up by a factor of 4 (via extra tapes, for example), our accelerated TM has the desired computation time satisfying condition **(C)**. $\qquad\square$

We can now use our techniques from Section 5 to construct for $d \geqslant 3$ an upgradable zero-entropy $\mathbb{Z}^d$ SFT X' on alphabet $\mathcal{A}'$ so that letters from a certain subset $\mathcal{A}'_1 \subsetneq \mathcal{A}'$ appear in all points of X' with frequency $\alpha' = \frac{\log L}{\log M}$. ($\mathcal{A}'_1$ is just the set of letters from $\mathcal{A}'$ which contain a 1 on the base layer.) Then, by splitting symbols in $\mathcal{A}'_1$ into M independent copies, we can create an upgradable $\mathbb{Z}^d$ SFT X'' with entropy $\mathrm{h}_{\mathrm{top}}(X'') = \alpha' \log M = \log L$, and by applying Theorem 5.11, we can upgrade X'' to a block gluing $\mathbb{Z}^d$ SFT X again with entropy $\mathrm{h}_{\mathrm{top}}(X) = \log L$.

In order to apply the techniques from [6], we need the following proposition:

Proposition 6.2. *Let Z be an upgradable $\mathbb{Z}^3$ shift of finite type with $\mathrm{h}_{\mathrm{top}}(Z) > 0$ and let $\widetilde{Z} \supsetneq Z$ be its equal entropy block gluing version as constructed in Theorem 5.11. Then any measure of maximal entropy on $\widetilde{Z}$ has support contained in Z.*

Proof. Consider any ergodic shift-invariant probability measure of maximal entropy μ on $\widetilde{Z}$, and assume for a contradiction that the support of μ is not contained in Z. Define the probabilities $p_w := \mu\big(\{z \in \widetilde{Z} \mid z_{\vec{0}} \text{ is a wall symbol}\}\big)$ and $p_s := \mu\big(\{z \in \widetilde{Z} \mid z_{\vec{0}} \text{ is a shadow symbol}\}\big)$. Clearly the sum $p_w + p_s$ has to be strictly positive. We break the argument into two cases depending on whether or not $p_w > 0$.

If $p_w = 0$, then $\mu\big(\bigcup_{\vec{\imath} \in \mathbb{Z}^3} \{z \in \widetilde{Z} \mid z_{\vec{\imath}} \text{ is a wall symbol}\}\big) = 0$ by shift-invariance of μ, and so μ-a.e. point of $\widetilde{Z}$ contains no wall symbols. Clearly the only such points either purely consist of shadow symbols – let us call them **pure shadow points** – or consist of a single non-empty (finite or infinite) rectangular prism filled with a locally admissible pattern from $\mathcal{L}^{\mathrm{loc}}(Z)$, surrounded by shadow symbols.

Consider the latter points first. For each $1 \leqslant i \leqslant 3$ and any non-empty (finite or one-sided infinite) interval $I \subsetneq \mathbb{Z}$, denote by $\widetilde{Z}_{\vec{e}_i, I} \subseteq \widetilde{Z}$ the set of points for which the $\vec{e}_i$-dimension of this rectangular prism is the interval I. The sets $\widetilde{Z}_{\vec{e}_i, I}$ cannot have positive μ-measure, since all their shifts $\sigma^{n\vec{e}_i}\big(\widetilde{Z}_{\vec{e}_i, I}\big) = \widetilde{Z}_{\vec{e}_i, I+n}$ ($n \in \mathbb{Z}$) are disjoint and have to have the same measure. Therefore, μ-a.s., the rectangular prism in question can not be a non-empty proper subset of $\mathbb{Z}^3$. Since μ is assumed to be ergodic with $p_s > 0$, this implies that μ-a.e. point in $\widetilde{Z}$ has to be a pure shadow point. However, there are only countably many pure shadow points most of which do not carry measure. A full proof would be quite technical and we leave it to the reader to confirm the details, but roughly speaking there are three cases: If a pure shadow point contains a shadow symbol "$S_{\pm \vec{e}_i}$" (w.l.o.g. assume $i = 1$), then such shadow symbols have to occur at sites in $\mathbb{Z} \times R$ for a (finite or infinite) rectangle $R \subseteq \mathbb{Z}^2$, and R uniquely determines the whole point. If a pure shadow point contains no shadow symbols "$S_{\pm \vec{e}_i}$" but contains a shadow symbol "$S_{\vec{c}}$" with $\|\vec{c}\|_1 = 2$ (w.l.o.g. assume $\vec{c} = (1, 1, 0)$), then such shadow symbols occur at sites in $\mathbb{Z}^2 \times I$ for a (finite or infinite) interval $I \subseteq \mathbb{Z}$, and I uniquely determines the whole point. Finally, if a pure shadow point contains no shadow symbols "$S_{\vec{c}}$" with $\|\vec{c}\|_1 < 3$, then it is constant; all sites contain identical shadow symbols. This clearly gives only countably many choices (selecting R, I and $\vec{c}$). However, again by shift-invariance of μ, all non-constant pure shadow points must have μ-measure zero, therefore μ is supported entirely on the 26 constant pure shadow points, and so has entropy zero, a contradiction.

