Sparse Probabilistic Models: Phase Transitions and Solutions via Spatial Coupling

THÈSE N^{O} 6625 (2015)

PRÉSENTÉE LE 26 JUIN 2015

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE THÉORIE DES COMMUNICATIONS
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Andrei GIURGIU

acceptée sur proposition du jury:

Prof. V. Cevher, président du jury
Prof. R. Urbanke, Dr N. Macris, directeurs de thèse
Prof. D. Gamarnik, rapporteur
Prof. H. Pfister, rapporteur
Prof. O. N. A. Svensson, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2015

Acknowledgements

The completion of this thesis would not have been possible without the constant support and infinite patience of my two advisors, Nicolas Macris and Rüdiger Urbanke. They have provided me with an amazing working environment by hosting me in the Information Processing Group at EPFL and they have introduced me to an amazing interdisciplinary field, at the crossroads of information theory, statistical physics and computer science. Their scientific and moral support was unbounded: they have always kept an open door and they were always ready to talk about any topic, be it research-related or not. Moreover, I have learned much from my experience as a teaching assistant for both of them. For these reasons and many others, it is very hard for me to fully express the gratitude that they deserve. I thank Rüdiger in particular for teaching me not to be afraid of seemingly complicated constructions, to always look for the simple core principles, and to favor down-to-earth approaches when solving problems. To Nicolas I will always remain indebted for all the hours spent in our discussions on our research problems, on statistical physics and on physics at large. I thank him for his patient and insightful explanations.

I am also grateful to the members of the thesis committee Volkan Cevher, David Gamarnik, Henry Pfister and Ola Svensson, for their careful review of the manuscript and their insightful comments.

Many thanks to the other senior members of IPG, Emre Telatar, Bixio Rimoldi and Olivier Lévêque for their lively and helpful discussions over the years. I am especially thankful to all of them for having assembled the vast and immensely useful collection of books that resides in Baghdad Café. I also had the pleasure to be a teaching assistant to Olivier in his probability courses, which was an exciting and unique experience.

Many thanks go to the permanent staff of IPG, Muriel Bardet, Françoise Behn and Damir Laurenzi, who keep the lab running smoothly and who have saved me numerous headaches on countless occasions. I will fondly remember the occasional treats provided by Muriel and Françoise, and also Damir's legendary cooking skills.

The lab would not have been such an extraordinary place were it not for the amazing colleagues that I encountered there. We have been a very diverse group, with many different cultural and academic backgrounds. This proved to be an extraordinary opportunity to learn, share and make good friends. I will first mention my excellent office mates Vassilis Kalofolias, Serj

Haddad and Young Jun Ko. I owe to Young Jun a great deal of things, including learning to come to work early in the mornings, a large amount of useful Linux hacks and vi tricks, and the many hours spent talking about linguistics, machine learning and everything else. Also many mountain peaks were climbed, many snowy slopes were skied, many steep uphill roads were biked in his company. I am grateful to Marc Vuffray for the memories of the time spent visiting Boston and Istanbul, the lengthy discussions on very wide-ranging subjects and his teachings of mathematical physics and of the replica method which I very much enjoyed. I also thank Marc Desgroseilliers for keeping the spirits high in the lab, for staunchly supporting the cause of Mathematics and for introducing me to rock climbing and ski touring, which have become some of my favorite pastimes. Special thanks go to Hamed Hassani for his kind and selfless help, to Rajai Nasser for the occasional excursions in formal logic and brain science, to Rafah El-Khatib for her cheerful spirit, to Marco Mondelli for being an awesome fellow TA and for the excellent pasta he cooks, to Stefano Rosati for the spirit of adventure, to Karol Kruzelecki for his kindness and sense of humour, to Adrian Tărniceriu for his permanent good mood and for the jokes, to Vahid Aref and Serj for opening my mind to the Persian and Lebanese cultures, to Mani Bastani Parizi, Alla Merzakreeva, Mine Alsan, Saeid Haghigatshoar, Nicu Chiurtu, and all the past and present members of IPG for making this group what it is today.

I am indebted to Dan Alistarh, who convinced me to come to Lausanne in the first place and with whom I worked on a project in the lab of Rachid Guerraoui before starting my PhD. I am very thankful to Rachid for this opportunity. I also thank the fellow PhD students with whom I shared a huge office space in our first year, for the memory of those days.

I extend my thanks to my friends from Lausanne and from all over the world, who have brought joy in my life and so contributed indirectly to the present endeavour. I am very grateful to Iulia Cârjeu for the years we spent together in Lausanne, and especially for the moral support she has provided me with when things were not going so well. I also thank Dan Tomozei and Ghid Maatouk for their wonderful music and for taking me out to enjoy cultural life from time to time. Hearty thanks to Alin Iacob, Irina Calciu, Mihai Hărănguș and other friends who have been far away and yet close.

This work is dedicated to my late father, Dan Andrei, who many years ago instilled in me a love of Mathematics and Science. I express my infinite gratitude to my mother Adriana, brother Mihai and grandparents Aurel, Natalia and Valeria for their unceasing encouragement, support, affection and kindness that they have provided throughout the years. Nothing would have materialized without them.

Lausanne, 27 April 2015 A. G.

Abstract

This thesis is concerned with a number of novel uses of *spatial coupling*, applied to a class of probabilistic graphical models. These models include *error correcting codes*, random *constraint satisfaction problems* (CSPs) and statistical physics models called *diluted spin systems*. Spatial coupling is a technique initially developed for channel coding, which provides a recipe to transform a class of sparse linear codes into codes that are longer but more robust at high noise level. In fact it was observed that for coupled codes there are efficient algorithms whose decoding threshold is the optimal one, a phenomenon called *threshold saturation*. The main aim of this thesis is to explore alternative applications of spatial coupling. The goal is to study properties of uncoupled probabilistic models (not just coding) through the use of the corresponding spatially coupled models. The methods employed are ranging from the mathematically rigorous to the purely experimental.

We first explore spatial coupling as a proof technique in the realm of LDPC codes. The Maxwell conjecture states that for arbitrary BMS channels the optimal (MAP) threshold of the standard (uncoupled) LDPC codes is given by the Maxwell construction. We are able to prove the Maxwell Conjecture for any smooth family of BMS channels by using (i) the fact that coupled codes perform optimally (which was already proved) and (ii) that the optimal thresholds of the coupled and uncoupled LDPC codes coincide. The latter statement is proved using the interpolation method, by gradually transforming the coupled code distribution into (a) a distribution of large standard codes and (b) a distribution of L independent standard codes, where L is the chain length of the coupled code. By monotonicity along the gradual transformation it follows that the free energy of (a) and (b) provide lower and upper bounds for the free energy of the coupled code. This with the fact that the free energy of (a) and (b) are asymptotically equal implies (ii). The method is used to derive two more results, namely the equality of GEXIT curves above the MAP threshold and the exactness of the averaged Bethe free energy formula derived under the RS cavity method from statistical physics.

As a second application of spatial coupling, we show how to derive novel bounds on the phase transitions in random constraint satisfaction problems, and possibly a general class of diluted spin systems. In the case of coloring, we investigate what happens to the dynamic and freezing thresholds. The phenomenon of threshold saturation is present also in this case, with the dynamic threshold moving to the condensation threshold, and the freezing moving to colorability. These claims are supported by experimental evidence, but in some cases, such as

the saturation of the freezing threshold it is possible to make part of this claim more rigorous. This allows in principle for the computation of thresholds by use of spatial coupling. The proof is in the spirit of the potential method introduced by Kumar, Young, Macris and Pfister [KYMP14] for LDPC codes.

