Finitary coding for the sub-critical Ising model with finite expected coding volume

Yinon Spinka*

Abstract

It has been shown by van den Berg and Steif [5] that the sub-critical Ising model on $\mathbb{Z}^d$ is a finitary factor of a finite-valued i.i.d. process. We strengthen this by showing that the factor map can be made to have finite expected coding volume (in fact, stretched-exponential tails), answering a question of van den Berg and Steif. The result holds at any temperature above the critical temperature. An analogous result holds for Markov random fields satisfying a high-noise assumption and for proper colorings with a large number of colors.

1 Introduction and main results

Let $(S, \mathcal{S})$ and $(T, \mathcal{T})$ be two measurable spaces, and let $X = (X_v)_{v \in \mathbb{Z}^d}$ and $Y = (Y_v)_{v \in \mathbb{Z}^d}$ be $(S, \mathcal{S})$-valued and $(T, \mathcal{T})$-valued stationary random fields (i.e., $\mathbb{Z}^d$-processes) for some $d \geq 1$. A *coding* from Y to X is a measurable function $\varphi \colon T^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$, which is *translation-equivariant,* i.e., commutes with every translation of $\mathbb{Z}^d$, and which satisfies that $\varphi(Y)$ and X are identical in distribution. Such a coding is also called a *factor map* or *homomorphism* from Y to X, and when such a coding exists, we say that X is a *factor* of Y.

The *coding radius* of φ at a point $y \in T^{\mathbb{Z}^d}$, denoted by $R(y)$, is the minimal integer $r \geq 0$ such that $\varphi(y')_{\mathbf{0}} = \varphi(y)_{\mathbf{0}}$ for almost all $y' \in T^{\mathbb{Z}^d}$ which coincide with y on the ball of radius r around the origin in the graph-distance, i.e., $y'_v = y_v$ for all $v \in \mathbb{Z}^d$ such that $\|v\|_1 \leq r$. It may happen that no such r exists, in which case, $R(y) = \infty$. Thus, associated to a coding is a random variable $R = R(Y)$ which describes the coding radius. We refer to R^d as the *coding volume*. A coding is called *finitary* if R is almost surely finite. When there exists a finitary coding from Y to X, we say that X is a *finitary factor* of Y.

---

*University of British Columbia, Canada. E-mail: yinon@math.ubc.ca

We say that a non-negative random variable R has *exponential tails* if $\mathbb{P}(R \geq r) \leq Ce^{-cr}$ for some $C, c > 0$ and all $r \geq 0$, and that it has *stretched-exponential tails* if $\mathbb{P}(R \geq r) \leq Ce^{-r^c}$ holds instead. When there exists a coding from Y to X whose coding radius has (stretched-)exponential tails, we say that X is a finitary factor of Y with (stretched-)exponential tails.

In this paper, we shall be concerned with finitary factors of i.i.d. (independent and identically distributed) processes, distinguishing between the cases when the i.i.d. process is finite-valued or infinite-valued. We use the abbreviation *ffiid* to denote a finitary factor of an i.i.d. process (perhaps infinite-valued), and *fv-ffiid* to denote a finitary factor of a finite-valued i.i.d. process.

Our main example is the (ferromagnetic) Ising model in $d \geq 2$ dimensions – a classical discrete spin system in statistical mechanics. A *Gibbs measure for the Ising model* on $\mathbb{Z}^d$ at inverse temperature $\beta > 0$ is a probability measure μ on $\{-1, +1\}^{\mathbb{Z}^d}$ which satisfies that, if the random field $X = (X_v)_{v \in \mathbb{Z}^d}$ has distribution μ, then for any vertex $v \in \mathbb{Z}^d$,

$$\mathbb{P}\Big(X_v = \pm 1 \mid X|_{\mathbb{Z}^d \setminus \{v\}}\Big) = \frac{\exp\Big[\pm \beta \sum_{u \in N(v)} X_u\Big]}{\exp\Big[\beta \sum_{u \in N(v)} X_u\Big] + \exp\Big[-\beta \sum_{u \in N(v)} X_u\Big]}, \tag{1.1}$$

almost surely, where $N(v)$ denotes the neighborhood of v. We note that although one usually defines Gibbs measures for the Ising model through their conditional distributions on any finite set, the above single-site specifications are sufficient as they determine the conditional finite-dimensional distributions.

It is well known (see, e.g., [8, Theorem 3.1] or [17, pages 189–190 and 204]) that there exists a critical value $\beta_c(d) \in (0, \infty)$ such that there is a unique Gibbs measure for the Ising model on $\mathbb{Z}^d$ at inverse temperature $\beta < \beta_c(d)$ and multiple such Gibbs measures at inverse temperature $\beta > \beta_c(d)$. Van den Berg and Steif [5] showed that the unique Gibbs measure in the former case is fv-ffiid. We improve upon this and answer a question from [5] by showing the following.

Theorem 1.1. *Let $d \geq 2$ and let μ be the unique Gibbs measure for the Ising model on $\mathbb{Z}^d$ at inverse temperature $\beta < \beta_c(d)$. Then μ is fv-ffiid with stretched-exponential tails.*

Remark 1.2. It has been shown in [5] that a phase transition (i.e., existence of multiple Gibbs measures) presents an obstruction for the existence of a finitary coding from an i.i.d. process. In particular, at inverse temperature $\beta > \beta_c(d)$, no translation-invariant Gibbs measure for the Ising model is ffiid.

Remark 1.3. In the course of proving that μ is fv-ffiid, it is shown in [5] that if one does not insist on a coding from a *finite-valued* i.i.d. process, then one may obtain a coding with exponential tails (so that μ is ffiid with exponential tails); see Section 2 for more details. Similarly, it also follows that the critical Ising measure (i.e., when $\beta = \beta_c(d)$) is ffiid (this relies on the fact that the phase transition is continuous [25, 3, 2]), though it is shown in [5] (a result which was obtained jointly with Peres) that the coding volume R^d cannot have finite expectation.

Remark 1.4. It is still unknown whether the critical Ising measure is fv-ffiid.

Remark 1.5. In dimension $d = 1$, the Ising model has a unique Gibbs measure μ at any finite inverse temperature β, and this measure is the distribution of an ergodic stationary Markov chain. It follows from a result in [12] that μ is fv-ffiid with exponential tails (in fact, μ is a finitary factor with exponential tails of any i.i.d. process with entropy strictly larger than that of μ, and it is finitarily isomorphic to any i.i.d. process with equal entropy [16]).

Remark 1.6. The FK random-cluster model is a dependent percolation model with infinite-range interactions, which is closely related to the Ising model. For background on

this model, see [9]. Using the Edwards–Sokal coupling [23, 7], it is an easy consequence of Theorem 1.1 that the sub-critical FK-Ising measure (i.e., the random-cluster measure with parameters $q = 2$ and any $p < p_c(q)$) is fv-ffiid with stretched-exponential tails. Indeed, under this coupling, given the Ising configuration, the state of the edges in the random-cluster configuration are independent. It is shown in [11] that this result extends to the random-cluster model with any $q \geq 1$ and $p < p_c(q)$.

Our second result concerns another well-known model of statistical mechanics – proper colorings. Let $q \geq 3$. A *proper q-coloring* of $\mathbb{Z}^d$ is a configuration $x \in \{1, \ldots, q\}^{\mathbb{Z}^d}$ satisfying that $x_u \neq x_v$ for any adjacent vertices u and v. A *Gibbs measure for proper q-colorings* of $\mathbb{Z}^d$ is a probability measure μ on $\{1, \ldots, q\}^{\mathbb{Z}^d}$ which is supported on proper q-colorings and satisfies that, if the random field $X = (X_v)_{v \in \mathbb{Z}^d}$ has distribution μ, then for any finite set $\Lambda \subset \mathbb{Z}^d$, the conditional distribution of X given its restriction to Λ^c is uniform on the set of proper q-colorings which agree with X on Λ^c. It is well known (e.g., by Dobrushin uniqueness [22]) that there is a unique Gibbs measure for proper q-colorings when $q > 4d$. We show that this measure is fv-ffiid with stretched-exponential tails when the number of colors is large enough.

Theorem 1.7. *Let $d \geq 2$ and $q \geq 4d(d+1)$. Let μ be the unique Gibbs measure for proper q-colorings of $\mathbb{Z}^d$. Then μ is fv-ffiid with stretched-exponential tails.*

Remark 1.8. The uniform proper q-coloring model is equivalent to the zero-temperature anti-ferromagnetic q-state Potts model. It is intuitively clear that increasing the temperature only makes interactions weaker (the high-temperature model even satisfies high-noise; see below), and indeed, we believe that Theorem 1.7 extends to the anti-ferromagnetic q-state Potts model at any temperature (and q as in the theorem above), though we do not pursue this here.

Our third result is not about a particular model, but rather about a class of translation-invariant high-noise Markov random fields, which we proceed to define. Let S be finite, let μ be a probability measure on $S^{\mathbb{Z}^d}$ and let $X = (X_v)_{v \in \mathbb{Z}^d}$ be distributed according to μ. We say that μ is a *Markov random field* if its conditional finite-dimensional distributions depend only on the immediate neighborhood of the finite set being inspected, i.e., if for any finite $V \subset \mathbb{Z}^d$ and any $\xi \in S^V$,

$$\mathbb{P}\Big(X|_V = \xi \mid X|_{\mathbb{Z}^d \setminus V}\Big) = \mathbb{P}\Big(X|_V = \xi \mid X|_{\partial V}\Big) \qquad \text{almost surely,}$$

where ∂V denotes the set of vertices at distance 1 from V. The Ising model and proper colorings (or rather the Gibbs measures for those models) are two examples of Markov random fields.

Suppose that μ is a translation-invariant Markov random field and, for $s \in S$, denote

$$\gamma_s := \inf_{\substack{\xi \in S^{N(\mathbf{0})} \\ \mathbb{P}(X|_{N(\mathbf{0})} = \xi) > 0}} \mathbb{P}\Big(X_{\mathbf{0}} = s \mid X|_{N(\mathbf{0})} = \xi\Big).$$

We say that μ satisfies *high-noise* if

$$\gamma := \sum_{s \in S} \gamma_s > 1 - \frac{1}{2d}.$$

The quantity γ is called the multigamma admissibility. It is essentially the probability that an update can be made to the spin at the origin without knowing anything about the values of the spins at its neighbors (see [10] for a more detailed explanation). We remark that Dobrushin's uniqueness condition [6] (or, alternatively, the "disagreement percolation" condition of van den Berg and Maes [4]) implies that if μ satisfies high-noise, then it is the only random field with the same conditional finite-dimensional distributions as μ.

Theorem 1.9. *Let μ be a translation-invariant Markov random field satisfying high-noise. Then μ is fv-ffiid with stretched-exponential tails.*

Theorem 1.9 improves on a result of Häggström and Steif [10] who showed that any translation-invariant high-noise Markov random field is fv-ffiid. Theorem 1.9 applies to numerous models of statistical physics, including the Potts model (both ferromagnetic and anti-ferromagnetic) at high temperature, the hard-core model at low fugacity and the Widom–Rowlinson model at low fugacity (see [10] for more details on this for the Potts and Widom–Rowlinson models). On the other hand, Theorem 1.1 does not follow from Theorem 1.9, as the Ising model does not satisfy high-noise when β is only slightly smaller than $\beta_c(d)$. Similarly, Theorem 1.7 does not follow from Theorem 1.9 (even for large values of q), as it is clear that $\gamma = 0$ for proper q-colorings, regardless of how large q is.

The three theorems will be proved using a general result introduced in Section 2 about finitary codings for limiting distributions of probabilistic cellular automata.

Background. We give here only a brief background and refer the reader to [5] for a more complete description of known results. A fundamental problem in ergodic theory is to understand which processes are isomorphic to which other processes (meaning that there is an almost everywhere invertible factor from one to the other). The very simplest of processes are the i.i.d. processes, and therefore, of particular interest are those processes which are isomorphic to an i.i.d. process; such processes are termed Bernoulli. The celebrated isomorphism theorem of Ornstein [19] states that any two i.i.d. processes of equal entropy are isomorphic (this result was later extended by Keane and Smorodinsky [15] who showed that any two such finite-valued processes are in fact finitarily isomorphic). Ornstein [19] further showed that any factor of an i.i.d. process is Bernoulli. This shed a more probabilistic light on the notion of Bernoullicity.

The notion of a finitary factor of i.i.d. has the advantage that it allows to compute a symbol in the target process by only revealing (almost surely) finitely many variables of the i.i.d. process. This gives a more concrete construction of the target process, which may also be useful for exact simulation algorithms. Besides this appealing feature, finitary factors of i.i.d. have particular relevance in the context of probabilistic models such as those considered here. Let us take the Ising model as an example. It has been shown [20] (see also [1]) that the so-called "plus state" (this is the Gibbs measure obtained by taking $+$ boundary conditions) is a factor of an i.i.d. process (i.e., is Bernoulli) for any value of the inverse temperature β. Thus, the phase transition is not reflected in this notion. However, as shown in [5], it is indeed reflected in the notion of a *finitary* factor: the "plus state" is a finitary factor of an i.i.d. process when $\beta < \beta_c(d)$, but not when $\beta > \beta_c(d)$.

In constructing a finitary coding from an i.i.d. process to a given process, it is desirable for efficiency purposes (e.g., for simulation algorithms) that the i.i.d. process be "small" and that the coding radius also be typically small. One such qualitative meaning of this is that the i.i.d. process is finite-valued and that the coding volume has finite expectation. A more quantitative meaning of this would be to require bounds on the entropy of the i.i.d. process and on the tail of the coding radius. Our results are a mixture of the two as they yield a finitary coding from a finite-valued i.i.d. process with stretched-exponential tails for the coding radius. In particular, our result about the Ising model (Theorem 1.1) answers a question of van den Berg and Steif [5, Question 2], who asked whether the sub-critical Ising measure is fv-ffiid with finite expected coding volume.

Notation. We consider $\mathbb{Z}^d$ as a graph in which two vertices u and v are adjacent if $|u - v| = 1$, where $|v| = \|v\|_1 := |v_1| + \cdots + |v_d|$ denotes the ℓ_1-norm. We denote

by $N(v) := \{u \in \mathbb{Z}^d : |u - v| = 1\}$ the neighborhood of v. For a set $U \subset \mathbb{Z}^d$, we write $\mathrm{dist}(v, U) := \min_{u \in U} |u - v|$. We use $\mathbf{0}$ to denote the origin $(0, \ldots, 0) \in \mathbb{Z}^d$ and $e_1 := (1, 0, \ldots, 0) \in \mathbb{Z}^d$. We use $\mathbb{N}$ to denote the non-negative integers.