If $p_w > 0$, then by the ergodic theorem, for μ-a.e. $z \in \widetilde{Z}$,

$$\lim_{n \to \infty} \frac{\left|\{\vec{\imath} \in [0, n-1]^3 \mid z_{\vec{\imath}} \text{ is a wall symbol}\}\right|}{n^3} = p_w \ .$$

Then, for any $0 < \epsilon < p_w$ and $n \in \mathbb{N}$, if we define $A_{n,\epsilon}$ to be the set of patterns in $\mathcal{L}_{n,n,n}(\widetilde{Z})$ containing at least $n^3(p_w - \epsilon)$ wall symbols, then for μ-a.e. $z \in \widetilde{Z}$, there exists $N \in \mathbb{N}$ so that $z|_{[0,n-1]^3} \in A_{n,\epsilon}$ for $n > N$. Therefore, there exists $N \in \mathbb{N}$ so that $\mu(A_{n,\epsilon}) > 1 - \epsilon$ for all $n > N$.

However, the same argument which gave (iv) in the proof of Theorem 5.11 shows that $|A_{n,\epsilon}|$ is bounded from above by

(v)
$$|A_{n,\epsilon}| \leqslant \sum_{q=\lceil n^3(p_w - \epsilon)\rceil}^{n^3} 28^{6n^2} \cdot \cancel{576^q} \cdot e^{\mathrm{h_{top}}(Z) \cdot (n+2s)^3 - \cancel{q\log 576} - q\,\mathrm{h_{top}}(Z)}$$

$$\leqslant (n^3 + 1) \cdot 28^{6n^2} \cdot e^{\mathrm{h_{top}}(Z)\big((n+2s)^3 - n^3(p_w - \epsilon)\big)} \ .$$

Let $A_{n,\epsilon}^{\complement} := \mathcal{L}_{n,n,n}(\widetilde{Z}) \backslash A_{n,\epsilon}$. By the definition of measure-theoretic entropy:

(vi)
$$h_\mu(\widetilde{Z}) = \lim_{n \to \infty} \frac{1}{n^3} \sum_{P \in \mathcal{L}_{n,n,n}(\widetilde{Z})} -\mu([P])\log\mu([P])$$

$$= \lim_{n \to \infty} \frac{1}{n^3}\Big(\sum_{P \in A_{n,\epsilon}} -\mu([P])\log\mu([P]) + \sum_{P \in A_{n,\epsilon}^{\complement}} -\mu([P])\log\mu([P]) \Big) \ .$$

Now the first sum is trivially bounded from above by $\log|A_{n,\epsilon}|$ while for $n > N$ the second sum is bounded by

$$\sum_{P \in A_{n,\epsilon}^{\complement}} -\mu([P])\log\mu([P]) \leqslant \sum_{P \in A_{n,\epsilon}^{\complement}} \big(-\tfrac{\epsilon}{|A_{n,\epsilon}^{\complement}|}\log\tfrac{\epsilon}{|A_{n,\epsilon}^{\complement}|}\big) \leqslant \epsilon\log\big(\tfrac{|\mathcal{L}_{n,n,n}(\widetilde{Z})|}{\epsilon}\big)$$

since $\mu(A_{n,\epsilon}^{\complement}) < \epsilon$. Putting those two bounds into (vi) and using (v), we get:

(vii)
$$h_\mu(\widetilde{Z}) \leqslant \lim_{n \to \infty} \frac{1}{n^3}\Big(\log(n^3 + 1) + 6n^2\log 28 + (n + 2s)^3\,\mathrm{h_{top}}(Z)$$

$$- n^3(p_w - \epsilon)\,\mathrm{h_{top}}(Z) + \epsilon\log|\mathcal{L}_{n,n,n}(\widetilde{Z})| - \epsilon\log\epsilon \Big)$$

$$= \mathrm{h_{top}}(Z) - (p_w - \epsilon)\,\mathrm{h_{top}}(Z) + \epsilon\,\mathrm{h_{top}}(\widetilde{Z}) \ .$$

However, clearly for small enough $\epsilon \ll p_w$, this is strictly less than $\mathrm{h_{top}}(Z)$, contradicting the fact that μ was a measure of maximal entropy for $\widetilde{Z}$. $\qquad\square$

Proof of Theorem 1.9. In [6], the following was shown: Suppose a $\mathbb{Z}^d$ SFT W with entropy $\mathrm{h_{top}}(W) = \log L$ is created from a zero-entropy $\mathbb{Z}^d$ SFT Z by introducing M independent copies of each symbol in a subset $\mathcal{A} \subsetneq \widetilde{\mathcal{A}}$ whose elements appear in points of Z with frequencies bounded from above by (but which can be arbitrarily close to) $\alpha' = \frac{\log L}{\log M}$. Then, if L, M are coprime, there cannot exist a measure μ of maximal entropy on W (i.e. $h_\mu(W) = \log L$) and a topological factor map ϕ under which μ maps to the uniform Bernoulli measure m_B on the full L-shift. This was used in [6] to show that such a subshift W does not factor topologically onto the full L-shift; if it did (say under ϕ), then the measure m_B would have to have a preimage under ϕ, which would be forced to have entropy $\log L$.