Finally, we explore how to find solutions in (uncoupled) probabilistic models. To test this, we start with a typical instance of random K-SAT (the *base problem*), and we build a spatially coupled structure that locally inherits the structure of the base problem. The goal is to run an algorithm for finding a suitable solution in the coupled structure and then "project" this solution to obtain a solution for the base problem. Experimental evidence points to the fact it is indeed possible to use a form of unit-clause propagation (UCP), a simple algorithm, to achieve this goal. This approach works also in regimes where the standard UCP fails on the base problem.

Keywords: spatial coupling, probabilistic models, LDPC codes, interpolation method, Maxwell conjecture, threshold saturation, sparse graph coloring, random formula satisfiability, freezing, unit clause propagation

Résumé

Cette thèse concerne un certain nombre de nouvelles utilisations du *couplage spatial* (*spatial coupling*), appliquées à une classe de modèles graphiques probabilistes. Parmi ces modèles on compte les *codes correcteurs d'erreurs*, les *problèmes de satisfaction de contraintes* (CSP) aléatoires et des modèles de physique statistique réunis sous le nom de *systèmes de spins dilués* . Le couplage spatial est une technique initialement développée pour le codage de canal, qui fournit un procédé pour transformer une classe de codes linéaires creux dans des codes plus longs mais aussi plus robustes en cas de bruit élevé. En fait, il a été observé que pour les codes couplés, il existe des algorithms efficaces dont le seuil de décodage devient optimal, un phénomène appelé *saturation de seuil*. Le thème principal de la thèse aborde le couplage spatial d'un autre point de vue. L'objectif est d'étudier les propriétés de modèles probabilistes non couplées (pas seulement de codage) en utilisant des modèles spatialement couplés correspondants. Les méthodes employées varient entre méthodes mathématiquement rigoureuses et méthodes fondées sur des expériments numériques.

En premier, nous explorons le couplage spatial comme technique de démonstration dans le domaine de codes de contrôle de parité de faible densité (LDPC). La Conjecture Maxwell indique que pour des canaux à entrées binaires sans mémoire et symé triques (BMS) arbitraires le seuil optimal (MAP) des codes LDPC standard (non couplés) est donné par la construction Maxwell. Nous sommes en mesure de prouver la Conjecture Maxwell pour toute famille lisse de canaux BMS en utilisant (i) le fait que les codes couplés fonctionnent de manière optimale (qui a été déjà accompli) et (ii) que les seuils optimaux des codes LDPC couplés et non couplés coïncident. Ce dernier fait est établi en utilisant la méthode d'interpolation, en transformant progressivement la distribution probabiliste du code couplé en (a) une distribution d'un code standard plus long et (b) une distribution de L codes standard indépendants, où L est la longueur de la chaîne du code couplé. Par monotonie tout au long de la transformation progressive il résulte que les énergie libre de (a) et (b) fournissent des bornes inférieures et supérieures pour l'énergie libre du code couplé. Le fait que les énergies libres de (a) et de (b) sont asymptotiquement égales implique (ii). La méthode est utilisée pour obtenir deux résultats additionnels : l'égalité des courbes GEXIT au-dessus du seuil MAP et l'exactitude de la formule de énergie libre de Bethe moyenne dérivée selon la méthode de cavité en symmetrie des repliques (RS) de physique statistique.

Comme une deuxième application du couplage spatial, nous montrons comment obtenir des

nouvelles bornes sur les transitions de phase dans des problèmes aléatoires de satisfaction de contraintes, et, éventuellement, une classe générale de systèmes de spins dilués. Dans le cas de la coloration, on étudie ce qui se passe avec les *seuils dynamiques* et *de rigidité* (freezing or rigidity threshold). Le phénomène de saturation de seuil est présent également dans ces cas, avec le seuil dynamique se déplaçant vers le seuil de condensation, et celui de rigidité vers le seuil de coloration. Ces affirmations sont étayées par des preuves expérimentales, mais dans certains cas, tels que la saturation du seuil de rigidité, il est possible de justifier cette affirmation d'une manière plus rigoureuse. Ceci permet en principe le calcul des seuils par utilisation de couplage spatial. La preuve est dans l'esprit de la méthode du potentiel présenté par Kumar, Young, Macris et Pfister [KYMP14] pour les codes LDPC.

Enfin, on explore comment trouver des solutions dans des modèles probabilistes non couplés. Pour tester cela, nous commençons par un échantillon typique de K-SAT aléatoire (*le problème de base*), et nous construisons une structure spatialement couplée qui hérite localement la structure du problème de base. Le but est d'exécuter un algorithme pour trouver une solution de la structure couplée, puis "projeter" cette solution pour obtenir une solution du problème de base. Des expériences numériques soulignent le fait qu'il est possible d'utiliser un algorithme simple (de type unit clause propagation, UCP) sur la structure couplée pour atteindre cet objectif. Cette approche fonctionne aussi dans des régimes où l'UCP échoue sur le problème de base.

Mots-clés : Couplage spatial, modèles probabilistes, codes LDPC, méthode d'interpolation, conjecture Maxwell, seuil de saturation, coloration des graphes creux, satisfaction des formules aléatoires, rigidité, unit clause propagation

Contents

List of Figures

List of Tables

1 Introduction

The running theme of this thesis is *spatial coupling*, a technique that was originally developed to obtain better error-correcting codes. To convey the basic idea of spatial coupling, let us illustrate it with the following scenario.

Take a linear chain of finite length and place at each link (or "position") a large number of bits. This chain is the "spatial dimension". The game is as follows. We only get partial information about the values (for example some of them we see and some we don't, but more complicated scenarios are possible). We also know that certain configurations of bits are not allowed. This latter fact is expressed through the existence of *constraints*, each of which is telling us that a certain set of bits cannot take certain values. Each constraint has the property that the bits involved in it are situated close to one another on the spatial chain. The goal is to use this information in order to uncover the hidden bits, a process we call *decoding*.

At the two ends of the chain, it so happens that we have more information available and thus the decoding is easier to perform. As we decode the bits at the boundaries, the constraints that they participate in enable us to infer values of bits further inside the chain. Consequently, a wave of decoding is produced which leads to the eventual discovery of all hidden bits. This is the mechanism underlying spatial coupling, and it has proved successful in situations where the standard way of coding (without spatial coupling) has failed.

The technique of spatial coupling was developed first for a class of codes called LDPC codes, but soon found other applications, notably in compressed sensing. More generally, it can be thought of as a paradigm that applies to graphical models of all sorts. Then whenever we deal with a task where it is up to us to design the constraints (and coding is the best example for this), spatial coupling offers us a recipe to alter the design and obtain some potential benefits.