Organization. In Section 2, we formulate the result about finitary codings for limiting distributions of probabilistic cellular automata (Theorem 2.1) and use it to prove Theorem 1.1, Theorem 1.7 and Theorem 1.9. In Section 3, we introduce an abstract tool (Proposition 3.1 and the more general Proposition 3.2) and show how to deduce Theorem 2.1 from it. In Section 4, we introduce and explain an algorithm, which is then used in Section 5 to prove Proposition 3.2. We end with open questions in Section 6.

2 Finitary codings for limiting distributions of PCAs

The goal of this section is to define the notion of a *probabilistic cellular automaton (PCA)* and other relevant notions, formulate a general result about finitary codings for limiting distributions of PCAs (Theorem 2.1 below), and then use this theorem to deduce the results stated in Section 1.

Before doing so, we give an informal description of the relevant ideas and concepts in the case of the Ising model: Consider the continuous-time Glauber dynamics for the sub-critical Ising model – each vertex has an exponential clock (with rate 1), and when its clock rings, it updates its spin value according to the conditional distribution given by the values of its neighbors as in (1.1). This is an ergodic process, whose unique stationary measure is μ (of Theorem 1.1), and thus, the distribution at time t converges to μ as $t \to \infty$, regardless of the initial configuration. As we are interested in finding a coding from a finite-valued process, we instead opt to use a discrete analogue of these dynamics, given by a PCA: at each discrete time step n, every vertex is independently set to active or inactive with some fixed probability, and every active vertex which has no active neighbors then updates its spin value as before. This too is an ergodic process and the distribution at time n converges to μ as $n \to \infty$. Convergence alone is not sufficient to obtain a coding of μ, as the latter requires an exact sample from μ. To get such a sample, one can employ the coupling-from-the-past technique of Propp and Wilson [21]. This then yields a finitary coding for μ from an infinite-valued i.i.d. process (showing that μ is ffiid). Using a result of Martinelli and Olivieri [18] that the convergence of the above process to stationarity occurs at an exponential rate, one may further show that this coding has a coding radius with exponential tails (showing that μ is ffiid with exponential tails). To get from this a (finitary) coding from a *finite-valued* i.i.d. process, still requires quite some work. All the above, including this last step, has been carried out by van den Berg and Steif [5]. Thus, they showed that μ is fv-ffiid. However, they gave no information on the coding radius beyond its almost sure finiteness. Our main contribution is to show how one can carry out this last step in a controlled manner which preserves the good tails of the coding radius (yielding stretched-exponential tails). We elaborate on this in the next sections.

The above includes general arguments about certain dynamics, along with some model-specific information. Indeed, van den Berg and Steif separated the two parts of the argument, and proved the more general result [5, Theorem 3.4] that the limiting distribution of a monotone, exponentially ergodic PCA is fv-ffiid. In order to accommodate for the different situations considered in Section 1, which include non-monotone models (proper colorings and high-noise Markov random fields), we work here in the more general setting of exponentially uniformly ergodic PCAs (instead of monotone, exponentially ergodic PCAs), defined below. The proof of [5, Theorem 3.4] may be extended to this setting to show that the limiting distribution of an exponentially uniformly ergodic PCA is fv-ffiid. As mentioned before, the main challenge, and our primary contribution, is to

show that this can be done while simultaneously controlling the coding radius.

Theorem 2.1. *The limiting distribution of an exponentially uniformly ergodic PCA is fv-ffiid with stretched-exponential tails.*

The results of Section 1 will follow from Theorem 2.1 by showing that the corresponding measures are limiting distributions of exponentially uniformly ergodic PCAs. Theorem 2.1 will be proved in Section 3. Let us also mention the following result which will easily follow from our definition of an exponentially uniformly ergodic PCA (unlike Theorem 2.1 which requires work).

Theorem 2.2. *The limiting distribution of an exponentially uniformly ergodic PCA is ffiid with exponential tails.*

We emphasize the differences between the two theorems: the second gives a finitary coding with exponential tails but does not provide any control on the i.i.d. process, while the first gives a coding from a finite-valued i.i.d. process but does slightly worse in terms of the tails of the coding radius.

Let us now proceed to give precise definitions. We begin by defining what a *PCA* is. For our purposes, a PCA is a discrete-time evolution on $S^{\mathbb{Z}^d}$ for some non-empty finite set S, which can be described as follows. Let $(W_{v,i})_{v \in \mathbb{Z}^d, i \in \mathbb{Z}}$ be a collection of i.i.d. random variables taking values in a finite set A. Let $F, F' \subset \mathbb{Z}^d$ be finite and let $f \colon S^F \times A^{F'} \to S$. The time evolution started from $\xi \in S^{\mathbb{Z}^d}$ is the process $(\omega_{v,i})_{v \in \mathbb{Z}^d, i \geq 0}$ defined by

$$
\begin{aligned}
\omega_{v,0} &:= \xi_v, & v \in \mathbb{Z}^d, \\
\omega_{v,i+1} &:= f\big((\omega_{v+u,i})_{u \in F}, (W_{v+u,i})_{u \in F'}\big), & v \in \mathbb{Z}^d,\ i \geq 0.
\end{aligned}
\tag{2.1}
$$

We stress that different choices of $W_{v,i}$ and f could give rise to the same time evolutions (i.e., the same distribution), however, for our purposes, a PCA is the data of the distribution of the $W_{v,i}$, the sets F and F' and the function f. In particular, we note that a PCA comes equipped with a simultaneous coupling of the time evolutions started from all starting states ξ. We remark that the usual definition of a PCA requires that $F' = \{0\}$, in which case, conditioned on $\{\omega_{v,i}\}_v$, the random variables $\{\omega_{v,i+1}\}_v$ are mutually independent. For the above approach to the construction of finitary codings, we will have $F = F' = N(0) \cup \{0\}$ (recall that $N(0)$ is the neighborhood of the origin), in which case, there are local conditional dependencies.

A PCA is said to be *ergodic* if there exists a probability measure μ on $S^{\mathbb{Z}^d}$ such that, for any starting state ξ, the distribution of $(\omega_{v,i})_{v \in \mathbb{Z}^d}$ converges weakly to μ as $i \to \infty$. An ergodic PCA converging to μ can be used to obtain an approximate sample from $\mu|_\Lambda$, the marginal of μ on a finite subset Λ of $\mathbb{Z}^d$, by running the time evolution of the PCA until some large time t and observing the restricted process $(\omega_{v,t})_{v \in \Lambda}$ at that time, noting also that the latter is determined by a finite collection of random variables, namely,

$$
(\omega_{v,t})_{v \in \Lambda} \text{ is determined by } \xi \text{ and } \{W_{v,t-i}\}_{\mathrm{dist}(v,\Lambda) \leq \Delta i, 1 \leq i \leq t}, \qquad \text{where } \Delta := \max_{u \in F \cup F'} \|u\|_1.
\tag{2.2}
$$

As is usual in these situations, determining how large t should be in order to obtain a sample whose distribution is close to the limiting distribution, is not an easy task.

One way around this is to devise a method to exactly sample from the limiting distribution. Coupling-from-the-past provides such a method, at the cost, however, of requiring a type of uniform ergodicity. To define this notion, we first extend the definition given in (2.1) of the time evolution of the PCA to allow starting at any integer time as follows. The time evolution started from $\xi \in S^{\mathbb{Z}^d}$ at time $i_0 \in \mathbb{Z}$ is the process

$(\omega_{v,i}^{\xi,i_0})_{v \in \mathbb{Z}^d, i \geq i_0}$ defined by

$$
\begin{aligned}
\omega_{v,i_0}^{\xi,i_0} &:= \xi_v, & v \in \mathbb{Z}^d, \\
\omega_{v,i+1}^{\xi,i_0} &:= f\big((\omega_{v+u,i}^{\xi,i_0})_{u \in F}, (W_{v+u,i})_{u \in F'}\big), & v \in \mathbb{Z}^d, \ i \geq i_0.
\end{aligned}
\tag{2.3}
$$

We say that an ergodic PCA is *uniformly ergodic* if

$$
\tau_v := \min\big\{i \geq 0 : \omega_{v,0}^{\xi,-i} \text{ does not depend on the starting state } \xi\big\}
\tag{2.4}
$$

is almost surely finite for all v. We remind the reader that in our definitions, a PCA always comes equipped with a function f, so that the notion of uniform ergodicity depends on this f. While this might not be the standard notion of uniform ergodicity, it will be the relevant one for us. We also remark that for monotone PCAs, ergodicity implies uniform ergodicity (see [5, Lemma 3.5]). We say that an ergodic PCA is *exponentially uniformly ergodic* if τ_v has exponential tails. For a uniformly ergodic PCA, we define the random field $\omega^* = (\omega_v^*)_{v \in \mathbb{Z}^d}$ by

$$
\omega_v^* := \omega_{v,0}^{\xi,-\tau_v}, \qquad v \in \mathbb{Z}^d,
\tag{2.5}
$$

noting that this is almost surely well-defined and does not depend on ξ. We point out that while earlier we needed $W_{v,i}$ with $i \geq 0$, for (2.4) and (2.5) we use $W_{v,i}$ with $i < 0$.

The following proposition encompasses the essence of coupling-from-the-past (in its infinite-volume version). An analogous statement for monotone ergodic PCAs was shown in [5] and a similar statement for PCAs arising from high-noise Markov random fields was shown in [10]. The proofs of these statements are easily adapted to the setting described here, and we include a short proof for completeness.

Proposition 2.3. *Suppose μ is the limiting distribution of a uniformly ergodic PCA having time evolution ω. Then ω^* has distribution μ.*

Proof. Fix a finite $\Lambda \subset \mathbb{Z}^d$ and denote $\Omega^t := (\omega_{v,0}^{\xi,-t})_{v \in \Lambda}$. Note that, since $\omega_{v,0}^{\xi,-\tau_v} = \omega_v^*$ for all ξ, it follows from (2.3) that $\omega_{v,0}^{\xi,-t} = \omega_v^*$ for all $t \geq \tau_v$. In particular, $\Omega^t = \omega^*|_\Lambda$ for all $t \geq \max_{v \in \Lambda} \tau_v$. Since $(\omega_{v,0}^{\xi,-t})_{v \in \mathbb{Z}^d}$ and $(\omega_{v,t}^{\xi,0})_{v \in \mathbb{Z}^d}$ are identical in distribution for any $t \geq 0$, it follows that Ω^t converges to $\mu|_\Lambda$ in distribution as $t \to \infty$. On the other hand, as we have seen that Ω^t eventually equals $\omega^*|_\Lambda$, we conclude that $\omega^*|_\Lambda$ has distribution $\mu|_\Lambda$. Since Λ was arbitrary, the proposition follows. $\qquad\square$

The proposition implies that the limiting distribution of a uniformly ergodic PCA is ffiid. Indeed, a moment of thought reveals that (2.3)–(2.5) describe such a finitary coding from the process $((W_{v,i})_{i<0})_{v \in \mathbb{Z}^d}$. Moreover, if the PCA is exponentially uniformly ergodic, then the coding radius of this coding has exponential tails, so that the limiting distribution is in fact ffiid with exponential tails. This establishes Theorem 2.2. The point is, however, that this coding is not from a finite-valued process. Restricting the i.i.d. process to be finite-valued, while keeping control of the coding radius, is the missing step in order to establish Theorem 2.1 and is what most of the remainder of the paper is devoted to. Before coming back to this in the next section, we explain how to deduce the results of Section 1 from Theorem 2.1.

We now prove Theorem 1.1, Theorem 1.7 and Theorem 1.9. In light of Theorem 2.1, this boils down to showing that in each case the corresponding measure is the limiting distribution of an exponentially uniformly ergodic PCA.

2.1 The Ising model – proof of Theorem 1.1

To deduce from Theorem 2.1 that the sub-critical Ising measure is fv-ffiid with stretched-exponential tails, we must know that it is the limiting distribution of an exponentially uniformly ergodic PCA. This was shown by van den Berg and Steif (see the proof

of Theorem 4.1 in [5]) who relied on a deep result of Martinelli and Olivieri [18] about the continuous-time Glauber dynamics for the Ising model (see also [5, Proposition 4.2]).

Proposition 2.4 ([5]). *Let $d \geq 2$ and let μ be the unique Gibbs measure for the Ising model on $\mathbb{Z}^d$ at inverse temperature $\beta < \beta_c(d)$. Then μ is the limiting distribution of an exponentially uniformly ergodic PCA.*

Theorem 1.1 follows immediately from Theorem 2.1 and Proposition 2.4.

Although for our purposes, we only need to know that a PCA as in Proposition 2.4 exists and its details are not important for us, in order to provide the reader with a full picture for the case of the Ising model, we nevertheless give a complete and formal description of the PCA used in the proof of Proposition 2.4 (but written in a slightly different way than in [5]). In fact, we have already given an informal description of this PCA in the beginning of Section 2. To define it precisely, set $S := \{-1, 1\}$, $A := \{0, 1\} \times \{-2d, \ldots, 2d + 1\}$, $F = F' := N(\mathbf{0}) \cup \{\mathbf{0}\}$ and define $f \colon S^F \times A^F \to S$ by

$$f(\eta, (\phi, \psi)) := \begin{cases} 2 \cdot \mathbb{1}_{\left\{ \psi_{\mathbf{0}} \leq \sum_{u \in N(\mathbf{0})} \eta_u \right\}} - 1 & \text{if } \phi_{\mathbf{0}} = 1 \text{ and } \phi_v = 0 \text{ for all } v \in N(\mathbf{0}) \\ \eta_0 & \text{otherwise.} \end{cases}$$

To complete the description of the PCA, we must also describe the distribution of the i.i.d. random variables $(W_{v,i})_{v \in \mathbb{Z}^d, i \in \mathbb{Z}}$. We let each $W_{v,i}$ consist of a pair of independent random variables, the first of which is a Bernoulli random variable with parameter, say, $1/2$, and the second of which has the distribution of $\mathcal{W}$, where $\mathcal{W}$ takes values in $\{-2d, \ldots, 2d + 1\}$ and satisfies

$$\mathbb{P}(\mathcal{W} \leq k) = p_k := \frac{e^{\beta k}}{e^{\beta k} + e^{-\beta k}} \qquad \text{for } -2d \leq k \leq 2d.$$

Observe that such a random variable exists since $(p_k)_{-2d \leq k \leq 2d}$ is increasing. Recalling (1.1), one may easily verify that any Gibbs measure for the Ising model on $\mathbb{Z}^d$ at inverse temperature β is a stationary measure for this PCA.