Now consider our situation. Suppose for a contradiction that X factors onto the full L-shift via a topological factor map ϕ. Then the uniform Bernoulli measure m_B on the full L-shift has a preimage μ under ϕ, which has entropy $h_\mu(X) = \log L$ and so is a measure of maximal entropy on X. By Proposition 6.2, μ has support contained in X''. However, X'' was constructed from X' by creating M independent

copies of each symbol from a set of symbols $\mathcal{A}_1' \subsetneq \mathcal{A}'$ whose elements appear in points of X' with frequencies bounded from above by (but arbitrarily close to) $\alpha' = \frac{\log L}{\log M}$, so by the result from [6] given above, μ cannot map to the uniform Bernoulli measure m_B on the full L-shift under ϕ, and we have a contradiction. $\square$

References

[1] N. Aubrun and M. Sablik, *Simulation of effective shift spaces by two-dimensional SFT and a generalization*, preprint.

[2] R.J. Baxter, *Hard hexagons: Exact solution*, J. Physics A **13** (1980), 1023–1030.

[3] R.J. Baxter, *Planar lattice gases with nearest-neighbor exclusion*, Ann. Comb. **3** (1999), 191–203.

[4] R. Berger, *The undecidability of the domino problem*, Mem. Amer. Math. Soc. **66** (1966).

[5] M. Boyle, R. Pavlov and M. Schraudner, *Multidimensional sofic shifts without separation, and their factors*, Trans. Amer. Math. Soc. **362** (2010), 4617–4653.

[6] M. Boyle and M. Schraudner, $\mathbb{Z}^d$ *shifts of finite type without equal entropy full shift factors*, J. Difference Equ. Appl. **15** (2009), no. 1, 47–52.

[7] S. Friedland, P. Lundow, and K. Markström, *The 1-vertex transfer matrix and accurate estimation of channel capacity*, IEEE Trans. Inform. Theory **56** (2010), no. 8, 3692–3699.

[8] M. Hochman, *On the dynamics and recursive properties of multidimensional symbolic systems*, Inventiones Mathematica, **176** (2009), no. 1, 131–167.

[9] M. Hochman and T. Meyerovitch, *A characterization of the entropies of multidimensional shifts of finite type*, Ann. Math. **171** (2010), no. 3, 2011–2038.

[10] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Series in Computer Science, Addison-Wesley, Reading, 1979.

[11] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, Oxford, 1938.

[12] P.W. Kasteleyn, *The statistics of dimers on a lattice I: the number of dimer arrangements on a quadratic lattice*, Physica **27** (1961), 1209–1225.

[13] E.H. Lieb, *Exact solution of the problem of the entropy of two-dimensional ice*, Phys. Rev. Lett. **18** (1967), 692–694.

[14] D. Lind, *The entropies of topological Markov shifts and a related class of algebraic integers*, Ergodic Theory Dynam. Systems **4** (1984), no. 2, 283–300.

[15] D. Lind and B. Marcus, *Introduction to symbolic dynamics and coding*, Cambridge University Press, Cambridge, 1995.

[16] M. Lothaire, *Algebraic combinatorics on words*, Encyclopedia of Mathematics and its Applications, vol. 90, Cambridge University Press, Cambridge, 2002.

[17] S. Mozes, *Tilings, substitution systems and dynamical systems generated by them*, J. Analyse Math. **53** (1989), 139–186.

[18] R. Pavlov, *Perturbations of multidimensional shifts of finite type*, Ergodic Theory Dynam. Systems **31** (2011), no. 2, 483–526.

[19] R. Pavlov, *Approximating the hard square entropy constant with probabilistic methods*, Ann. Probab. **40** (2012), no. 6, 2362–2399.

[20] C. Radin, *Miles of Tiles*, Student Mathematical Library, vol. 1, AMS, Providence, 1999.

[21] R. Robinson, *Undecidability and nonperiodicity for tilings of the plane*, Inv. Math. **11** (1971), 177–190.

[22] M. Schraudner, *Projectional entropy and the electrical wire shift*, Discrete Contin. Dyn. Syst. Series A **16** (2010), no. 1, 333–346.

Ronnie Pavlov, Department of Mathematics, University of Denver, 2360 S. Gaylord St., Denver, CO 80208
    E-mail address: rpavlov@du.edu
    URL: www.math.du.edu/~rpavlov/

Michael Schraudner, Centro de Modelamiento Matemático, Universidad de Chile, Av. Blanco Encalada 2120, Santiago de Chile
    E-mail address: mschraudner@dim.uchile.cl
    URL: www.cmm.uchile.cl/~mschraudner/