However, here we will be more interested in a different line of thought. This arises from the phenomenon of threshold saturation, which we briefly illustrate here by recurring to our earlier game with hidden bits. The number of hidden bits quantifies the amount of information withheld from us. Naturally, the more information is hidden, the less likely it is that we are

able to decode. As we increase the amount of withheld information, it turns out that there is a "fundamental" threshold, beyond which it is impossible to decode even theoretically, since there is more than one solution to the problem. Apart from this theoretical threshold, there is another limitation that sets in much earlier. Let us call it the algorithmic threshold: it is the point up to which we can decode "reasonably fast". In between the two thresholds there still exists a unique solution to the problem, but no fast enough algorithm is known that is capable of finding it. It turns out that the "fundamental" threshold is the same no matter whether we use spatial coupling or not. However, with spatial coupling, the algorithmic threshold moves up to the theoretical threshold. Thus, using simple algorithms we are able to decode optimally. The key fact now is that the algorithmic threshold "saturates", i.e. it moves exactly to the place of the hard threshold. In particular, if we did not know the value of the hard threshold a priori, spatial coupling would offer a way to compute it by seeing what the algorithmic one is.

This thesis does not focus on the engineering task of building better codes or compressed sensing schemes, which has been the main direction of research in spatial coupling, but rather considers the theoretical question of what can be learned about basic problems from their spatially coupled versions. We have first done this in the case of error-correcting codes, in order to prove rigorously a conjecture regarding the location of the theoretical threshold of standard LDPC codes. But in fact this approach has a much wider scope of application, since we are not limited to design problems, where the goal is to come up with a smart placement of constraints, for example. We can now examine models which were already of interest in statistical physics community, such as random graph coloring or logical formula satisfiability. For these problems it might not make sense a priori to consider spatially coupled instances in themselves as the focus of research. But if there are cases where for example threshold saturation holds, we can gain insight into the original problem from the spatially coupled version.

Another thread that we explore is the possibility of algorithms that run on special spatially coupled instances of a problem, in such a way that from their output we can construct solutions for an original instance of non-coupled version. This could potentially have implications for a wide scope of more practical problems. However, research in this direction so far is in its infancy.

The work is both theoretical and experimental, in the sense that rigorous results were sought and sometimes found, sometimes only part of the overall picture could be made rigorous, and sometimes the picture itself needed to be discovered and the phenomena were not so well understood in order to be turned into mathematical proofs.

1.1 Outline of the thesis

Apart from this introduction, the thesis is divided into three parts, which correspond to three broad applications of spatial coupling as a tool. They are ordered by the degree of mathematical rigor which supports the facts, from fully rigorous to almost fully experimental.

- In the rest of this chapter, we review the basics of constraint satisfaction problems and channel coding, with a focus on LDPC codes. We introduce spatial coupling in the wider context of sparse graphical models. These problems have traditionally been studied by different communities from different perspectives, most notably by statistical physicists, computer scientists and information theorists and we will need to work with specific methods coming from all those different worlds. The parts of the introduction that introduce the bulk of notation are those on random CSPs (Sections 1.3 and 1.5), Section 1.6 on statistical physics and the cavity method and Section 1.8 on basics of information theory and LDPC codes. The chapter ends on the same theme as it started, with spatial coupling in action.

- In the second chapter we make use of spatial coupling as a proof technique. We prove that in the context of LDPC codes a spatially coupled and a corresponding standard code have the same theoretical ("MAP") threshold. This also proves the location of the MAP threshold for standard LDPC codes, which was previously conjectured to be given by the Area threshold formula. It proves also the correctness of the replica symmetric (RS) approximation inspired from statistical physics. To do this we employ the *interpolation method*, a proof technique to show inequalities of thermodynamic potentials between different types of random structures. A novelty in our proof is the application of the interpolation method to factor graphs with arbitrary bit degree distributions.

- In the third chapter we analyze spatial coupling in the context of constraint satisfaction problems, with a focus on random graph coloring. We briefly explain the replica symmetry breaking (RSB) formalism, which allows us to determine the position of the dynamic, condensation, freezing and coloring thresholds. We are able to show that by spatially coupling the survey-propagation equations one obtains that the SP threshold saturates to the coloring threshold. We also obtain numerical evidence that for spatially coupled random graphs the dynamic threshold saturates to the condensation threshold and that there is no phase where clusters of colorings contain frozen nodes.

- In the fourth chapter we investigate the possibility of finding solutions to uncoupled random logical formulas by transforming standard instances of K-SAT into coupled ones and then running *unit clause propagation* (UCP), a simple greedy algorithm, on the coupled formulas. The idea is to project the solution obtained on the coupled formula onto a solution for the original formula. We describe a mechanism which drives UCP towards a solution which can be projected. This results in a modified form of UCP, which finds satisfying assignments in a regime where the original UCP fails.

1.2 Optimization and factor graphs

All problems that will concern us, including graph coloring, formula satisfiability and LDPC coding can ultimately be cast in the following form. We have a number of variables $\sigma_1, \ldots, \sigma_N$, each of which can take values from a finite set Ω. The goal is to find an *assignment*, i.e. a tuple

$(\sigma_1,\dots,\sigma_N)$ that maximizes some real-valued target $\Psi(\sigma_1,\dots,\sigma_N)$. We require this function Ψ to have a decomposition into factors that depend on a small number of variables:

$$\Psi(\sigma_1,\dots,\sigma_N) = \psi_1(\sigma_{i_1^1},\dots,\sigma_{i_{K_1}^1})\cdots\psi_M(\sigma_{i_1^M},\dots,\sigma_{i_{K^M}^M}). \tag{1.1}$$

In order to avoid this clumsy notation, we adopt the following conventions, which are quite common in the literature. We index the factors by a, b, etc. and the variables by i, j, etc. Keeping things simple requires a fair degree of notation abuse: we will simply know by the choice of letter whether we mean to index variables or factors. Also, it may be that these appear alone as summation indices, in which case it is implied that they range over all variables/factors. By ∂a we denote the set of variable indices on which the factor ψ_a depends functionally. Also, by $\sigma_{\partial a}$ we mean the values of the said variables, so that we can write things like $\psi_a(\sigma_{\partial a})$ compactly. But we can also think of ∂a as a set, so we can write $i \in \partial a$ if factor ψ_a depends on variable σ_i. Likewise, by ∂i we mean the set of factor indices that depend on i. Moreover, we will write $\partial i \setminus a$ as a shorthand for $\partial i \setminus \{a\}$ and likewise for $\partial a \setminus i$. Then (1.1) can be written succinctly as

$$\Psi(\underline{\sigma}) = \prod_a \psi_a(\sigma_{\partial a}), \tag{1.2}$$

where the underline is used to emphasize that the quantity is a vector.

The structure of the decomposition (1.1) can be described by a *factor graph* [KFL01]. By this we mean the bipartite graph constructed in the following way. There are N nodes, one for each variable, and M nodes, one for each of the factor functions $\psi_1,\dots,\psi_m$. The former we call *variable nodes* and the latter *function nodes*. We put an edge between a function node ψ_a and a variable node σ_i if $i \in \partial a$ (or equivalently, $a \in \partial i$). In figures we will usually represent variable nodes by circles and factor nodes by squares.

1.3 Constraint satisfaction problems

In constraint satisfaction problems (*CSP*s) we are interested in finding or counting assignments that fulfill certain constraints. A constraint is a logical predicate that depends on a subset of the variables. If an assignment makes the predicate true, we say that the assignment *satisfies* the constraint, otherwise we say that the assignment *violates* the constraint. The assignment is a *solution* to the problem if it satisfies *all* the constraints in the problem. The nature of these constraints differs from problem to problem, and the terminology used to refer to constraints, assignments, etc. may also be problem-specific. We associate to each constraint a a binary *cost* $H_a(\sigma_{\partial a})$, which is 0 when the constraint is satisfied and 1 otherwise. The goal is to minimize the total cost $H(\underline{\sigma}) = \sum_a H_a(\sigma_{\partial a})$. Note that this has the same structure as (1.2), but we prefer to keep it in summation form for reasons that will become apparent soon.