We note that this PCA is monotonic in the sense that $f(\eta, (\phi, \psi)) \leq f(\eta', (\phi, \psi))$ for any (ϕ, ψ) and (η, η') such that $\eta_v \leq \eta'_v$ for all $v \in F$, and we remark that due to this monotonicity, Proposition 2.4 is essentially a statement about the probability that the value of the spin at the origin after time t depends on whether the starting state is the constant plus or constant minus state – namely, that this probability is exponentially small in t.

2.2 High-noise Markov random fields – proof of Theorem 1.9

To deduce Theorem 1.9 from Theorem 2.1, we need to know that a translation-invariant high-noise Markov random field is the limiting distribution of an exponentially uniformly ergodic PCA. This was shown by Häggström and Steif in [10] (essentially Proposition 2.1 there).

Proposition 2.5 ([10]). *Let μ be a translation-invariant Markov random field satisfying high-noise. Then μ is the limiting distribution of an exponentially uniformly ergodic PCA.*

Given this proposition, Theorem 1.9 is an immediate corollary of Theorem 2.1.

2.3 Proper colorings – proof of Theorem 1.7

Theorem 1.7 will follow from Theorem 2.1 once we establish the following.

Proposition 2.6. *Let $d \geq 2$ and $q \geq 4d(d + 1)$. Let μ be the unique Gibbs measure for proper q-colorings of $\mathbb{Z}^d$. Then μ is the limiting distribution of an exponentially uniformly ergodic PCA.*

Proof. The proof uses ideas of Huber [13, 14] for exact sampling of proper colorings on finite graphs and ideas of Häggström and Steif [10] from the proof of Proposition 2.5. The proof of the latter proposition uses an auxiliary PCA on a larger space (which the authors there call a super-PCA), which "bounds" the original PCA simultaneously for all starting states, and thus allows to "detect" when the original PCA has coalesced. Huber used a similar idea (which he called bounding chains), together with model-specific arguments, to provide an exact sampling algorithm for proper colorings (and other models) on a finite graph. Putting these ideas together, we show how this can be done for proper colorings of $\mathbb{Z}^d$.

We first describe the PCA in words: at each time step, every vertex is independently set to active or inactive with some fixed probability, and every active vertex which has no active neighbors then updates its color to be uniformly chosen from the set of colors not appearing at any of its neighbors. More precisely, a uniform permutation of the colors is chosen, and the first color not appearing at any neighbor is chosen. This PCA may be realized as follows. Let $S := \{1, \ldots, q\}$ and let $\mathcal{S}_q$ be the symmetric group on S. Let $F = F' := N(\mathbf{0}) \cup \{\mathbf{0}\}$, $A := \{0, 1\} \times \mathcal{S}_q$ and define $f \colon S^F \times A^F \to S$ by

$$f(\eta, (\phi, \psi)) := \begin{cases} g(\{\eta_v\}_{v \in N(\mathbf{0})}, \psi_{\mathbf{0}}) & \text{if } \phi_{\mathbf{0}} = 1 \text{ and } \phi_v = 0 \text{ for all } v \in N(\mathbf{0}) \\ \eta_{\mathbf{0}} & \text{otherwise,} \end{cases}$$

where $g \colon 2^S \times \mathcal{S}_q \to S$ is defined by

$$g(D, \pi) := \pi(\min\{i \in S : \pi(i) \notin D\}).$$

The time evolution ω of this PCA is then given by (2.3), where the i.i.d. random variables $(W_{v,i})$ are chosen to be uniformly distributed over A, so that each $W_{v,i}$ represents an unbiased coin toss (the unbiasedness will not be important for us) and an independent uniformly chosen permutation of the colors. It is straightforward to check that any Gibbs measure for proper q-colorings is a stationary distribution for this PCA.

To show that this PCA is exponentially uniformly ergodic, we use the method of bounding chains discussed above. Consider the following PCA (or super-PCA in the language of [10]) on $(2^S)^{\mathbb{Z}^d}$ given by $\hat{f} \colon (2^S)^F \times A^F \to 2^S$, where

$$\hat{f}(\hat{\eta}, (\phi, \psi)) := \begin{cases} \hat{g}(\hat{\eta}|_{N(\mathbf{0})}, \psi_{\mathbf{0}}) & \text{if } \phi_{\mathbf{0}} = 1 \text{ and } \phi_v = 0 \text{ for all } v \in N(\mathbf{0}) \\ \hat{\eta}_{\mathbf{0}} & \text{otherwise,} \end{cases}$$

where $\hat{g} \colon (2^S)^{N(\mathbf{0})} \times \mathcal{S}_q \to S$ is defined by

$$\hat{g}(\hat{\eta}, \pi) := \bigcup_{\substack{\eta \in S^{N(\mathbf{0})} : \\ \eta_v \in \hat{\eta}_v \ \forall v \in N(\mathbf{0})}} g(\{\eta_v\}_{v \in N(\mathbf{0})}, \pi).$$

The time evolution $\hat{\omega}$ of this PCA is then defined as in (2.3), using the same random variables $(W_{v,i})$ as above, so that the two PCAs are coupled, with the crucial property that $\hat{\omega}$ bounds ω in the sense that

$$\omega_{v,j}^{\xi, i} \in \hat{\omega}_{v,j}^{\hat{\xi}, i} \qquad \text{for any } \xi \in S^{\mathbb{Z}^d}, \ v \in \mathbb{Z}^d \text{ and } i \leq j,$$

where $\hat{\xi}$ is the maximal element in $(2^S)^{\mathbb{Z}^d}$ defined by $\hat{\xi}_v := 2^S$ for all $v \in \mathbb{Z}^d$. In particular, recalling (2.4), we have

$$\tau_v \leq \hat{\tau}_v = \min\{i \geq 0 : |\hat{\omega}_{v,0}^{\hat{\xi}, -i}| = 1\}.$$

It therefore suffices to show that $\hat{\tau}_v$ has exponential tails.

We begin by observing that, for $t \geq 0$,

$$\mathbb{P}(\hat{\tau}_v > t) \leq \mathbb{P}\left(|\hat{\omega}_{v,0}^{\hat{\xi},-t}| > 1\right) = \mathbb{P}\left(|\hat{\omega}_{v,t}^{\hat{\xi},0}| > 1\right) =: p_t,$$

where p_t is of course independent of v. To ease notation, let us denote $Y_t(v) := \hat{\omega}_{v,t}^{\hat{\xi},0}$. Let α denote the probability that a vertex is updated in any given time step, i.e., $\alpha = \beta(1-\beta)^{2d}$, where β is the probability that a vertex is activated (for an unbiased coin toss, we have $\beta = 1/2$ and $\alpha = 2^{-2d-1}$, but this will not be used). Note that $|\hat{g}(\hat{\eta}, \pi)| \leq 2d + 1$ for any $\hat{\eta} \in (2^S)^{N(\mathbf{0})}$ and $\pi \in \mathcal{S}_q$. Hence,

$$\mathbb{P}(|Y_t(u)| > 2d + 1) = (1-\alpha)^t \quad \text{for any } u \in \mathbb{Z}^d \text{ and } t \geq 0, \tag{2.6}$$

since $|Y_t(u)| > 2d + 1$ if and only if u has never been updated by time t.

Let us see what happens when the origin is updated. Let $D := \bigcup_{u \in N(\mathbf{0})} Y_t(u)$ be the set of colors which may appear in some neighbor of $\mathbf{0}$, and let $D' := \bigcup_{u \in N(\mathbf{0}), |Y_t(u)|=1} Y_t(u)$ be those colors which are known to appear in some neighbor of $\mathbf{0}$. Observe that if $\pi \in \mathcal{S}_q$ is such that $g(D', \pi) \notin D$, then $g(D, \pi) = g(D', \pi)$ and $\hat{g}((Y_t)|_{N(\mathbf{0})}, \pi) = \{g(D', \pi)\}$. Thus, given Y_t and given that $\mathbf{0}$ is updated at time $t + 1$, the probability that $|Y_{t+1}(\mathbf{0})| > 1$ is at most the probability that the $g(D', \pi) \in D$. When $\pi \in \mathcal{S}_q$ is chosen uniformly, $g(D', \pi)$ is uniformly distributed in $S \setminus D'$, so that the latter probability is $\frac{|D \setminus D'|}{q - |D'|} \leq \frac{\sum_{u \in N(\mathbf{0})} |Y_t(u)| \mathbb{1}_{\{|Y_t(u)| > 1\}}}{q - 2d + 1}$. This shows that

$$\mathbb{P}(|Y_{t+1}(\mathbf{0})| > 1 \mid Y_t) \leq (1-\alpha)\mathbb{1}_{\{|Y_t(\mathbf{0})| > 1\}} + \alpha \sum_{u \in N(\mathbf{0})} \frac{|Y_t(u)| \mathbb{1}_{\{|Y_t(u)| > 1\}}}{q - 2d + 1}.$$

Together with (2.6), this yields

$$p_{t+1} = \mathbb{E}\left[\mathbb{P}(|Y_{t+1}(\mathbf{0})| > 1 \mid Y_t)\right] \leq 2d(1-\alpha)^t + \left(1 - \alpha\left(1 - \frac{2d(2d+1)}{q - 2d + 1}\right)\right) p_t.$$

Thus, p_t decays exponentially in t when $q \geq 4d(d+1)$, and Proposition 2.6 follows. □

3 A general result and proof of Theorem 2.1

In this section, we introduce a general result which will allow us to deduce Theorem 2.1. This result is an abstract tool and is not, a priori, related to the problems originally discussed in Section 1.

Let $X = (X_{v,i})_{v \in \mathbb{Z}^d, i \geq 0}$ be a process taking values in a finite set S. Let $B = (B_n)_{n \geq 0}$ be a strictly increasing sequence of subsets of $\mathbb{Z}^d \times \mathbb{N}$ with $B_0 := \{(\mathbf{0}, 0)\}$, and consider the associated σ-algebras $\{\mathcal{F}_v^n\}_{v \in \mathbb{Z}^d, n \geq 0}$ defined by

$$\mathcal{F}_v^n := \sigma\left(\{X_{v+u,i}\}_{(u,i) \in B_n}\right). \tag{3.1}$$

An $\mathbb{N}$-valued random field $\tau = (\tau_v)_{v \in \mathbb{Z}^d}$ is said to be a B-*stopping-process* for X if, for every v, τ_v is an almost surely finite stopping time with respect to the filtration $(\mathcal{F}_v^n)_{n \geq 0}$. When we say that such a stopping-process is stationary, we shall mean that the same stopping rule is used at every vertex (rather than just meaning that its law is translation-invariant). Given a B-stopping-process τ, we denote by X^τ the random field

$$X^\tau := \left((X_{v+u,i})_{(u,i) \in B_{\tau_v}}\right)_{v \in \mathbb{Z}^d}.$$

Note that $(X^\tau)_v$ takes values in the finite-configuration space $\bigcup_{n \geq 0} S^{B_n}$. We say that B is *linear* if

$$\Delta_n := \max\left\{\max\{|u|, i\} : (u, i) \in B_n\right\} \leq \Delta n \quad \text{for some } \Delta \geq 1 \text{ and all } n \geq 0. \tag{3.2}$$

Proposition 3.1. *Let $X = (X_{v,i})_{v \in \mathbb{Z}^d, i \geq 0}$ be a finite-valued i.i.d. process, let B be linear and let τ be a stationary B-stopping-process for X. Suppose τ_v has exponential tails and $\mathbb{E}|B_{\tau_v}| < M$ for some integer M. Then X^{τ} is a finitary factor of $((X_{v,i})_{0 \leq i < M})_{v \in \mathbb{Z}^d}$ with stretched-exponential tails.*

Before using Proposition 3.1 to prove Theorem 2.1, we briefly explain the proposition and how it relates to the setting of the theorem. Recall that, given a uniformly ergodic PCA, (2.3)–(2.5) explicitly express the random field ω^* as a finitary factor of the i.i.d. process $((W_{v,i})_{i<0})_{v \in \mathbb{Z}^d}$, defined via certain stopping times. Moreover, it is clear from this and from (2.2) that the value of the output ω_u^* for any given u depends only on the variables $W_{v,i}$ within a certain "cone" in space-time emanating from $(u, 0)$ (this is because as one goes back in time, the spatial dependency grows linearly). The above setup generalizes this situation to an abstract setting (which has nothing to do with coupling-from-the-past or PCAs), where the sequence $(B_n)_n$ replaces the cones arising from (2.2), the stopping process replaces the coupling-from-the-past stopping times given in (2.4), and the variables $(W_{v,i})_{v \in \mathbb{Z}^d, i<0}$ are now called $(X_{v,i})_{v \in \mathbb{Z}^d, i \geq 0}$. With this interpretation in mind, for any given u, we may think of $(X^{\tau})_u$ as containing all the variables that are "needed" for the computation of the output at u, and the proposition states (ignoring the tails of τ_v and the coding) that if, on average, the number of variables needed to compute the output at a given vertex is less than M, then one can "emulate" the process X^{τ} (consisting of all the needed variables) from a process which has precisely M variables at each vertex. In other words, if one has an algorithm which can a priori need access to any number of variables at a given vertex, but typically does not need many such variables, then by "transporting" variables from one space-time location to another as needed, it is possible to rewrite the algorithm in such a way that it only has access to a bounded number of variables at each vertex. We note that we continue to refer to $\mathbb{Z}^d \times \mathbb{N}$ as space-time, although the interpretation of $\mathbb{N}$ as a time dimension is perhaps less proper.

Proof of Theorem 2.1. Suppose that μ is the limiting distribution of an exponentially uniformly ergodic PCA with time evolution ω, defined via variables $(W_{v,i})$, sets F and F', and function f. By Proposition 2.3, it suffices to show that ω^*, defined by (2.5), is fv-ffiid with stretched-exponential tails.

Recall the definition of τ_v from (2.4) and the definition of Δ from (2.2). By definition of τ_v and (2.2) (or rather the analogue of (2.2) for the time evolution started at time $-t$ and run up to time 0), the value of ω_v^* is a deterministic function of the variables $(W_{v+u,-i})_{|u| \leq \Delta i, 0 \leq i \leq \tau_v}$ (actually, the variable $W_{v,0}$ corresponding to $i = 0$ is not needed, but we include it nevertheless). Moreover, this function does not depend on v, in the sense that, for some deterministic function ψ, we have that $\omega_v^* = \psi((W_{v+u,-i})_{|u| \leq \Delta i, 0 \leq i \leq \tau_v})$ for all v.

Towards applying Proposition 3.1, define $B = (B_n)_{n \geq 0}$ by $B_n := \{(u, i) : |u| \leq \Delta i, 0 \leq i \leq n\}$ and define the i.i.d. process $X = (X_{v,i})_{v \in \mathbb{Z}^d, i \geq 0}$ by $X_{v,i} := W_{v,-i}$. Note that $\tau = (\tau_v)_{v \in \mathbb{Z}^d}$ is a linear stationary B-stopping-process for X, and that $\omega_v^* = \psi((X^{\tau})_v)$ so that ω^* is a finitary factor of X^{τ} with coding radius 0. It therefore suffices to show that X^{τ} is fv-ffiid with stretched-exponential tails. Indeed, letting M be any integer larger than $\mathbb{E}|B_{\tau_v}|$, Proposition 3.1 yields that X^{τ} is a finitary factor of $((X_{v,i})_{0 \leq i < M})_{v \in \mathbb{Z}^d}$ with stretched-exponential tails. Since $((X_{v,i})_{0 \leq i < M})_{v \in \mathbb{Z}^d}$ is a finite-valued i.i.d. process, this yields the required coding for ω^*. □

Our method of proof of Proposition 3.1 gives a slightly stronger result. We call σ a *simple stopping-process* if it is a B^*-stopping-process, where B^* is defined by $B_n^* := \{\mathbf{0}\} \times \{0, 1, \ldots, n\}$. In this case, X^{σ} can unambiguously be thought of as $(X_{v,i})_{v \in \mathbb{Z}^d, 0 \leq i \leq \sigma_v}$.

Proposition 3.2. *Let $X = (X_{v,i})_{v\in\mathbb{Z}^d, i\geq 0}$ be a finite-valued i.i.d. process, let B be linear, let τ be a stationary B-stopping-process for X and σ a stationary simple stopping-process for X. Suppose τ_v has exponential tails and $\mathbb{E}|B_{\tau_v}| < \mathbb{E}\sigma_v + 1$. Then X^τ is a finitary factor of X^σ with stretched-exponential tails.*

Note that, since σ is simple, the condition $\mathbb{E}|B_{\tau_v}| < \mathbb{E}\sigma_v + 1$ may be more naturally written as $\mathbb{E}|B_{\tau_v}| < \mathbb{E}|B_{\sigma_v}^*|$. Proposition 3.1 is the special case of Proposition 3.2 in which σ is taken to be the deterministic simple stopping-process given by $\sigma_v = M - 1$ for all v. The rest of the paper is devoted to the proof of Proposition 3.2.

Remark 3.3. One may make slight modifications to the proof of the proposition to obtain various improvements. For instance, the same conclusion holds under the weaker assumptions that τ_v has only stretched-exponential tails and that Δ_n grows polynomially fast in n. In fact, one could even allow somewhat heavier tails and faster growing Δ_n at the expense of obtaining a coding radius with heavier tails. This is true even to the extent that, with no assumptions on the tails of τ_v or on the growth of Δ_n, the conclusion still holds albeit with no information on the coding radius. On the other hand, under the stronger assumption that τ is also a simple stopping-process, the coding radius can be shown to have exponential tails.

Remark 3.4. Proposition 3.2 holds also for *random* simple stopping-processes σ (though we do not allow randomness in the B-stopping-process τ), provided the randomness is made independent for each vertex in the following sense: There exists an i.i.d. process $X' = (X'_v)_{v\in\mathbb{Z}^d}$, independent of X, such that, for each v, σ_v is an almost surely finite stopping time with respect to the filtration $(\mathcal{F}_v^n \vee \sigma(X'_v))_{n\geq 0}$, where $\mathcal{F}_v^n \vee \sigma(X'_v)$ is the smallest σ-algebra containing $\mathcal{F}_v^n$ and the one generated by X'_v. By working conditionally on X', the proofs go through essentially unchanged.

4 The algorithm

In this section, we provide the algorithm used to construct the finitary coding stated in Proposition 3.2. We then use it in Section 5 to prove the proposition.

Throughout this section, we work in the setting of Proposition 3.2 so that $X = (X_{v,i})_{v\in\mathbb{Z}^d, i\geq 0}$ is an i.i.d. process taking values in a finite set S, B is linear, τ is a stationary B-stopping-process for X and σ is a stationary simple stopping-process for X. In addition, τ_v has exponential tails and $\mathbb{E}|B_{\tau_v}| < \mathbb{E}\sigma_v + 1$. We may also assume without loss of generality that σ_v is bounded.

We construct an algorithm which, given a realization $\mathcal{Y}$ of the "source" process X^σ, deterministicly computes an output $\mathcal{Z}$ having the distribution of the "target" process X^τ. In the special case where $\sigma_v = M - 1$ deterministically for all v (as in Proposition 3.1), we could imagine that there is a single space-time landscape, initially containing variables in the subset $\mathbb{Z}^d \times \{0, 1, \ldots, M-1\}$ of space-time, and that these variables may be "transported" from their original locations to new locations as needed to construct $\mathcal{Z}$. However, in general, as σ is a stopping-process, we do not know which subset of space-time initially contains variables unless we expose some of the variables, but this would bias them and so we could not easily use them to construct $\mathcal{Z}$. Thus, instead of revealing the entire random field $\mathcal{Y}$ at once, the algorithm slowly reveals more and more of $\mathcal{Y}$ as is needed to generate more and more of $\mathcal{Z}$. As both $\mathcal{Y}$ and $\mathcal{Z}$ are realizations of stopping-processes, it is convenient to think that the input to the algorithm is in fact a realization of X, which the algorithm uses to simultaneously construct both $\mathcal{Y}$ and $\mathcal{Z}$ in such a manner that the variables of X used to construct $\mathcal{Z}$ are a subset of those used to construct $\mathcal{Y}$. We may thus imagine that there are in fact three space-time landscapes: one corresponding to the original process X, one to the source process $\mathcal{Y}$, and one to the target process $\mathcal{Z}$. When the algorithm wishes to reveal an additional piece of $\mathcal{Y}$, the

required variable is easily generated – it is simply read from the same location in the X process. On the other hand, when an additional piece of $\mathcal{Z}$ needs to be generated, it must be matched to a variable used by $\mathcal{Y}$. Here comes into play the crucial assumption that $\mathbb{E}|B_{\tau_v}| < \mathbb{E}\sigma_v + 1$, which ensures that $\mathcal{Z}$ uses less variables than $\mathcal{Y}$ on average. Thus, from the point of view of $\mathcal{Z}$, as any variable used by $\mathcal{Y}$ is "available" to be used by $\mathcal{Z}$, there are many available variables (much more than needed) for $\mathcal{Z}$, and one needs only to find a suitable way of "transporting" these from the source to the target.

We call the variables of the source process $\mathcal{Y}$ *inputs* and the variables of the target process $\mathcal{Z}$ *outputs*. We stress that transporting a variable from (u,i) to (w,j) simply means that the source location (u,i) and target location (w,j) are matched to one another so that the input $\mathcal{Y}_{u,i}$ and the output $\mathcal{Z}_{w,j}$ are identified. Moreover, when we say that an input is generated, say at location (u,i), we simply mean that the corresponding variable $X_{u,i}$ is revealed and identified with $\mathcal{Y}_{u,i}$, and when we say that an output is generated, say at location (u,i), we mean that a suitable input is transported to (u,i).

The algorithm consists of a "simulator" for each vertex $v \in \mathbb{Z}^d$, which has an associated *source location* (thought of as a space-time location of $\mathcal{Y}$) and *target location* (thought of as a space-time location of $\mathcal{Z}$). At each time step n, the simulators simultaneously execute a common procedure (this will guarantee that any output of the algorithm is translation-equivariant). The goal of the v-simulator is to ensure that its stopping time τ_v is reached (with respect to the target process $\mathcal{Z}$) and that all relevant outputs for $\mathcal{Z}_v$ (i.e., those corresponding to space-time locations in $v + B_{\tau_v}$) have been generated (that is, to determine an integer $t_v \geq 0$ and a configuration $\xi \in S^{v+B_{t_v}}$ for which $\tau_v(\xi) = t_v$). Once this happens, v will be "satisfied", the final output $\mathcal{Z}_v$ will be known, and the v-simulator will remain idle; until then, the v-simulator will be in a constant state of searching (in that its source location will change at every time step), trying to find an unused input at the source location which it can transport to the target location. We note that there is a complex interplay between the different simulators. On the one hand, they are competing for shared resources, namely, the inputs. On the other hand, as different sites v may rely on common outputs in order to compute their final output $\mathcal{Z}_v$, the simulators may occasionally "unintentionally help" each other reach their goals (as long as it helps them too) by generating an output which is also required by another simulator (though we do not exploit this in the proof). This is in fact the origin of some complications, which presumably cannot be avoided. Our algorithm is inspired partly by the algorithms in [5, 12] (see Section 4.4 for a comparison between our algorithm and the one in [5]).

4.1 Informal description of the algorithm

The goal of the v-simulator is to make sure that the final output $\mathcal{Z}_v$ becomes known after some finite number of steps. To do this, the v-simulator proceeds as follows: Initially, at time step $n = 0$, it reveals the variable $X_{v,0}$, which corresponds to the single space-time location in $v + B_0 = \{(v,0)\}$ (see Section 4.3 for a formal definition of sets of the form $v + A$). It then consults the stopping rule τ_v to see whether or not it should continue. If it has reached the stopping time, i.e., $\tau_v = 0$, then the final output is known, namely, $\mathcal{Z}_v$ is the element in S^{B_0} given by $(\mathcal{Z}_v)_{(v,0)} = X_{v,0}$, so that the simulator is satisfied and can stop. If it has not reached the stopping time, i.e., $\tau_v > 0$, its next goal becomes to generate the outputs in $v + (B_1 \setminus B_0)$. Let us come back to how this is done in a moment. Once these have been generated (which may require many steps of the algorithm), the v-simulator consults the stopping rule τ_v again, this time to check whether $\tau_v = 1$. If indeed $\tau_v = 1$, then it is satisfied and the final output is known, namely, $\mathcal{Z}_v$ is an element in S^{B_1} given by the generated outputs at space-time locations $v + B_1$. If instead $\tau_v > 1$, the v-simulator continues in a similar manner, with the general rule

being that once the v-simulator learns that $\tau_v > k$, it continues to generate the outputs in $v + (B_{k+1} \setminus B_k)$, and then to check the stopping rule in order to determine whether or not it should continue. Eventually the stopping time is reached, the final output is known and the v-simulator is satisfied.

Let us now explain how the v-simulator generates the outputs in $v + (B_k \setminus B_{k-1})$. Firstly, it does so one output at a time (in an arbitrary order), and so we merely focus on how it generates a single output at space-time location (w, j). Of course, one way to do this is simply to use the original variable residing at that location, namely, $X_{w,j}$. However, since we want to obtain a coding from X^σ, we must be sure to only use inputs (those variables residing in the scope of the source process), i.e., we cannot use $X_{w,j}$ unless $\sigma_w \geq j$. We also cannot use an input if it has already been used (transported away) by some other simulator at a previous time. Thus, we may need to search for an input at a different location (u, i) and transport it from there to (w, j). Roughly speaking, the simulator moves along the space-time landscape of the source process, checking to see whether there is an unused input which it can transport to the target location (w, j). At every time step, it checks a single source location (u, i). If the input at that location is not available for use, the simulator simply advances its current source location, and does nothing further in that step of the algorithm. This procedure is repeated until the simulator eventually finds an unused input that it can transport. At that time, assuming the required output has not meanwhile been generated by another simulator, it transports it. Either way, the output at (w, j) is sure to have been generated by the end of that step.

Of course, as we are trying to construct a coding, the above procedure must be carried out simultaneously by all the simulators. This leads to some interaction between the different simulators. Let us now give some more specific details about this and the above procedure. We first explain how the v-simulator behaves with regards to the source process in each step:

- If the simulator is satisfied, it does nothing. If it is unsatisfied, it will necessarily move its source location and it does so as follows. It first tries to move up one step in the pile of the vertex u it is currently at. If it cannot, i.e., if it is already at the top of an exhausted pile (in the sense that the stopping time σ_u has been reached), then it moves to the bottom of the pile located one step to the right of u (i.e., to $u + e_1$). Here we informally refer to the inputs at locations (u, i) as the *pile* at u, and think of the pile there as initially empty and then growing as inputs there are generated until it becomes exhausted (i.e., until it reaches its full size given by the stopping time σ_u).

- The above choice implies that if the v-simulator is at the top of a pile which has not yet been exhausted (we shall later call such a pile *loaded*), then the input just above the top of that pile has not yet been used/revealed by any simulator. Thus, it is an unbiased input (having the same distribution as $X_{\mathbf{0},0}$) and is available to be transported. In this situation, regardless of whether or not it is indeed transported, the source location is moved one step up the pile.

- We initially set the v-simulator's source location to be $(v, -1)$ so that it is necessarily at the top of a loaded pile when the algorithm starts.

- Let us point out that when the pile sizes are deterministically fixed (as in the situation of Proposition 3.1), the evolution of the source location is also deterministic (up to knowing at what time the simulator becomes satisfied and stops). However, in general, as σ is a simple stopping-process, the evolution is random: to decide whether or not a pile is exhausted, we must inspect the variables in the pile.

- We could have chosen different conventions here. Our choice has the advantage that there cannot be more than one unsatisfied simulator at any location at any

given time. This means that we do not need to worry about different simulators trying to transport the same input.

Next, we explain how the v-simulator behaves with regards to the target process:

- If the simulator is satisfied, it does nothing. If it is unsatisfied, it might move its target location and it might not. Specifically, it moves precisely when its source is at the top of a loaded pile. Indeed, when this happens, we are assured that the required output can be generated. Moreover, when it moves, it moves to the next element in $v + B_\infty$, where the elements of $B_\infty = \bigcup_{n \geq 0} B_n$ are ordered in any way which respects the inclusions $B_0 \subset B_1 \subset \cdots$.

- Note that the times at which the target location changes is completely determined by the source. In particular, even if the output at the target location has been previously generated by some other simulator, this does not mean that the v-simulator will necessarily advance its target location. In other words, the output at the target location may have already been generated, and it may take the simulator many more steps until it finds an unused input (i.e., its source is at the top of a loaded pile), only to realize at that point in time that it does not need it after all (in which case that input will be wasted – it will not be transported later). This is not the most efficient choice, but it is the one we make.