Here is a collection of common CSPs:

- Maximum Independent Set. In this case the variables take values in $\{0,1\}$. All constraints involve exactly two variables and are violated when both variables are 1. The structure of this problem is thus a graph on the variables, with the edges corresponding to the constraints. A valid assignment is one for which no edge connects variables that are 1. Equivalently, a valid assignment corresponds to a subset of the set variables (those that have value 1) on which the induced graph has no edges. We call such a subset an independent set. One can immediately see that the all-zeros assignment is a valid one (i.e. the empty set is independent), and that independent sets are closed under inclusion. The questions that can arise are what is the maximum size of an independent set, finding such an independent set, counting them, etc. The problem of deciding whether a graph has an independent set of a certain size is NP-complete (we refer the reader to [Pap03] for NP-completeness reductions).

- Graph Q-Coloring (Q-COL). We are given a number of colors Q and the variables take values ("colors") in $[Q] = \{1,\dots,Q\}$. In this context, assignments will be referred to as (Q-)colorings. As in the case of Independent Set, the constraints involve two variables and for this reason we will continue to use graph terminology. Each constraint ensures that its two variables do not take the same color. In other words, for each edge a with $\partial a = \{i,j\}$, we have $H_a(\sigma_i,\sigma_j) = \mathbb{1}[\sigma_i = \sigma_j]$. Questions of interest here are the existence of colorings, finding colorings, etc. For $Q < 3$ the problem is easy, in particular 2-colorings exist if and only if the graph is bipartite, whereas Graph 3-Coloring is already NP-complete [Pap03].

- K-Satisfiability (K-SAT). In this case the variables are again binary and it helps to think of them as the logical values *true* and *false*. Each constraint (in SAT terminology: *clause*) involves K variables. Out of the 2^K possible configurations of these variables, the clause is violated on exactly one, and this violating configuration is clause-dependent.[1] A clause is determined by the indices of variables that take part (note that the order of those matters) *and* the violating configuration. The latter is not encoded by the usual factor graph.[2] Finding satisfying assignments in formulas is one of the most famous problems in complexity theory. For $K < 3$, the problem is again easy, but for $K \geq 3$, it is proven to be NP-complete [Coo71a].

- K-XOR-Satisfiability (K-XORSAT). The structure of this problem is similar to that of K-SAT, in that clauses involve K literals, but this time a clause is satisfied when the exclusive disjunction of the literals is true. Equivalently, if we consider the values of variables to be in $\{0,1\}$, the clause is satisfied when the sum modulo 2 is equal to either 0 or 1, a value different for each clause, representing the parity of the number of negated

[1] In logical terms, a clause is a disjunction of *literals*, where a literal is either a variable or a negation of a variable. The *sign* of a literal is the information of it being negated or not. For example, a 3-clause $\partial a = \{i,j,k\}$ that is violated on the K-tuple $(\sigma_i,\sigma_j,\sigma_k) = (1,0,1)$ could be written as $\overline{\sigma_i} \vee \sigma_j \vee \overline{\sigma_k}$. Then $\overline{\sigma_i}$, σ_j, $\overline{\sigma_k}$ are literals, of which the second has a positive sign and other a negative one. Rather than using this logical language common in computer science, we will mostly express the formula algebraically, with sums and products. The condition of satisfiability of an assignment is expressed as a conjunction of clauses, which we call a *formula*.

[2] This can be changed by making use of two types of edges, continuous and dashed, for example.

literals present in the clause. For this reason this type of clause is called a *parity check*. This leads to a formulation of K-XORSAT as a linear system of equations in $GF(2)$, which is solvable in polynomial time by Gaussian elimination. Even though it is not hard, XORSAT is still interesting because certain phenomena that occur in the random version of K-SAT, such as clustering, are also present in random XORSAT. In random XORSAT, their presence is much easier to prove. Furthermore, by the nature of its clauses XORSAT has a similar structure to parity check codes, and some proof methods work equally well on both.

1.4 Adversarial hardness

The above problems have been the main subject of scrutiny in the early days of complexity theory. The type of results that one typically obtains within the framework of complexity theory give worst-case guarantees. For example, XOR-SAT above is clear to be in the class P of polynomial time algorithms because every single instance in this class can be solved in polynomial time (the time needed to solve a linear system of equations). This is, however, not true when one considers K-SAT for $K \geq 3$, unless P is equal to NP[3], where NP is the class of problems whose solution is checkable in polynomial time.

Much of the fame of these problems stems from their status as benchmarks of hardness. In fact 3-SAT was the first class of problems to be proved NP-complete [Coo71b], i.e. at least as hard as any other problem in NP. This follows from the nature of SAT: the ability of logical formulas to encode instances of other problems. It is not clear, however, how many or what proportion of all the instances of K-SAT is actually hard. Complexity theory has not yet answered that question (that is, not even modulo P $\neq$ NP), so it could still be that in fact most problems that one could think of are actually easy and the hard ones are concentrated in some small and hard to reach cluster of strange formulas.

The latter is one reason why random instances are interesting to study. It is believed that genuinely hard instances can be obtained by just sampling an instance of Q-COL or K-SAT at random. Methods of statistical physics can and have been used extensively to assess the hardness of random instances, but so far no relation to the adversarial hardness in the sense of complexity theory has been established rigorously.

As a historical side note, the (non-random) coloring problem is the oldest of the group. The initial focus was on the minimum number of colors needed to color countries on any map (i.e. coloring of a planar graph), which turns out to be four. A first attempt at proving this was made in 1876 by Kempe [Kem79], but his proof was flawed. Nonetheless, some of his constructions have found use much later [Mol12]. It was only a century later that a (computer-aided) proof

[3] It is one of the most famous conjectures in all of mathematics that this is not the case. This question withstood the efforts of generations of researchers by now, and it does not look like we are any closer to proving it. Yet the consequences of the conjecture being false are so mind-boggling that very few people would actually not believe in it.

was found for the Four Color Theorem by Appel and Haken [AH89].

1.5 Random instances

Two of the first (and most popular) models of random graphs were introduced by Erdős and Rényi and by Gilbert in the 1950s [ER59, Gil59], and a significant body of tools and techniques to deal with them has been developed in the meanwhile [Bol01, JLR11]. Following established terminology, we will subsequently refer to the random graph model where potential edges appear independently with probability p as the *Erdős-Rényi model*, even though this is in fact the version introduced by Gilbert.[4]

The way in which we sample the instances is just a generalization of the aforementioned model [GM75, Łuc91]. We need to first draw the factor graph at random, and also draw at random any additional information (like the signs of the literals for each clause in case of K-SAT). We assume here a model in which all function nodes have the same degree, let us call it K (in case of coloring, $K = 2$). The factor graph is sampled as follows. For each function node in the graph, independently choose K variable nodes uniformly at random and link them to the function node.[5] In the case of boolean satisfiability, the sign of the K literals is chosen by independently flipping fair coins. For coloring, after the generation, we purge the resulting graph of checks that connect to the same node, since those make the graph uncolorable. This will affect us little, since the number of such checks is $O(1)$ w.h.p. and we are interested in the large-N behaviour.