- We point out that, unlike for the source, there may be many different simulators at a given target location at the same time. This situation just means that the different simulators all wish to generate the same output. Among these simulators, many may also be at the top of a loaded pile (in the source), which means that they can transport an input. Thus, we must take care that different simulators do not generate the same output. We must therefore prioritize the simulators in some manner. To this end, we simply make the choice that the lexicographical-minimal simulator (among those at the top of a loaded pile) takes priority, namely, it is the one to generate the output, while the others do not transport an input (note that this is again not the most efficient way to do things, since we are throwing away inputs which could have been used later, but this is not too wasteful and we simply made a choice which we found convenient).

We emphasize that the algorithm may transport an input away from a certain location at some point in time, and then transport some other input into that same location at a later point in time. That is, even if eventually there is an input at location (u, i) (in the sense that $\sigma_u \geq i$) and the output at that same location is eventually needed by some simulator (in the sense that $(u, i) \in v + B_{\tau_v}$ for some v), there is no guarantee that the variable that will eventually end up to be the output at (u, i) is the one that was originally the input there. The important property is that any given input can only be transported away once, and any given output can only be generated (i.e., transported into) once. This is another reason it is helpful to imagine separate space-time landscapes for the source and target processes.

We refer the reader to Figure 1 for an illustration of the algorithm.

4.2 Further explanation of the figure

Figure 1 illustrates the first several steps of the algorithm. The figure contains a detailed caption, and here we provide some additional information.

Let us first address the setting considered in the figure. Of course we consider $d = 1$ as it would be difficult to provide a useful picture for two of more dimensions. On the other hand, the specific B_n considered there is not essential, and the reason for that choice was to allow the simulators to "climb up" in a short number of steps. We note
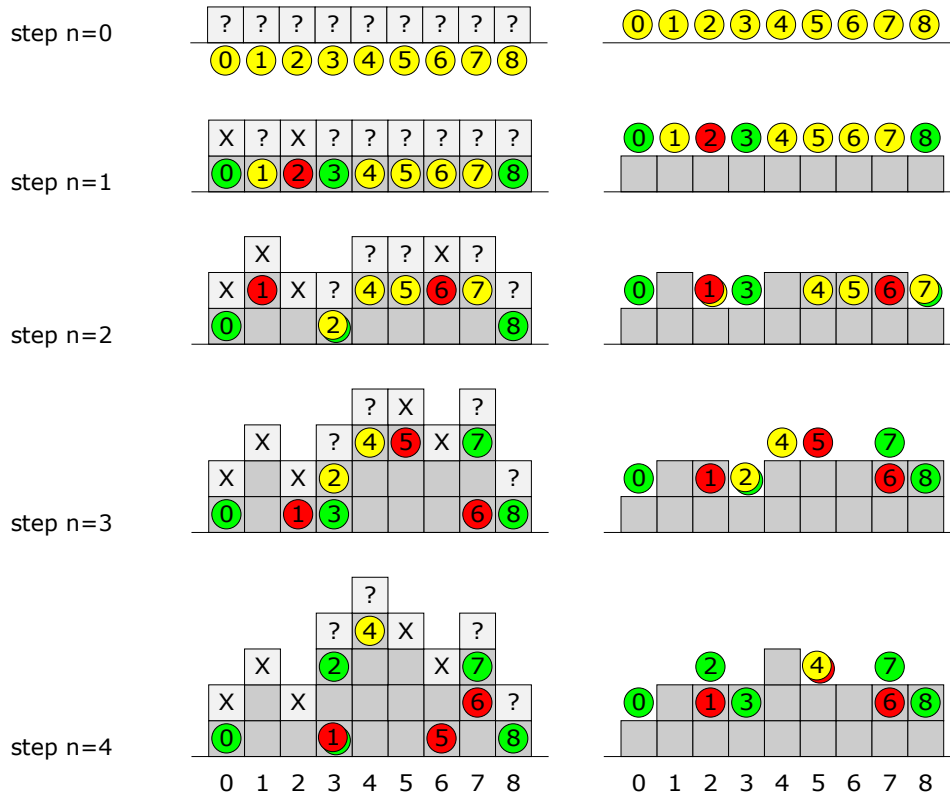
Figure 1: An illustration of the first five steps of the algorithm. For illustration purposes, we consider here the case where $d = 1$ and $B_n = \{(u, i) : 0 \le u \le i \le n\}$, ordering the elements of B_∞ as $(0, 0), (0, 1), (1, 1), (0, 2), (1, 2), (2, 2)$ and so on. The figure depicts the processes Y^n and Z^n and the state of the simulators at the end of step n for $n = 0, \dots, 4$. The row just above the horizontal axis corresponds to the portion $\mathbb{Z} \times \{0\}$ of space-time. *Left:* The source process Y^n and the source locations (U^n_v, I^n_v) of the simulators. A gray background at space-time location (u, i) indicates that the input $Y^n_{u,i}$ has been generated. An $\times$ indicates an unloaded vertex, while a question mark indicates a loaded vertex. *Right:* The target process Z^n and the target locations (W^n_v, J^n_v) of the simulators. A gray background at space-time location (w, j) indicates that the output $Z^n_{w,j}$ has been generated. *Simulators:* The simulators are depicted in green, yellow or red according to whether they are satisfied, unsatisfied but at the top of a loaded pile, or otherwise. A green simulator does not move as it has finished running (case (i) in the algorithm). A yellow simulator advances its source location by moving up one step in its current pile, reads the unused input at that new location, transports this input to its current target location (if needed), and then advances its target location by moving to the "next place in line" according to the ordering on B_∞ (case (iv) in the algorithm). We note that when two yellow simulators occupy the same target location, only one of them actually generates the output (i.e., transports an input to that location). A red simulator does not have access to an unused input, and so it advances its source location by either moving up the current pile if it is not yet at the top (case (ii) in the algorithm) or otherwise by moving to the bottom of the next pile (case (iii) in the algorithm), while its target location remains unchanged. In particular, a red simulator does not advance its target location even if the corresponding output is (or was previously) generated by a different simulator. See Section 4.2 for further details about the figure.

that this choice for B_n may be regarded as a simplification of what would be used for the one-dimensional case of Theorem 2.1 (since the B_n are only "one-sided cones", whereas the theorem would require symmetric "two-sided cones").

Let us now consider the evolution of the simulators throughout the steps depicted in the figure. Initially, the source and target locations of each v-simulator are set to $(v, -1)$ and $(v, 0)$, respectively. This means that v-simulator is currently trying to generate the output at location $(v, 0)$ and it is currently looking for an unused input (which it would like to transport) just above the source location $(v, -1)$, namely, at $(v, 0)$. Indeed, initially there is always an unused input there (since $\sigma_v \geq 0$ by assumption). This situation is depicted at the top of the figure. Thus, at step $n = 1$ of the algorithm, every v-simulator moves its source location one step up the pile to $(v, 0)$, (vacuously) transports the input from $(v, 0)$ to $(v, 0)$, and advances its target location to $(v, 1)$ (note that $(0, 1)$ is the successor of $(0, 0)$ in the chosen ordering of B_∞). At this stage, some simulators have already become green (satisfied) and thus have $\tau_v = 0$ – these are simulators 0, 3 and 8. Let us follow what happens next to simulator 1 (which is still unsatisfied): since the 1-simulator is yellow (it is at the top of a loaded pile), it moves its source location one step up the pile to $(1, 1)$, (vacuously) transports the input from $(1, 1)$ to $(1, 1)$, and advances its target location to $(2, 1)$ (because $(1, 1)$ follows $(0, 1)$ in the order on B_∞). Since at the end of step 2, the 1-simulator is red (it is no longer in a loaded pile), in step 3 it does not change its target location and simply moves its source location to the bottom of the next pile, which is $(2, 0)$. Since it is still red, in step 4 it again only moves its source location, this time to $(3, 0)$. We stress that even though, at the end of step 3, the output at the 1-simulator's target location $(2, 1)$ has already been generated (it was transported from location $(3, 0)$ by the 2-simulator in step 3), the 1-simulator still does not advance its target location; it will only do so once it becomes yellow. Finally, since the 1-simulator is still red at the end of step 4, in the next step (which is not depicted in the figure) it will move its source location one step up the pile to $(3, 1)$. At this stage, we still do not know the eventual value of τ_1, we only know that $\tau_1 \geq 1$ (since there is an output in $1 + B_1$ which is needed). Similarly, at the end of step 4, we know that $\tau_0 = \tau_3 = \tau_8 = 0$, $\tau_2 = \tau_7 = 1$, $\tau_6 \geq 1$, $\tau_4 \geq 2$ and $\tau_5 \geq 2$. In particular, we know the final output for vertices $\{0, 2, 3, 7, 8\}$, but not yet for $\{1, 4, 5, 6\}$.

4.3 Formal definition of the algorithm

Before providing the algorithm, we require some preparation.

Let us employ the following useful convention regarding stopping times. Suppose that π is an almost surely finite stopping time with respect to the filtration $(\mathcal{F}_0^n)_{n \geq 0}$ defined in (3.1). We may regard π as a deterministic function from $\bigcup_{n \geq 0} S^{B_n}$ to $\mathbb{N} \cup \{*\}$ having the property that, for any $n \geq 0$, $\xi \in S^{B_n}$ and $\xi' \in S^{B_{n+1}}$ such that $\xi'|_{B_n} = \xi$, we have $\pi(\xi) \in \{0, \ldots, n, *\}$, we have $\pi(\xi') = \pi(\xi)$ when $\pi(\xi) \neq *$, and we have $\pi(\xi') \in \{n + 1, *\}$ when $\pi(\xi) = *$. The interpretation here is that a value of $*$ means that the stopping time has not been reached. Note, in particular, that for $m \geq 0$ and $\eta \in S^{B_{n+m}}$, the expression $\pi(\eta) > n$ depends only on $\eta|_{B_n}$ (where it is understood that $* > n$ for all integer n). With this in mind, we note that (with a slight abuse of notation), if $A \subset \mathbb{Z}^d \times \mathbb{N}$ contains B_n, then the expression $\pi(\eta) \leq n$ is well-defined for any $\eta \in S^A$ and depends only on $\eta|_{B_n}$, and thus, the expression $b \in B_{\pi(\eta)}$ is also well-defined for any $b \in B_{n+1}$ (and depends only on $\eta|_{B_n}$). We further abuse notation by identifying an element $\eta \in (S \cup \{\emptyset\})^{\mathbb{Z}^d}$ with the element $\eta' \in S^A$ in the obvious way, by taking $A := \{a \in \mathbb{Z}^d \times \mathbb{N} : \eta(a) \neq \emptyset\}$ and $\eta' := \eta|_A$.

We order the elements of $B_\infty := \bigcup_{n \geq 0} B_n$ in such a manner that, for any n, every element of B_n appears before every element of $B_\infty \setminus B_n$. This induces a notion of

successor for elements in B_∞. For $A \subset \mathbb{Z}^d \times \mathbb{N}$, we write $v + A$ for the translated set $\{(v + u, i) : (u, i) \in A\}$. Translating the ordering from B_∞ to $v + B_\infty$, we obtain a notion of v-*successor* for elements in $v + B_\infty$. More precisely, the v-successor of an element $(v + u, i) \in v + B_\infty$ is $(v + u', i')$, where (u', i') is the successor of (u, i).

At each step $n \geq 0$, we define variables:

- $(U_v^n, I_v^n) \in \mathbb{Z}^d \times \mathbb{N}$, the source location of the v-simulator.
- $(W_v^n, J_v^n) \in \mathbb{Z}^d \times \mathbb{N}$, the target location of the v-simulator.
- $T_v^n \in \{0, 1\}$, the indicator of whether the v-simulator transported input (generated output).

Once the above variables are defined at step n, we further define several objects, all of which are deterministic functions of the above variables. For some of these definitions to make sense, it is important to note that the following properties are satisfied at every step n:

$$\sum_v \sum_{t=1}^n \mathbb{1}_{T_v^t = 1, (U_v^t, I_v^t) = (u, i)} \leq 1 \qquad \text{for all } (u, i) \in \mathbb{Z}^d \times \mathbb{N}, \qquad (4.1)$$

$$\sum_v \sum_{t=1}^n \mathbb{1}_{T_v^t = 1, (W_v^{t-1}, J_v^{t-1}) = (w, j)} \leq 1 \qquad \text{for all } (w, j) \in \mathbb{Z}^d \times \mathbb{N}. \qquad (4.2)$$

Equation (4.1) says that each input is transported away at most once by at most one simulator. Similarly, (4.2) says that every output is generated (transported into) at most once by at most one simulator. As the target location of the v-simulator will be updated immediately after the required output is generated, there is a shift in the time index in (4.2). Thus, $T_v^n = 1$ means that at time step n the v-simulator transported an input from the source location (U_v^n, I_v^n) to the target location (W_v^{n-1}, J_v^{n-1}), thus generating the output at (W_v^{n-1}, J_v^{n-1}).

Consider the set of source-target locations of simulators at transport times:

$$D^n := \left\{ (v, u, i, w, j) : \begin{array}{l} v \in \mathbb{Z}^d, \ T_v^t = 1, \ (U_v^t, I_v^t) = (u, i), \\ (W_v^{t-1}, J_v^{t-1}) = (w, j) \text{ for some } 1 \leq t \leq n \end{array} \right\}.$$

We use D^n and X to construct two $(S \cup \{\emptyset\})$-valued processes $Y^n = (Y_{u,i}^n)_{u \in \mathbb{Z}^d, i \geq 0}$ and $Z^n = (Z_{w,j}^n)_{w \in \mathbb{Z}^d, j \geq 0}$, which represent the partial information on $\mathcal{Y}$ and $\mathcal{Z}$ (the realizations of X^σ and X^τ) that has been revealed by time n. The algorithm may use one of two "update methods": for $(v, u, i, w, j) \in D^n$, we define

$$\textbf{(A)} \ Y_{u,i}^n = Z_{w,j}^n := X_{u,i}, \qquad \textbf{(B)} \ Y_{u,i}^n = Z_{w,j}^n := X_{w,j}.$$

If (u, i) is not in the projection of D^n on the 2nd and 3rd coordinates, then set $Y_{u,i}^n := \emptyset$, and similarly, if (w, j) is not in the projection of D^n on the 4th and 5th coordinates, then set $Z_{w,j}^n := \emptyset$. Note that (4.1) and (4.2) ensure that both update methods are well-defined. We stress that the two update methods are never used in conjunction with one another – either update method **(A)** is used throughout all steps of the algorithm or update method **(B)** is.