In order to generate instances, we need to specify the number of variables N and the number of function factors M. We will be exclusively interested in the case where the ratio of M and N is fixed and N tends to infinity, since for coloring and formula satisfiability this is the scaling where interesting phenomena are observed. Note that this scaling means we consider *sparse* factor graphs, i.e. those where the node degrees are $O(1)$ w.h.p.[6] In the case of graph coloring, the main parameter will be the average node degree, $\alpha = \frac{2M}{N}$. In the case of K-SAT it is just the ratio of the number of clauses to the number of variables, $\alpha = \frac{M}{N}$. We maintain this inconsistent notation since important thresholds are usually quoted for each problem in terms of these parameters.

By varying this parameter, we can find ourselves in regimes which are qualitatively very different. Thus for α small enough, the typical instance that we sample is algorithmically easy: even greedy algorithms can find solutions with high probability. As α increases, we pass to

[4] The original Erdős-Rényi model prescribes apriori the number M of edges, and the set is chosen uniformly at random from the $\binom{\binom{N}{2}}{M}$ possibilities.

[5] The question might arise whether the K variable nodes should be chosen with repetition or not. For the problems we are considering will not make a difference, so in order to simplify matters, let us assume it is chosen with repetition.

[6] When we say that an event $\mathscr{E}_n$ indexed on n happens *with high probability* we mean that $\Pr[\mathscr{E}_n] = 1 - o(1)$ as $n \to \infty$.

a regime where the solutions are fewer and they are more likely to concentrate into hard to reach clusters. Roughly said, this latter phenomenon is what makes the problem hard. Beyond yet another critical value of α, the instance becomes unsolvable with high probability. We call such values of α where the structure of the solution space has a certain property w.h.p. below and another one w.h.p. above *sharp thresholds*, or simply *thresholds*.[7]

The latter threshold, which we denote by α_s, is the one that characterizes the existence of solutions. We call this the coloring threshold in the case of Q-COL or satisfiability threshold in the case of K-SAT. In the most general setting, its existence (i.e. the fact that for $\alpha < \alpha_s$ the CSP is solvable w.h.p., whereas for $\alpha > \alpha_s$ it is unsolvable w.h.p.) is not completely settled mathematically. However, a result of Friedgut [FB99] comes very close to this ideal: it proves the existence of a threshold sequence[8].

In the case of K-SAT, it was shown very recently by Ding, Sly and Sun that there is in fact a sharp satisfiability threshold for large enough K, and moreover its location is given by the *Survey Propagation* equations [DSS14]. It is conjectured, however, that the SP equations predict the threshold for any $K \geq 3$ and also in the case of Q-COL for any $Q \geq 3$.

Bounds were obtained for the location of the solvability thresholds for most common CSPs. The upper and lower bounds are typically derived by different flavors of the first moment method and the second moment method, respectively [AM97, AF99, ANP05, AP04, COV13, COE14].

Certainly one application of random instances is to provide a testbed for general methods developed by statistical physicists, such as the cavity method, more of which we will see later. Also, as mentioned earlier, we have strong evidence so far that for many NP-complete problems, the typical random instances can still be very hard to solve. This has applications in bench-marking algorithms for solving the hard problem in question, since it is very hard to invent nonrandom instances that are consistently hard for all algorithms.

1.6 Statistical physics

Tools inspired from statistical physics have been successfully used to study random graphical models. We focus on the cavity method, which we introduce below and which is used to predict relevant thresholds in both random CSPs and random codes. We first concentrate on random CSPs and leave out the application to coding for the next section.

[7] The qualifier *sharp* is used in opposition to *coarse* thresholds, where the said properties do not hold w.h.p. above and below, i.e. the transition is smoother. See [FB99] for more details.

[8] It proves the existence of a sequence $\alpha_s^{(N)}$ so that for any $\epsilon > 0$ the CSP is solvable w.h.p. if we take the parameter $\alpha^{(N)}$ to depend on N itself and to be less than $\alpha_s^{(N)} - \epsilon$. It is, however, not shown that $\alpha_s^{(N)}$ converges.

1.6.1 The setting

Problems that are represented as factor graphs have a long tradition in statistical physics. We are looking at probability measure over a large number of variables σ_i, also called *spins*. The mass at each spin configuration is influenced by factors of the type $\psi_a(\sigma_{\partial a})$, each of which depend on a restricted number of spins. The probability measure, called *Gibbs measure*, is given by

$$\mu(\underline{\sigma}) = \frac{1}{Z}\prod_a \psi_a(\sigma_{\partial a}), \qquad Z = \sum_{\underline{\sigma}}\prod_a \psi_a(\sigma_{\partial a}). \tag{1.3}$$

Usually the factors ψ are are given by *energy penalties* of the form $H_a(\sigma_{\partial a})$ that are incurred when the spins take certain values, by the relation

$$\psi_a(\sigma_{\partial a}) = e^{-\beta H(\sigma_{\partial a})}, \tag{1.4}$$

where β is a parameter called *inverse temperature*. The connection with CSPs is obtained by interpreting H_a as the cost and sending $\beta \to \infty$ ("the zero-temperature regime"). Then the Gibbs measure concentrates on the spin configurations of least energy. Note that (1.3) has the product form of (1.2). The normalization factor Z, called *partition function*[9], ensures that μ is a probability measure.

Note that for random instances there are now two levels of randomness. The "inner" one is given by the Gibbs distribution for any fixed instance, while the "outer" one is the probability distribution of instances. We will tend to avoid terminology like *probability*, *distribution*, *expectation* for the "inner" randomness, and rather use terms such as *mass*, *measure*, *average*, etc. The former, together with the usual notation Pr, $\mathbb{E}$, will be reserved for the "outer" randomness.

Spin systems were introduced in the 1920s by Ising with his model of ferromagnetism. This model assumes a fixed graph, the integer lattice [Isi25]. In time, numerous other models were put forward: we mention the Sherrington-Kirkpatrick model [SK75] and the diluted spin glasses [EA75]. In the former, the model is on the complete graph, and there is an energy penalty for each pair of spins of the form $J_{ij}\sigma_i\sigma_j$, with J_{ij} being independent Gaussians. In the *diluted spin glass* model, there are M energy penalty contributions, each connected randomly to K spins. This is exactly the Erdős-Rényi model of random CSPs that we have introduced before, and this is the one we will focus on. In the case of Q-COL, there are Q possible spin values: such models are called Potts models in the physics literature [Wu82].

One of the pursuits of statistical physics is the study of *phase transitions*. Intuitively, it can happen that when we vary continuously parameters of the model, such as temperature, we

[9] Sometimes the partition function is written $Z(\beta)$, making the dependence on temperature explicit (which also explains the term *function*). This dependence is not so important for us, as we will almost always work in the limit $\beta \to \infty$

obtain drastic changes of the Gibbs measure as we cross these values. These phenomena are called phase transitions. One common example is the transition of water from say, solid phase to liquid phase at 273K, or closer to our focus, the transition from the ferromagnetic phase to the paramagnetic phase in a piece of iron as temperature crosses 1043K.

Thus, the phase transitions are intimately connected with the thresholds introduced earlier. Phase transitions are actually defined as the values of parameters where the *thermodynamic potentials* are not analytic. The fact that these correspond to thresholds of appearance of certain properties is not fully understood in all generality, and lies very much at the heart of the problem.