The Y process is associated with σ, and the Z process with τ. Thus, in update method **(A)**, σ "sees" the original process X, while τ "sees" a transformed process in which inputs have been transported between space-time locations; in update method **(B)**, the situation is reversed – τ sees the origin process and σ sees a transformed process. Another point of view is that D^n (after forgetting the first coordinate) defines a bipartite graph between two copies of $\mathbb{Z}^d \times \mathbb{N}$ in which any vertex of one copy is matched to at most one vertex in the other copy. The two update methods can then be thought of as

orienting all edges from the first copy to the second, or vice versa, where the orientation of an edge determines the direction of flow of information, with the original process X always associated with the copy from which the edges are oriented outwards (so that variables are transported along the edges in the direction of orientation). As we are interested in realizing the τ process via the σ process, the natural choice is to transport variables from the latter to the former as in update method **(A)**. Nevertheless, it will turn out to be a helpful idea to consider also the reversed direction of flow. Thus, update method **(A)** will yield the required coding, whereas update method **(B)** will only be used as a comparison tool in the analysis (namely in the proof of Lemma 5.3). As such, we mainly have update method **(A)** in mind in our definitions.

We further define

- $L_u^n := \max\{i : (v, u, i, w, j) \in D^n \text{ for some } v, w, j\}$, the last input revealed at u.

- u is *loaded* at time n if $\sigma_u(Y^n) > L_u^n$.

- v is *satisfied* at time n if $(W_v^n, J_v^n) \notin v + B_{\tau_v(Z^n)}$.

Thus, L_u^n is the size of the pile at u (in the source process) at time n. A vertex u is loaded at time n if there are more inputs available at u than have already been used by time n, i.e., if the pile at u has not been exhausted by time n. A vertex v is satisfied at time n if the output at the target location of the v-simulator is not needed in order to compute the final output $\mathcal{Z}_v$. In particular, due to way that the target location evolves, this implies (but is not precisely equivalent to) that the outputs that v needs for its final output have already been generated by time n (see (4.4) below), so that the final output is known at this time.

The fact that the notions of loaded and satisfied are well-defined is not obvious from their definitions. The fact that the notion of loaded is well-defined follows from the above discussion about stopping times and the following property which will hold at each step n:

$$Y_{u,i}^n \neq \emptyset \quad \text{if and only if} \quad i \leq L_u^n, \qquad \text{for all } (u, i) \in \mathbb{Z}^d \times \mathbb{N}. \tag{4.3}$$

Similarly, the fact that the notion of satisfied is well-defined follows from the following property, which will hold for all n:

$$Z_{w,j}^n \neq \emptyset \qquad \begin{array}{l} \text{for all } (w, j) \in \mathbb{Z}^d \times \mathbb{N} \text{ that strictly} \\ v\text{-precede } (W_v^n, J_v^n) \text{ for some } v \in \mathbb{Z}^d. \end{array} \tag{4.4}$$

Finally, for $(w, j) \in \mathbb{Z}^d \times \mathbb{N}$, we also define

$$Q^n(w, j) := \left\{ v : (W_v^n, J_v^n) = (w, j), \begin{array}{l} U_v^n \text{ is loaded at time } n, \ I_v^n = L_{U_v^n}^n \\ v \text{ is unsatisfied at time } n, \ Z_{w,j}^n = \emptyset \end{array} \right\}.$$

Thus, $Q^n(w, j)$ consists of those simulators who both wish to generate the output at (w, j) and can also do so (they wish to do so as they are unsatisfied, meaning that they need that output, and as the output has not yet been generated; they can do so as they are at the top of a loaded pile in the source process). Since only one such simulator can be allowed to actually generate the output at (w, j), we will let the lexicographical-minimal one do so.

With these definitions, we can now present the algorithm. We refer the reader to Section 4.1 for an informal description and to Figure 1 for an illustration.

Algorithm Finitary coding from X^σ to X^τ

---

    for $v \in \mathbb{Z}^d$ (simultaneously) **do**

        $(U_v^0, I_v^0, W_v^0, J_v^0, T_v^0) \leftarrow (v, -1, v, 0, 0)$

    end for

    for $n = 1, 2, \ldots$ **do**

        for $v \in \mathbb{Z}^d$ (simultaneously) **do**

            if v is satisfied at time $n-1$ **then**

  i. $(U_v^n, I_v^n, W_v^n, J_v^n, T_v^n) \leftarrow (U_v^{n-1}, I_v^{n-1}, W_v^{n-1}, J_v^{n-1}, 0)$

            else if $I_v^{n-1} < L_{U_v^{n-1}}^{n-1}$ **then**

  ii. $(U_v^n, I_v^n, W_v^n, J_v^n, T_v^n) \leftarrow (U_v^{n-1}, I_v^{n-1} + 1, W_v^{n-1}, J_v^{n-1}, 0)$

            else if U_v^{n-1} is unloaded at time $n-1$ **then**

  iii. $(U_v^n, I_v^n, W_v^n, J_v^n, T_v^n) \leftarrow (U_v^{n-1} + e_1, 0, W_v^{n-1}, J_v^{n-1}, 0)$

            else

  iv. $(U_v^n, I_v^n) \leftarrow (U_v^{n-1}, I_v^{n-1} + 1)$

                $(W_v^n, J_v^n) \leftarrow v\text{-successor of } (W_v^{n-1}, J_v^{n-1})$

                $T_v^n \leftarrow \mathbb{1}(v$ is the lexicographical-minimal element of $Q^{n-1}(W_v^{n-1}, J_v^{n-1}))$

            end if

        end for

    end for

4.4 Comparison between our algorithm and that of van den Berg and Steif [5]

The two algorithms are similar in spirit (though they are not set up in the same way) and we focus here on the moral differences between the two. We have identified two such differences, the primary one being in how they relate to unneeded variables and, consequently, in how they transport such variables between space-time locations. Here, "needed" may refer to either an input or an output, where an input (output) at location (u, i) is needed by time n if $Y_{u,i}^n \neq \emptyset$ ($Z_{u,i}^n \neq \emptyset$). Roughly speaking, the algorithm in [5] declares an input variable unneeded at a certain time once it is guaranteed that the output variable at the same location will not be needed at any later time (and was also not needed until that time). Only inputs which are marked as unneeded in this sense are allowed to be transported. On the other hand, our algorithm never declares an input variable unneeded. Instead, we only concern ourselves with whether an input was not needed by a certain time, and any such variable is allowed to be transported at that time. If at a later time it turns out that the output at the same location was needed after all, another input variable will be transported to that location. In other words, the algorithm in [5] transports an input from location (u, i) to another location (w, j) only if the output at (u, i) is never needed, whereas our algorithm does not have this restriction, and may transport from (u, i) to (w, j) at some time, and then from (u', i') to (u, i) at a later time. The latter approach is essential in the generality of Proposition 3.1 and Proposition 3.2. The reason is that, while for some choices of $B = (B_n)_n$, any particular output variable could only be potentially needed by finitely many vertices (e.g., as for the "cones" used in the proof of Theorem 2.1, where the output at (w, j) can only be needed by vertices at distance at most Δj from w), in general, any vertex might need that variable at some time (e.g., as for the "cubes" given by $B_n = \{(u, i) : |u| \leq \Delta n, \, 0 \leq i \leq n\}$) so that it is not possible to know (in a finitary manner) whether or not an output variable will be needed eventually. The second difference between the algorithms is that, unlike the algorithm in [5], ours is somewhat wasteful (by design; see Section 4.1) in that in certain situations

it decides not to use an available input variable (and to simply throw it away). We found this useful (though it is probably not essential) for keeping track of how far variables are transported, which was important for understanding the coding radius.

5   Proof of Proposition 3.2

In this section, we use the algorithm described in Section 4 to prove Proposition 3.2.

The following claim establishes some simple properties of the algorithm. Let $\preceq$ denote the partial order on $\mathbb{Z}^d$ in which $u \preceq u'$ if $u' = u + ke_1$ for some $k \geq 0$. We also denote by $\preceq$ the partial order on $\mathbb{Z}^d \times \mathbb{Z}$ in which $(u,i) \preceq (u',i')$ if $u \preceq u'$ and $(i' - i)\mathbb{1}_{u=u'} \geq 0$.

Claim 5.1. *In either update method, almost surely, for all $n \geq 1$, $i,j \geq 0$ and $u,v,v',w \in \mathbb{Z}^d$,*

1. *(4.1), (4.2), (4.3) and (4.4) hold.*
2. *If $v \prec v'$ and both are unsatisfied at time $n - 1$, then $(U_v^{n-1}, I_v^{n-1}) \prec (U_v^n, I_v^n) \prec (U_{v'}^n, I_{v'}^n)$.*
3. *If $v \preceq w \prec U_v^n$, then w is unloaded at time $n - 1$.*

Proof. The claim follows easily by induction on n. $\qquad\square$

The following lemma states precisely the intuitive fact that transporting inputs from one space-time location to another does not change the resulting distribution. Denote the state at time n by $\$^n := (U^n, I^n, W^n, J^n, T^n, D^n)$, where $U^n = (U_v^n)_{v \in \mathbb{Z}^d}$, $I^n = (I_v^n)_{v \in \mathbb{Z}^d}$ and so forth.

Lemma 5.2. *The distribution of $(\$^n, Y^n, Z^n)_{n \geq 0}$ does not depend on whether update method **(A)** or **(B)** is used in the algorithm.*

Proof. Observe that the algorithm does not explicitly depend on the update method used, but rather depends on it implicitly through the definitions of Y^n and Z^n. We prove by induction that the distribution of $\mathcal{S}^n := (\$^m, Y^m, Z^m)_{0 \leq m \leq n}$ does not depend on the update method. This is immediate for $n = 0$, since $\mathcal{S}^0$ is deterministic. Fix $n \geq 1$ and observe that $\$^n$ is measurable with respect to $\mathcal{S}^{n-1}$. It thus suffices to show that (i) when using update method **(A)**, conditioned on $\mathcal{S}^{n-1}$, $(X_{u,i})_{(v,u,i,w,j) \in D^n \setminus D^{n-1}}$ is a sequence of independent random variables having the distribution of $X_{\mathbf{0},0}$, and (ii) when using update method **(B)**, conditioned on $\mathcal{S}^{n-1}$, $(X_{w,j})_{(v,u,i,w,j) \in D^n \setminus D^{n-1}}$ is such a sequence. Indeed, (i) follows easily from (4.1) and (ii) from (4.2). $\qquad\square$

Lemma 5.3. *Almost surely, every $v \in \mathbb{Z}^d$ is eventually satisfied. Moreover,*

$$\mathbb{P}(v \text{ is not satisfied at time } n) = e^{-\Omega\left(n^{1/(d+2)}\right)} \qquad \text{as } n \to \infty.$$

As will be explained in the proof of Proposition 3.2 below, Lemma 5.3 implies that the algorithm "locally terminates" in finite time in the sense that the final output at any vertex is determined at some finite step. Nevertheless, this does not yet imply that the algorithm yields a finitary coding. What is missing is some control on the propagation of information in each step. This is the content of the following lemma. Let Δ be as in (3.2). Denote $D_v^n := \{(u,i,w,j) : (v,u,i,w,j) \in D^n\}$.

Lemma 5.4. *When using update method **(A)**, for any $n \geq 0$ and $v \in \mathbb{Z}^d$, the following random variables are measurable with respect to $(X_{u,i})_{|u-v| \leq 5\Delta n^2, 0 \leq i \leq \sigma_u}$:*

1. *$\$_v^n = (U_v^n, I_v^n, W_v^n, J_v^n, T_v^n, D_v^n)$,*
2. *$\{Z_{w,j}^n\}_{|w-v| \leq \Delta n, j \geq 0}$.*

Before proving Lemma 5.3 and Lemma 5.4, we first use them to prove the proposition.

5.1 Proof of Proposition 3.2

Consider the algorithm using update method **(A)**. For $n \geq 1$, define

$$M_v^n := \min\{m \geq 0 : (W_v^n, J_v^n) \in v + B_m\}.$$

Note that $M_v^n \geq 1$ so that (4.4) implies that $Z_{w,j}^n \neq \emptyset$ for all $(w,j) \in v + B_{M_v^n - 1}$. Thus,

$$\mathscr{Z}_v^n := (Z_{v+w,j}^n)_{(w,j) \in B_{M_v^n - 1}}$$

takes values in $\bigcup_{n \geq 0} S^{B_n}$. Let N_v denote the time at which v is first satisfied. By Lemma 5.3, N_v is almost surely finite. Recall our conventions about stopping times discussed in the beginning of Section 4. By the definition of satisfied, $(W_v^{N_v}, J_v^{N_v}) \notin v + B_{\tau_v(Z^{N_v})}$, so that $\tau_v(Z^{N_v}) < M_v^{N_v}$. Similarly, since v is not satisfied at time $N_v - 1$, it follows that $\tau_v(Z^{N_v-1}) \geq M_v^{N_v-1}$, and hence, also that $\tau_v(Z^{N_v}) \geq M_v^{N_v-1}$. Since $M_v^n \leq M_v^{n-1} + 1$, we conclude that $\tau_v(Z^{N_v}) = M_v^{N_v} - 1$. Thus, $\mathscr{Z}_v^{N_v} = (Z_{v+w,j}^{N_v})_{(w,j) \in B_{\tau_v}}$, where $\tau_v := \tau_v(Z^{N_v}) = \tau_0(\mathscr{Z}_v^{N_v})$. Lemma 5.2 now implies that $(\mathscr{Z}_v^{N_v})_{v \in \mathbb{Z}^d}$ equals X^τ in distribution. Since all the operations in the algorithm are translation-equivariant, we have thus obtained a coding from X^σ to X^τ.

Let us check that this coding is finitary and that its coding radius R has stretched-exponential tails. Indeed, since Lemma 5.4 implies that $\{N_0 \leq n\}$ and $\mathscr{Z}_0^n$ are measurable with respect to $(X_{u,i})_{|u| \leq 5\Delta n^2, 0 \leq i \leq \sigma_u}$, it follows that $R \leq 5\Delta N_0^2$. Lemma 5.3 then yields that

$$\mathbb{P}(R > 5\Delta n^2) \leq \mathbb{P}(N_0 > n) = \mathbb{P}(0 \text{ is not satisfied at time } n) = e^{-\Omega\left(n^{1/(d+2)}\right)}. \qquad \square$$

5.2 Proof of Lemma 5.3

For the proof of Lemma 5.3, we require a large-deviation-type result, which we now describe. Let $X = (X_i)_{i \in \mathbb{Z}}$ be a sequence of non-negative random variables. We say that X is *stopping-like* if there exists $\Delta > 0$ such that for any finite $I, J \subset \mathbb{Z}$ and any non-negative numbers $(r_i)_{i \in I \cup J}$, the two events $\{X_i > r_i \text{ for } i \in I\}$ and $\{X_j > r_j \text{ for } j \in J\}$ are independent whenever the two sets $\bigcup_{i \in I}[i - \Delta r_i, i + \Delta r_i]$ and $\bigcup_{j \in J}[j - \Delta r_j, j + \Delta r_j]$ are disjoint. Observe that, if there exists a sequence $(Y_i)_{i \in \mathbb{Z}}$ of independent random variables satisfying that, for any $i \in \mathbb{Z}$ and any $r \geq 0$, the event $\{X_i > r\}$ is measurable with respect to $\{Y_j\}_{|i-j| \leq \Delta r}$, then X is stopping-like. Observe also that, if X is a stopping-like process, then $(X_i \mathbb{1}_{\{X_i \leq r\}})_{i \in \mathbb{Z}}$ is a $2\Delta r$-dependent process for any $r > 0$, where a process $(Y_i)_{i \in \mathbb{Z}}$ is said to be k-dependent if $(Y_i)_{i \in I}$ and $(Y_j)_{j \in J}$ are independent whenever $I, J \subset \mathbb{Z}$ satisfy that $|i - j| > k$ for all $i \in I$ and $j \in J$.

Lemma 5.5. *Let $X = (X_i)_{i \in \mathbb{Z}}$ be a non-negative stopping-like stationary sequence and suppose that X_0 has exponential tails. Let $f \colon [0, \infty) \to [0, \infty)$ be a measurable function satisfying that $f(t) \leq Bt^b$ for some $B, b > 0$ and all $t \geq 1$. Denote $\mu := \mathbb{E}f(X_0)$ and $\beta := \frac{1}{1+b}$. Then, for any $a > \mu$,*

$$\mathbb{P}(f(X_1) + \cdots + f(X_n) \geq an) = e^{-\Omega(n^\beta)} \qquad \text{as } n \to \infty.$$

Proof. The proof involves a truncation argument. Thus, we first write $f(X_i) = Y_i + Y_i'$, where

$$Y_i := f(X_i)\mathbb{1}_{\{X_i \leq n^\alpha\}} \qquad \text{and} \qquad Y_i' := f(X_i)\mathbb{1}_{\{X_i > n^\alpha\}},$$

and α is any positive number less than $\frac{1}{2}\beta(1-\beta)$. Then, for any $\epsilon > 0$,

$$\mathbb{P}(f(X_1) + \cdots + f(X_n) \geq (\mu + 2\epsilon)n) \leq \mathbb{P}(Y_1 + \cdots + Y_n \geq \mu n + \epsilon n) + \mathbb{P}(Y_1' + \cdots + Y_n' \geq \epsilon n).$$

Thus, it suffices to bound separately the two terms on the right, showing that each is $e^{-\Omega(n^\beta)}$.

Finitary coding for the sub-critical Ising model with finite expected coding volume

For the first term, we prove the stronger bound

$$\mathbb{P}\big(Y_1 + \cdots + Y_n \geq \mu n + \epsilon n\big) = e^{-\Omega(n^{1-2\alpha/\beta})}. \tag{5.1}$$

For $0 \leq i \leq n$, denote

$$Z_i := \mathbb{E}[Y_1 + \cdots + Y_n \mid \mathcal{F}_i], \qquad \text{where } \mathcal{F}_i := \sigma\big(\big\{X_j \mathbb{1}_{X_j \leq n^\alpha}, \mathbb{1}_{X_j \leq n^\alpha}\big\}_{1 \leq j \leq i}\big).$$

Note that $(Z_i)_{0 \leq i \leq n}$ is a martingale satisfying $Z_0 = (\mathbb{E}Y_0)n \leq \mu n$ and $Z_n = Y_1 + \cdots + Y_n$. Hence, by the Azuma–Hoeffding inequality (see, e.g., [24]),

$$\mathbb{P}\big(Y_1 + \cdots + Y_n \geq \mu n + t\big) \leq \mathbb{P}\big(Z_n \geq Z_0 + t\big) \leq \exp\left(-\frac{t^2}{2\sum_{i=1}^n c_i^2}\right), \qquad t \geq 0,$$

where $c_i := \|Z_i - Z_{i-1}\|_\infty$ is the essential supremum of the increment $Z_i - Z_{i-1}$. Thus, (5.1) will follow if we show that $c_i \leq Cn^{\alpha/\beta}$. Indeed, since Y is a $Bn^{\alpha b}$-bounded $2\Delta n^\alpha$-dependent process,

$$|Z_i - Z_{i-1}| = \left|\sum_{j=i}^{n}\big(\mathbb{E}[Y_j \mid \mathcal{F}_i] - \mathbb{E}[Y_j \mid \mathcal{F}_{i-1}]\big)\right| \leq \sum_{j=i}^{i+\lfloor 2\Delta n^\alpha\rfloor}\big|\mathbb{E}[Y_j \mid \mathcal{F}_i] - \mathbb{E}[Y_j \mid \mathcal{F}_{i-1}]\big| \leq Cn^{\alpha/\beta}.$$

We now turn to the second term. Note that $\{Y_1' + \cdots + Y_n' \geq \epsilon n\} \subset E \cup F$, where

$$\mathcal{I} := \big\{1 \leq i \leq n : X_i > n^\alpha\big\}, \qquad E := \Big\{|\mathcal{I}| \geq (\epsilon n)^\beta\Big\}, \qquad F := \Big\{\max_{1 \leq i \leq n} X_i \geq \tfrac{1}{B}(\epsilon n)^\beta\Big\}.$$

For $I \subset \mathbb{Z}$, denote $d(I) := \min\{|i - j| : i, j \in I, \ i \neq j\}$. Since, for any $I \subset \mathbb{Z}$ and integer $d \geq 1$, there exists a subset $I' \subset I$ such that $|I'| \geq |I|/d$ and $d(I') \geq d$, we obtain

$$\mathbb{P}(E) \leq \mathbb{P}\Big(\exists I \subset \mathcal{I}, \ |I| = \Big\lceil \tfrac{(\epsilon n)^\beta}{2n^\alpha+2}\Big\rceil, \ d(I) \geq 2n^\alpha + 1\Big).$$

Since the events $\{X_i > r\}_{i \in I}$ are independent for any finite $I \subset \mathbb{Z}$ and $0 \leq r < d(I)/2$, we have

$$\mathbb{P}(E) \leq \binom{n}{\left\lceil \frac{(\epsilon n)^\beta}{2n^\alpha+2}\right\rceil} \cdot \mathbb{P}(X_0 > n^\alpha)^{\left\lceil \frac{(\epsilon n)^\beta}{2n^\alpha+2}\right\rceil} \leq e^{\frac{(\epsilon n)^\beta}{n^\alpha}\log n - c(\epsilon n)^\beta} = e^{-\Omega(n^\beta)}.$$

Finally, it is immediate that $\mathbb{P}(F) \leq n \cdot \mathbb{P}(X_0 \geq \tfrac{1}{B}(\epsilon n)^\beta) = e^{-\Omega(n^\beta)}$. □

Remark 5.6. The bound in Lemma 5.5 is tight, as the following simple example shows. Let $(Y_i)_{i \in \mathbb{Z}}$ be independent unbiased coin tosses, and let X_i be the length of the streak of heads containing position i, i.e., $X_i := \max\{k + m : Y_j = 1 \text{ for } i - k \leq j < i + m, \ k, m \geq 0\}$. Clearly, X is a stationary sequence (in fact, it is ffiid with exponential tails) and X_0 has exponential tails. Moreover, since X_i is a stopping time with respect to $(\{Y_j\}_{|j-i| \leq n})_n$, it follows that X is stopping-like (with $\Delta = 1$). On the other hand, $\mathbb{P}(X_1^b + \cdots + X_n^b \geq an) \geq \mathbb{P}(Y_1 = \cdots = Y_{\lceil (an)^\beta\rceil} = 1) = 2^{-\lceil (an)^\beta\rceil}$.

Proof of Lemma 5.3. As Lemma 5.2 implies that both update methods yield the same probability for the event in question, we may assume here that update method **(B)** is used in the algorithm. Denote $L_w^\infty := \sup_n L_w^n$. Let $\mathcal{L}$ denote the vertices which remain loaded indefinitely. Let $v \in \mathbb{Z}^d$. For an integer i, we write $v + i$ for the element $v + ie_1 \in \mathbb{Z}^d$. Let us check that if, for some $k \geq 0$,

$$\mathcal{L}_k := \{0 \leq i \leq k : v + i \in \mathcal{L}\} \neq \emptyset \quad \text{or} \quad N_k := \sum_{i=0}^{k}(L_{v+i}^\infty + 1) > \sum_{i=0}^{k}|B_{\tau_{v+i}(X)}| =: M_k,$$

then v is satisfied at time N_k. Assume towards a contradiction that v is not satisfied at time N_k. Then, by Claim 5.1,

$$(v, -1) = (U_v^0, I_v^0) \prec (U_v^1, I_v^1) \prec \cdots \prec (U_v^{N_k+1}, I_v^{N_k+1}).$$

Thus, since $0 \leq I_v^n \leq L_{U_v^n}^\infty$ for all $n \geq 1$ by definition of L_u^n, we have $v + k \prec U_v^{N_k+1}$. In particular, the set of times

$$T := \{1 \leq n \leq N_k : U_v^n \prec U_v^{n+1}\}$$

at which the v-simulator moved its source location to the right is of size $|T| \geq k + 1$. Moreover, since, for $1 \leq n \leq N_k$, $n \in T$ if and only if case (iii) of the algorithm is executed at step n for the vertex v, which in turn occurs only if $U_v^{n-1} \notin \mathcal{L}$ and $I_v^{n-1} = L_{U_v^{n-1}}^{n-1} = L_{U_v^{n-1}}^\infty$, we conclude that

$$\mathcal{L}_k = \emptyset, \qquad |T| = k + 1, \qquad (U_v^{N_k+1}, I_v^{N_k+1}) = (v + k + 1, 0).$$

Note that if the input from some location (u, i) is transported by some v'-simulator by time n (i.e., $(v', u, i, w, j) \in D^n$ for some (w, j)), then $v' \preceq u$ and every v'' such that $v' \prec v'' \preceq u$ must be satisfied at time n. Thus, since by step $N_k + 1$, N_k inputs were transported from locations (u, i) with $v \preceq u \preceq v + k$, but no more than M_k inputs were transported by v'-simulators with $v \preceq v' \preceq v + k$, it follows that v is satisfied at time N_k, which is a contradiction. Hence, v is satisfied at time N_k.

We have thus shown that

$$\mathbb{P}(v \text{ is not satisfied at time } n) \leq \mathbb{P}(\forall k \geq 0 \; (\mathcal{L}_k = \emptyset \text{ and } N_k \leq M_k) \text{ or } N_k > n).$$

Using that $L_v^n \leq \sigma_v(Y^n) \leq m$ almost surely for some $m \geq 1$, and taking $k = \frac{n}{4m}$, we get

$$\mathbb{P}(v \text{ is not satisfied at time } n) \leq \mathbb{P}(\mathcal{L}_{\frac{n}{4m}} = \emptyset \text{ and } N_{\frac{n}{4m}} \leq M_{\frac{n}{4m}}).$$

Let a be such that $\mathbb{E}\sigma_v + 1 > a > \mathbb{E}|B_{\tau_v}|$, and note that

$$\mathbb{P}(\mathcal{L}_{\frac{n}{4m}} = \emptyset \text{ and } N_{\frac{n}{4m}} \leq M_{\frac{n}{4m}}) \leq \mathbb{P}(\mathcal{L}_{\frac{n}{4m}} = \emptyset \text{ and } N_{\frac{n}{4m}} \leq \frac{an}{4m}) + \mathbb{P}(M_{\frac{n}{4m}} \geq \frac{an}{4m}).$$

It remains to bound the terms on the right-hand side. Note that, if $u \notin \mathcal{L}$ then $L_u^\infty = \sigma_u(X)$. Thus, since $(\sigma_{v+i}(X) + 1)_{i \in \mathbb{Z}}$ is an i.i.d. sequence of bounded random variables with expectation strictly larger than a, standard large deviation bounds yield that $\mathbb{P}(\mathcal{L}_{n/4m} = \emptyset \text{ and } N_{n/4m} \leq \frac{an}{4m})$ is exponentially small in n (alternatively, we could appeal to Lemma 5.5 with the sequence $(m - \sigma_{v+i}(X))_{i \in \mathbb{Z}}$ to obtain the required stretched-exponential bound). Towards establishing the bound on the second term, observe that, by (3.1) and (3.2), $(\tau_{v+i})_{i \in \mathbb{Z}}$ is a non-negative stopping-like stationary sequence with exponential tails. Thus, since $|B_n| = O(n^{d+1})$ by (3.2), Lemma 5.5 implies that

$$\mathbb{P}(M_{n/4m} \geq \frac{an}{4m}) \leq e^{-\Omega(n^{1/(d+2)})} \qquad \text{as } n \to \infty. \qquad \square$$

5.3 Proof of Lemma 5.4

Let $\mathcal{F}_{v,r}$ denote the σ-algebra generated by $(X_{u,i})_{|u-v| \leq r, 0 \leq i \leq \sigma_u}$. Set $r_0 := 0$ and let r_n denote the smallest integer $r > r_{n-1}$ for which the random variables stated in the lemma are $\mathcal{F}_{v,r}$-measurable. To prove the lemma, it suffices to show that $r_n \leq r_{n-1} + 5\Delta n$ for $n \geq 1$, as this implies that $r_n \leq 5\Delta n^2$. We henceforth abbreviate "is $\mathcal{F}$-measurable" to "is in $\mathcal{F}$".