The thermodynamic potential which applies to our case is the *free entropy density* [MM07], which is defined as the logarithm of the partition function:

$$\Phi = \frac{1}{N} \log Z(\beta). \tag{1.5}$$

We will compute these quantities in the limit $N \to \infty$, also called the *thermodynamic limit*, and in the low temperature regime $\beta \to \infty$. In the physics literature it is more common to work with the free energy density, $-\frac{1}{\beta}\Phi$, where the $1/\beta$ factor ensures that the quantity has indeed units of energy. Intuitively, the free entropy density is the exponential order of the number of spin configurations of energy 0 (assuming energies are non-negative), while the free energy density is related to the typical value of the energy penalty incurred (i.e. the ratio of unsatisfied clauses for CSPs). We will be more interested in the former than the latter.

1.6.2 Message passing and the cavity method (the replica-symmetric case)

The free entropy is in general hard to compute. However, in the cases where the factor graph is a tree, there is a way to obtain an exact answer by using *message passing*. In certain circumstances, the equations that we derive for trees will work also when the factor graph is only *locally tree-like*, i.e. where the finite-depth neighborhood around a random vertex is w.h.p. a tree. To write down the message passing equations, it is enough to pretend the graph is a tree and then message computation is equivalent to dynamic programming.

Let (i) $\mu^{a\to i}$ and (ii) $\mu^{i\to a}$ be the marginals on the spin i when (i) all the function nodes in $\partial i \setminus a$ are deleted and (ii) the function node a is deleted, respectively. Note that these objects belong to $\Delta(\Omega)$, the set of probability measures on Ω, or the $|\Omega|-1$-dimensional simplex. Using the tree-like assumption, we can write the following relations between the marginals:

$$\mu^{i\to a}(\sigma_i) = \frac{1}{Z^{i\to a}} \prod_{b\in\partial i\setminus a} \mu^{b\to i}(\sigma_i), \qquad Z^{i\to a} = \sum_{\sigma_i} \prod_{b\in\partial i\setminus a} \mu^{b\to i}(\sigma_i), \tag{1.6}$$

$$\mu^{a\to i}(\sigma_i) = \frac{1}{Z^{a\to i}} \sum_{\sigma_{\partial a\setminus i}} \psi_a(\sigma_{\partial a}) \prod_{j\in a\setminus i} \mu^{j\to a}(\sigma_j), \qquad Z^{a\to i} = \sum_{\sigma_{\partial a}} \psi_a(\sigma_{\partial a}) \prod_{j\in a\setminus i} \mu^{j\to a}(\sigma_j)$$

The messages enable us to compute the marginals of the Gibbs distribution at each spin as

$$\mu^i(\sigma_i) = \frac{1}{Z^i} \prod_{b \in \partial i} \mu^{b \to i}(\sigma_i), \qquad Z^i = \sum_{\sigma_i} \prod_{b \in \partial i} \mu^{b \to i}(\sigma_i). \tag{1.7}$$

The free entropy is then (see Appendix A)

$$N\Phi = \sum_i \log \left[\sum_{\sigma_i} \prod_{b \in \partial i} \mu^{b \to i}(\sigma_i) \right] + \tag{1.8}$$

$$+ \sum_a \log \left[\sum_{\sigma_{\partial a}} \psi_a(\sigma_{\partial a}) \prod_{j \in \partial a} \mu^{j \to a}(\sigma_j) \right] - \tag{1.9}$$

$$- \sum_{i \sim a} \log \left[\sum_{\sigma_i} \mu^{a \to i}(\sigma_i) \mu^{i \to a}(\sigma_i) \right]. \tag{1.10}$$

In the case of coloring, since each constraint involves only two vertices, then messages $\mu^{i \to a}$ and $\mu^{a \to j}$ can be both expressed using messages $\mu^{i \to j}$, so we would only keep one type of message. We will derive the simplified equations for coloring in more detail in Chapter 3.

For the case where the factor graph contains cycles, we can still iterate Equations (1.6) and hope that we obtain good approximations to the true marginals and free entropy. In general, the form (1.8), viewed as a function of the messages is called the Bethe functional. Such means of estimating the free energy were already employed in the 1930s by Bethe [Bet35], Onsager [Ons36] and Peierls [Pei36].

This method is referred to as the *cavity method* since at least on a tree, the messages can be described by true marginals when nodes are removed from the graph (creating cavities). On a tree, the incoming messages in a particular node represent probabilities that are independent. If cycles exist, but the graph is still locally tree-like, the method still yields good approximations when there is little correlation (under the Gibbs measure) between spins that are far away from each other in the graph. In that case, the independence of incoming messages (also called "cavity fields" in the physics jargon) is replaced by the weaker assumption of low correlation.

Formalizing the previous statement is in fact a challenging task, which is not yet mathematically settled. When there are no long-range correlations, there is just one solution to the system of message passing equations, and it can be reached by iterating (1.6). After the point where long-range correlations appear, there will be many such solutions, in fact exponentially many. That point is denoted by α_d, the *dynamic threshold*. There are attempts to characterize this threshold using *reconstruction* on random trees [MM06, Sly09]. Beyond the point α_d the long correlations prevent this approach from working. The cavity method can still be used (see below), albeit on a more complicated model. We will refer to the message-passing equations and the free energy approximation that we have seen so far as the *replica-symmetric* approach.

1.6.3 Replica symmetry breaking

The reason why the Replica Symmetric approach fails after the dynamic threshold is that the solutions tend to form *clusters*. In the case of low temperature ($\beta \to \infty$), a cluster corresponds to a set of satisfying spins configurations (i.e. configurations of minimal energy) that are all close to each other and are well separated from the other clusters. We present here just a very broad overview of the method. We will show the derivation in much more detail in Chapter 3, where it will be specialized to coloring.

The ansatz on which the RSB cavity method relies is the fact that these clusters correspond to solutions of the system of message passing equations. Instead of using message passing to study the space of solutions (the RS approach), we can employ message passing to study the distribution of clusters. This happens because the distribution on clusters can be written down also as a tree-like graphical model on which the cavity method can be used. It turns out that this meta-model exhibits decay of long-range correlations beyond the dynamic threshold.

The RSB approach allows us to compute the number of clusters of each size (in the large N limit both the number and the size are characterized by their exponential order). There will typically exist a particular size, and clusters of that size will be dominating, in the sense that one valid configuration picked at random will be part of a cluster of that size. To sample such a cluster at random, clusters are weighted by their size in the RSB distribution on clusters that we mentioned earlier.

The above scheme manages to sample clusters of the right size, in the case where their number is exponential. As the parameter α increases, there will be a threshold where this number ceases to be exponential. In that case, the scheme outline above fails to sample a dominating cluster. The reason why this happens is the following. The model is random, and our way of sampling assumes that the factor graph is chosen randomly. In the sampling method the two types of randomness, the one in the choice of the factor graph, and the one in choosing the cluster at random are in fact mixed together. What we would like to obtain is the size of the dominating cluster in a typical random instance; what we actually do is equivalent to computing the expected value of the cluster size when instances are picked at random. The latter approach is prone to be influenced by rare events: exponentially rarely it happens that an abnormally sized cluster appears, and its size compensates for the rarity. This is not what we want to get, but this defect is built in the method itself: we do not afford to sample a huge typical instance first and then do the computations, rather the quantities we compute are already averaged over the randomness in the instance.