Let $n \geq 1$ and denote $r := r_{n-1}$. We aim to show that $\mathbb{S}_v^n$ and $\{Z_{w,j}^n\}_{|w-v| \leq \Delta n, j \geq 0}$ are in $\mathcal{F}_{v,r+5\Delta n}$, using that $\mathbb{S}_v^{n-1}$ and $\{Z_{w,j}^{n-1}\}_{|w-v| \leq \Delta(n-1), j \geq 0}$ are in $\mathcal{F}_{v,r}$. Throughout the proof, we repeatedly make use of the following easily verifiable properties:

$$ v \preceq U_v^n \preceq v + n e_1 \quad \text{and} \quad (W_v^n, J_v^n) \in v + B_n \quad \text{for all } v \in \mathbb{Z}^d \text{ and } n \geq 0, $$

which, in particular, by (3.2), imply that

$$ |U_v^n - v| \leq n \quad \text{and} \quad |W_v^n - v| \leq \Delta n \quad \text{for all } v \in \mathbb{Z}^d \text{ and } n \geq 0. $$

Step 1: Consider step n of the algorithm for v and let $\mathcal{C}_v^n \in \{\mathrm{i}, \ldots, \mathrm{iv}\}$ denote which case of the algorithm was executed. Let us show that $\mathcal{C}_v^n$ is in $\mathcal{F}_{v,r+n}$. To this end, we first check that the event $\{\mathcal{C}_v^n = \mathrm{i}\} = \{v \text{ is satified at time } n-1\} = \{(W_v^{n-1}, J_v^{n-1}) \notin B_{\tau_v(Z^{n-1})}\}$ is in $\mathcal{F}_{v,r}$. Indeed, since $(W_v^{n-1}, J_v^{n-1}) \in v + B_{n-1}$, this event depends only on (W_v^{n-1}, J_v^{n-1}) and $\{Z_{w,j}^{n-1}\}_{(w,j) \in v + B_{n-1}}$, both of which are in $\mathcal{F}_{v,r}$ by the definition of r and by (3.2).

Next, let us check that $L_{U_v^{n-1}}^{n-1}$ is in $\mathcal{F}_{v,r+n}$. Since U_v^{n-1} is in $\mathcal{F}_{v,r}$ and since $v' \preceq U_{v'}^{n-1} \prec v' + n e_1$ for all v', it suffices to check that the event $\{(v', u, i, w, j) \in D^{n-1} \text{ for some } w, j\}$ is in $\mathcal{F}_{v,r+n}$ for any $v - n e_1 \preceq v' \preceq u \prec v + n e_1$ and any i. Indeed, this event is in $\mathcal{F}_{v',r} \subset \mathcal{F}_{v,r+|v'-v|} \subset \mathcal{F}_{v,r+n}$.

Finally, we check that the event that U_v^{n-1} is loaded at time $n-1$ is in $\mathcal{F}_{v,r+n}$. Note that, by (4.3), $Y_{u,i}^{n-1} = X_{u,i}$ for $i \leq L_u^{n-1}$ and $Y_{u,i}^{n-1} = \emptyset$ for $i > L_u^{n-1}$. Thus, since U_v^{n-1} and $L_{U_v^{n-1}}^{n-1}$ are in $\mathcal{F}_{v,r+n}$ and since $|U_v^{n-1} - v| \leq n$, it follows that $(Y_{U_v^{n-1},i}^{n-1})_{i \geq 0}$ is in $\mathcal{F}_{v,r+n}$. Hence, as σ is a simple stopping-process, we have that $\{\sigma_{U_v^{n-1}}(Y^{n-1}) > L_{U_v^{n-1}}^{n-1}\}$ is in $\mathcal{F}_{v,r+n}$, showing that the event that U_v^{n-1} is loaded at time $n-1$ is in $\mathcal{F}_{v,r+n}$.

Step 2: Observe that $(U_v^n, I_v^n, W_v^n, J_v^n)$ is in $\mathcal{F}_{v,r+n}$. Indeed, this follows from step 1, since in any case of the algorithm, $(U_v^n, I_v^n, W_v^n, J_v^n)$ is a deterministic function of $(U_v^{n-1}, I_v^{n-1}, W_v^{n-1}, J_v^{n-1})$.

Step 3: Let us check that T_v^n is in $\mathcal{F}_{v,r+3\Delta n}$. To this end, it suffices to check that the event that v is the lexicographical-minimal element of $Q^{n-1}(W_v^{n-1}, J_v^{n-1})$ is in $\mathcal{F}_{v,r+3\Delta n}$, as T_v^n equals 1 in this case and 0 otherwise. For this, it suffices to check that the set $Q^{n-1}(W_v^{n-1}, J_v^{n-1})$ itself is in $\mathcal{F}_{v,r+3\Delta n}$. Since (W_v^{n-1}, J_v^{n-1}) is in $\mathcal{F}_{v,r}$ and since $|W_v^{n-1} - v| \leq \Delta(n-1)$, it suffices to check that $Q^{n-1}(w, j)$ is in $\mathcal{F}_{v,r+3\Delta n}$ for any (w, j) such that $|w - v| \leq \Delta n$. Fix such a (w, j). We need to show that the event $\{v' \in Q^{n-1}(w, j)\}$ is in $\mathcal{F}_{v',r+3\Delta n}$ for any $v' \in \mathbb{Z}^d$. Fix v' and note that $v' \notin Q^{n-1}(w, j)$ unless $|v' - w| \leq \Delta(n-1)$. Thus, we may assume that $|v' - w| \leq \Delta(n-1)$. Then, by what we have shown in step 1 and since $Z_{w,j}^{n-1}$ is in $\mathcal{F}_{v',r}$ (by the definition of r), the event $\{v' \in Q^{n-1}(w, j)\}$ is in $\mathcal{F}_{v',r+n} \subset \mathcal{F}_{w,r+n+|v'-w|}$. We have therefore shown that $Q^{n-1}(w, j)$ is in $\mathcal{F}_{w,r+2\Delta n} \subset \mathcal{F}_{v,r+2\Delta n+|w-v|} \subset \mathcal{F}_{v,r+3\Delta n}$.

Step 4: Observe that $\mathbb{S}_v^n$ is in $\mathcal{F}_{v,r+3\Delta n}$. Indeed, this follows immediately from steps 2 and 3 using that D_v^n is determined by $\{(U_v^t, I_v^t, W_v^t, J_v^t, T_v^t)\}_{1 \leq t \leq n}$.

Step 5: Let us show that $\{Z_{w,j}^n\}_{|w-v| \leq \Delta n, j \geq 0}$ is in $\mathcal{F}_{v,r+5\Delta n}$. To this end, let (w, j) be such that $|w - v| \leq \Delta n$ and denote

$$ D_{w,j}^n := \{(v', u, i) : (v', u, i, w, j) \in D^n\}. $$

By step 4, for any (v', u, i), the event $\{(v', u, i) \in D_{w,j}^n\}$ is in $\mathcal{F}_{v',r+3\Delta n} \subset \mathcal{F}_{w,r+3\Delta n+|v'-w|}$. Since $(v', u, i) \notin D_{w,j}^n$ unless $|v' - w| \leq \Delta n$, we conclude that $D_{w,j}^n$ is in $\mathcal{F}_{w,r+4\Delta n} \subset \mathcal{F}_{v,r+4\Delta n+|w-v|} \subset \mathcal{F}_{v,r+5\Delta n}$. Recall that $|D_{w,j}^n| \leq 1$ by (4.2). If $D_{w,j}^n = \emptyset$ then $Z_{w,j}^n = \emptyset$. Otherwise, $D_{w,j}^n = \{(v', u, i)\}$ and $Z_{w,j}^n = X_{u,i}$ for some (v', u, i) such that $|u - v'| \leq n$, in which case, $|u - v| \leq |u - v'| + |v' - w| + |w - v| \leq 3\Delta n$. It follows that $Z_{w,j}^n$ is in $\mathcal{F}_{v,r+5\Delta n}$. $\qquad \square$

6 Open questions

We have shown in Theorem 1.1 that the sub-critical Ising measure is fv-ffiid with stretched-exponential tails, and we know from Remark 1.3 that it is also ffiid with exponential tails. The following question naturally arises:

Question 6.1. Let $d \geq 2$ and let μ be the unique Gibbs measure for the Ising model on $\mathbb{Z}^d$ at inverse temperature $\beta < \beta_c(d)$. Is μ fv-ffiid with exponential tails?

A similar situation occurs in the more general setting of PCAs considered in Section 2, where we have shown in Theorem 2.1 that the limiting distribution of an exponentially uniformly ergodic PCA is fv-ffiid with stretched-exponential tails, and we know from Theorem 2.2 that it is also ffiid with exponential tails. A positive answer to the following natural question would yield a positive answer to the previous one:

Question 6.2. Let μ be the limiting distribution of an exponentially uniformly ergodic PCA (as defined in Section 2). Is μ fv-ffiid with exponential tails?

As we have mentioned in remarks after Theorem 1.1, the critical Ising measure is known to be ffiid, but is not known to be fv-ffiid. This question was raised by van den Berg and Steif [5, Question 1] and we reiterate it here:

Question 6.3. Let $d \geq 2$ and let μ be the unique Gibbs measure for the Ising model on $\mathbb{Z}^d$ at the critical inverse temperature $\beta = \beta_c(d)$. Is μ fv-ffiid?

References

[1] Scot Adams, *Følner independence and the amenable Ising model*, Ergodic Theory Dynam. Systems **12** (1992), no. 4, 633–657. MR-1200333

[2] Michael Aizenman, Hugo Duminil-Copin, and Vladas Sidoravicius, *Random currents and continuity of Ising model's spontaneous magnetization*, Comm. Math. Phys. **334** (2015), no. 2, 719–742. MR-3306602

[3] Michael Aizenman and Roberto Fernández, *On the critical behavior of the magnetization in high-dimensional Ising models*, J. Statist. Phys. **44** (1986), no. 3–4, 393–454. MR-857063

[4] Jacob van den Berg and Christian Maes, *Disagreement percolation in the study of Markov fields*, Ann. Probab. **22** (1994), no. 2, 749–763. MR-1288130

[5] Jacob van den Berg and Jeffrey E. Steif, *On the existence and nonexistence of finitary codings for a class of random fields*, Ann. Probab. **27** (1999), no. 3, 1501–1522. MR-1733157

[6] Roland L. Dobrushin, *The description of a random field by means of conditional probabilities and conditions of its regularity*, Theor. Probab. Appl. **13** (1968), 197–224.

[7] Robert G. Edwards and Alan D. Sokal, *Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm*, Phys. Rev. D (3) **38** (1988), no. 6, 2009–2012. MR-965465

[8] Hans-Otto Georgii, Olle Häggström, and Christian Maes, *The random geometry of equilibrium phases*, Phase transitions and critical phenomena, Vol. 18, Phase Transit. Crit. Phenom., vol. 18, Academic Press, San Diego, CA, 2001, pp. 1–142. MR-2014387

[9] Geoffrey Grimmett, *The random-cluster model*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 333, Springer-Verlag, Berlin, 2006. MR-2243761

[10] Olle Häggström and Jeffrey E. Steif, *Propp-Wilson algorithms and finitary codings for high noise Markov random fields*, Combin. Probab. Comput. **9** (2000), no. 5, 425–439. MR-1810150

[11] Matan Harel and Yinon Spinka, *Finitary codings for the random-cluster model and other infinite-range monotone models*, arXiv:1808.02333 (2018).

[12] Nate Harvey, Alexander E. Holroyd, Yuval Peres, and Dan Romik, *Universal finitary codes with exponential tails*, Proc. Lond. Math. Soc. (3) **94** (2007), no. 2, 475–496. MR-2308235

[13] Mark Huber, *Exact sampling and approximate counting techniques*, STOC '98 (Dallas, TX), ACM, New York, 1999, pp. 31–40. MR-1731559

[14] Mark Huber, *Perfect sampling using bounding chains*, Ann. Appl. Probab. **14** (2004), no. 2, 734–753. MR-2052900

[15] Michael Keane and Meir Smorodinsky, *Bernoulli schemes of the same entropy are finitarily isomorphic*, Ann. of Math. (2) **109** (1979), no. 2, 397–406. MR-528969

[16] Michael Keane and Meir Smorodinsky, *Finitary isomorphisms of irreducible Markov shifts*, Israel J. Math. **34** (1979), no. 4, 281–286 (1980). MR-570887

[17] Thomas M. Liggett, *Interacting particle systems*, Classics in Mathematics, Springer-Verlag, Berlin, 2005, Reprint of the 1985 original. MR-2108619

[18] Fabio Martinelli and Enzo Olivieri, *Approach to equilibrium of Glauber dynamics in the one phase region. I. The attractive case*, Comm. Math. Phys. **161** (1994), no. 3, 447–486. MR-1269387

[19] Donald S. Ornstein, *Ergodic theory, randomness, and dynamical systems*, Yale University Press, New Haven, Conn.-London, 1974, James K. Whittemore Lectures in Mathematics given at Yale University, Yale Mathematical Monographs, No. 5. MR-0447525

[20] Donald S. Ornstein and Benjamin Weiss, $\mathbf{Z}^d$-actions and the Ising model, Unpublished, 1977.

[21] James G. Propp and David B. Wilson, *Exact sampling with coupled Markov chains and applications to statistical mechanics*, Proceedings of the Seventh International Conference on Random Structures and Algorithms (Atlanta, GA, 1995), vol. 9, 1996, pp. 223–252. MR-1611693

[22] Jesús Salas and Alan D. Sokal, *Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem*, J. Statist. Phys. **86** (1997), no. 3-4, 551–579. MR-1438965

[23] Robert H. Swendsen and Jian-Sheng Wang, *Nonuniversal critical dynamics in Monte Carlo simulations*, Physical Review Letters **58** (1987), no. 2, 86.

[24] David Williams, *Probability with martingales*, Cambridge Mathematical Textbooks, Cambridge University Press, Cambridge, 1991. MR-1155402

[25] Chen N. Yang, *The spontaneous magnetization of a two-dimensional Ising model*, Phys. Rev. (2) **85** (1952), 808–816. MR-51740

Electronic Journal of Probability
Electronic Communications in Probability

Advantages of publishing in EJP-ECP

- Very high standards

- Free for authors, free for readers

- Quick publication (no backlog)

- Secure publication (LOCKSS[1])

- Easy interface (EJMS[2])

Economical model of EJP-ECP

- Non profit, sponsored by IMS[3], BS[4] , ProjectEuclid[5]

- Purely electronic

Help keep the journal free and vigorous

- Donate to the IMS open access fund[6] (click here to donate!)

- Submit your best articles to EJP-ECP

- Choose EJP-ECP over for-profit journals

---

[1] LOCKSS: Lots of Copies Keep Stuff Safe http://www.lockss.org/
[2] EJMS: Electronic Journal Management System http://www.vtex.lt/en/ejms.html
[3] IMS: Institute of Mathematical Statistics http://www.imstat.org/
[4] BS: Bernoulli Society http://www.bernoulli-society.org/
[5] Project Euclid: https://projecteuclid.org/
[6] IMS Open Access Fund: http://www.imstat.org/publications/open.htm