The point α_c where the dominating clusters become sub-exponential in number is called *condensation threshold* or *Kauzmann transition* [Kau48]. At this point, the *complexity*, defined as the exponential order of the number of dominating clusters is 0. For higher values of α, the vanilla RSB cavity method predicts a negative complexity, which is physically impossible, due to the reasons outlined above.

There is, however, a way to overcome this deficiency. This is done by reweighing the clusters in the distribution over clusters. Instead of weighing each cluster by its size, we weigh it by its size to a power m (the Parisi parameter). The vanilla method corresponds to choosing $m = 1$. By varying m, we are able to artificially make clusters of other sizes dominant, sample from among them, and so compute their numbers. Beyond the condensation threshold, while the instances are still solvable, it turns out that the "right" value $m^\star$ is something in between 1 and 0.

The value $m = 0$ is also an interesting case. It allows us to weigh all the clusters in the same way, regardless of their size. If the complexity in this case is positive, it means that there are still clusters of solutions. If it is negative, it means clusters of any sort appear only exponentially rarely, so we must be already in the unsolvable region. Thus running the RSB cavity method at $m = 0$ enables us to compute the solvability threshold α_s (further referred to as the *colorability threshold* or *satisfiability threshold* as the case may be). Calculations to determine the values of free energy and complexity simplify greatly for the two values $m = 1$ and $m = 0$.

The formalism at $m = 0$ is usually referred to as *survey propagation* (SP). The simplified message passing can be actually run on real instances in order to estimate marginals. This gives rise to a very effective algorithm, called *SP with guided decimation*. This algorithm works by repeatedly running SP on the graph, selecting the nodes with the highest bias, assigning them the corresponding spin values and then removing from the graph.

Another notion that plays a role in the RSB formalism is *freezing*. We say that a cluster is frozen if a nonzero fraction of variables take the same value under all configurations of spins in that cluster. The *freezing threshold* α_f is defined as the point where freezing starts to occur in all dominating clusters. Freezing is important for multiple reasons: (i) it is believed that freezing represents an algorithmic barrier in search algorithms and (ii) certain proofs only work in a regime where freezing occurs (for example the proof of the condensation transition in Q-COL [BCOH$^+$14]). The location of α_f can be determined directly from the RSB cavity method, and it can occur both below and above α_c [Sem08].

The RSB formalism was developed in the context of the Sherrington-Kirkpatrick (SK) model [Par80, MPV87]. The first application of the RSB cavity method to diluted spin glass models (i.e. models with a sparse and locally tree-like factor graph) was done for the Bethe lattice spin glass [MP01, MP03] and the SAT problem [MPZ02, MZ02]. Similar results were obtained then for coloring [MPWZ02, VMS02, BMP$^+$03, KPW04]. The organization of solutions into clusters and their "geometry" was studied in [MMZ05, MPR05]. The question of stability of the replica symmetric solution was investigated in [MPRT04]. The formalism at $m = 0$, i.e. the SP equations, was developed in [MPZ02, MZ02], and SP-guided decimation was studied in [BZ04, BMZ05, MMW07, ZK07]. There are other algorithms based on guided decimation algorithms using belief propagation (BP). These are typically easier to analyze but perform less well when compared to SP-guided decimation [MRTS07].

In principle, nothing prevents us from hypothesizing the existence of clusters of clusters.

Q	α_d	α_f	α_c	α_s
3	4.00	4.66	4.00	4.69
4	8.35	8.83	8.46	8.90
5	12.84	13.55	13.23	13.67
6	17.64	18.68	18.44	18.88
7	22.70	24.16	24.01	24.45
8	27.95	29.93	29.90	30.33
9	33.45	35.66	36.08	36.49
10	39.01	41.51	42.50	42.93

Table 1.1 – Threshold values for Q-COL [ZK07]. Note that for $Q \leq 8$ we have that $\alpha_f > \alpha_c$.

These would be analyzed by a meta-meta-model. In diluted spin glasses this is conjectured to be unnecessary, since self-consistency checks indicate that one level of RSB is enough. In the case of the Sherrington-Kirkpatrick model, however, it turns out there is an infinite hierarchy of clusters, and a formalism named *full replica-symmetry-breaking* is necessary. Because of the inherent symmetry of the problem (the factor graph is the complete graph), this was historically the first use of the RSB method and also the only case where the full RSB formalism was carried out [Par80]. The approach was made mathematically rigorous in a breakthrough result by Talagrand [Tal03].

As mentioned earlier, the RSB formalism is not (yet) fully rigorous. In many situations, however, proofs were obtained which confirm the location of thresholds. XORSAT is a problem that is not genuinely hard, since solutions can always be found by linear algebra. However, it exhibits clustering [AM13], with the clusters forming linear spaces that are isomorphic to each other. This allows the rigorous determination of the dynamic and satisfiability thresholds[10]. As we will soon see, this was also achieved in the case of LDPC codes by means of spatial coupling.

Significant progress has been made on two open questions very recently. First, the location of the condensation threshold was determined to coincide with the one predicted by statistical physics in the case of Q-COL with Q large enough [BCOH$^+$14]. Secondly, the location of the satisfiability threshold was fixed for K-SAT for K large enough [DSS14], thereby also closing the theoretical gap left open by Friedgut [FB99] in that regime. Similar results were obtained for the condensation threshold in the case of K-hypergraph-2-coloring [COZ12]. The location of α_f was proved rigorously for large K and Q in the case of Q-COL [Mol12] and NAE-SAT and K-hypergraph 2-coloring [MR13].

[10] The condensation and satisfiability thresholds are the same in this case. This is because all clusters have the same size, so the formalism at any value of m will yield the same results.

K	α_d	α_c	α_s	α_f
3	3.86	3.86	4.267	*
4	9.38	9.55	9.931	9.88
5	19.16	20.80	21.117	*
6	36.53	43.08	43.37	39.87

Table 1.2 – Threshold values for K-SAT [MRTS08]. Note that for $K \leq 5$, the freezing threshold occurs after the condensation threshold. For the evaluation of α_f one must then use the right value of $m^\star$. This has been done only for $K = 4$.

1.7 Planted models

Studying the clusters of a random instance of a CSP is usually a non-trivial task. We can adopt the following strategy to modify a random model of CSP. Draw uniformly at random a configuration of spins and fix it; we will call this the *planted solution*. Then we sample the constraints independently at random as in the original CSP model, but conditioned on the fixed configuration being a solution. This is called a *planted model*. In general, the planted model and the original one are not equivalent, in the following sense: picking an instance from the original CSP model and then a random solution is different from picking a solution at random and then an instance of the planted model that satisfies that solution.

In the case of Q-COL, it was proved that the planted model is equivalent to the original for $\alpha < \alpha_c$ [BCOH$^+$14]. For $\alpha > \alpha_c$, the two start to differ. In particular for $\alpha > \alpha_s$, the planted model is still always solvable, by construction, while the original model is not. The planted model presents interest also on its own, of a cryptographic flavor, in that one can hide the planted solution and see if there are phases where the solution is unique but hard to find [KZ09].

1.8 Error correcting codes

1.8.1 Channels

We are sending a bit vector $\underline{X}$ of length N over a noisy channel. The output of the channel is a vector $\underline{Y}$ of the same length, a corrupted form of $\underline{X}$. Even though in general a channel is described by any conditional probability distribution $p_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x})$, the channels we consider have the following properties:

- *binary-input*: the entries of $\underline{X}$ come from a binary alphabet; purely for notational convenience we assume this alphabet is $\{+1, -1\}$;
- *memoryless*: conditioned on the vector $\underline{X}$, the random entries of the output $\underline{Y}$ are i.i.d.;

for this reason the channel is described by its action $p_{Y|X}(y|x)$ on a single symbol. We make the convention that whenever X, Y, x, y appear without a bar we refer to a single use of the channel.

- *symmetric*: in the case where the output alphabet consists of real numbers, symmetry means that for all y, x, $p_{Y|X}(y|x) = p_{Y|X}(-y|-x)$. However, we will work with a more general characterization for symmetry, in terms of log-likelihoods, which we introduce soon.

Note that we leave open the nature of the output alphabet and of the distribution $p(y|x)$, which might not be necessarily discrete. We call such channels *BMS* channels. The three most common examples of BMS channels are the following.

- The *binary erasure channel* BEC(ϵ). In this case, the output alphabet is $\{+1, -1, 0\}$. The channel has a parameter ϵ, the erasure probability. With probability ϵ it outputs the erasure symbol 0, otherwise it simply copies the input to the output.

- The *binary symmetric channel* BSC(p). Here the output alphabet is $\{+1, -1\}$. The channel has a parameter p, the flipping probability. It either does nothing to the input ($Y = X$) or flips it $Y = -X$, with probabilities $1 - p$ and p, respectively.

- The *binary additive white gaussian noise channel* BAWGNC(σ). Here the output alphabet is $\mathbb{R}$. The channel samples a random value Z from a normal distribution $N(0, \sigma^2)$, where standard deviation σ is the channel parameter. It then outputs $Y = X + Z$.

1.8.2 Codes and capacity

We want to maximize the information contained in the input $\underline{X}$, while still being able to reconstruct it from the output $\underline{Y}$. To account for the transmission errors, we will use error-correcting codes. A *code* is simply a set $\mathscr{C}$ of input vectors, whose elements are called *codewords*. We will restrict our transmission to codewords, thereby introducing redundancy in the input and transmitting less information. This redundancy will then help us reconstruct the input from the corrupted output.

We assume that $\underline{X}$ has a uniform distribution over the code $\mathscr{C}$.[11] Retrieving $\underline{X}$ from $\underline{Y}$ is a process called decoding, and it is prone to errors, since in general an output vector can correspond to multiple inputs, and the decoder will just choose one of them. Eliminating errors completely is for most channels impossible. However, we can ask that the probability of having errors tend to 0 as $N \to \infty$. In a celebrated result that lies at the foundation of

[11] Here $\underline{X}$ is already encoded. Of course, in real systems there is also the task of converting the useful information (the source bits) into codewords, with which we will not be concerned here. This task is much easier when the code is structured (for example, when it is a linear space). The fact that we require $\underline{X}$ to be uniformly distributed is easily accomplished using lossless compression on the source

information theory, Shannon [Sha48] showed that not only such schemes exist, but he also characterized exactly the amount of information one can send through a channel.

We will measure the information contained in a random variable by its *Shannon* entropy[12],

$$H(\underline{X}) = -\sum_{\underline{x}} p_{\underline{X}}(\underline{x}) \log p_{\underline{X}}(\underline{x}), \tag{1.11}$$

and the information shared by two random variables $\underline{X}$ and $\underline{Y}$ by the *mutual information*

$$I(\underline{X};\underline{Y}) = \sum_{\underline{x}} \sum_{\underline{y}} p_{\underline{X},\underline{Y}}(\underline{x},\underline{y}) \log \frac{p_{\underline{X},\underline{Y}}(\underline{x},\underline{y})}{p_{\underline{X}}(\underline{x}) p_{\underline{Y}}(\underline{y})}. \tag{1.12}$$

The entropy simply measures how many bits of useful information are transmitted. In the case of the uniform distribution on $\mathscr{C}$, this is just $\log|\mathscr{C}|$. The *rate* of the code is defined as the ratio between useful information and the codeword length, $\frac{1}{N}\log|\mathscr{C}|$.

Shannon's Theorem of Channel Coding states that (i) the maximum amount of information we can hope to pack into the input is upper-bounded by the channel capacity

$$C = \sup_{p_{\underline{X}}} I(\underline{X};\underline{Y}), \tag{1.13}$$

a quantity that only depends on the channel, and (ii) the capacity is also achievable, in the sense that there exist coding schemes with the rate arbitrarily close to C and with error probability tending to 0 as $N \to \infty$.

The codes that achieve capacity in Shannon's proof are random codes. That is, the codewords are chosen independently and uniformly at random. This makes random codes impractical, since one would need to store in memory all codewords, and these are exponentially many. This inconvenience can be alleviated by using random codes with a structure, for example ones where the codewords form a linear space. These are still capacity achieving, but for many years is was not clear how to find an efficient algorithm for decoding up to capacity. Moreover, a linear space of codewords in general needs memory space on the order of N^2 just for storage. But all the good properties still hold if we would restrict ourselves to linear codes that are sparse. This is the case for LDPC codes. These have the advantage that they can be stored in linear space and have efficient algorithms for decoding. Moreover, for the spatially coupled variant there are efficient algorithms that decode up to capacity.

In what follows we will only consider LDPC codes, because their structure is similar to that of random CSPs and diluted spin systems. There are, however, other efficient codes that provably achieve capacity, a notable example being polar codes [Ari09].

[12] We use the convention $0\log 0 = 0$ everywhere.

1.8.3 LDPC codes

Low-density parity check (LDPC) codes were first introduced by Gallager [Gal63], but they were not so popular in the beginning, as the computation needed for decoding was considered too high at that time.

The codewords are defined by parity check constraints. These are relations of the form $x_{i_1} x_{i_2} \cdots x_{i_K} = 1$, where $i_1, \ldots, i_K$ are specific to each parity check constraint. It is easy to see that the totality of all parity checks forms a homogeneous system of linear equations over the field with two elements. As such, the codewords (the solutions of this system) form a linear subspace, whose dimension is given by N minus the number of independent check constraints. This dimension equals NR, where R is the rate of the code.

The structure of the code can be represented easily by what is called the Tanner graph. As we will see soon, we will identify this graph with part of the factor graph of the aposteriori distribution used when decoding. This graph is bipartite, with the two types of nodes called *variable* and *check* nodes. The variable nodes correspond to the bits, while the check nodes correspond to the check constraints. We have an edge between a variable and a check node if the corresponding bit takes part in the check constraint.

This structure is sampled at random, in order to obtain a random code, much in the same way as we are choosing the factor graph of CSPs at random. The distribution of the Tanner graph is usually called in the literature an *ensemble*. There is one major difference to the Erdős-Rényi model that we considered for CSPs. There the links of the constraints were chosen independently, so the degree distribution on the variable node side was Poisson. Such a distribution is not usually good for LDPC codes. The rule of thumb is that the more constraints a variable is connected to, the more that it is guarded against errors. If the degree distribution were Poisson, a nonzero fraction of these nodes would participate in no check whatsoever, and any error that were to occur on such a variable node would not be fixed. For this reason, LDPC codes are usually sampled in such a way that the degrees of both variables and checks are fixed; in other words, the ensemble is *regular*. For regular ensembles we need to give up the useful property of the independent sampling of check nodes, which tends to somewhat complicate the picture in our proofs.

In a general setting, we fix two *target degree distributions*, one for the variable nodes and one for the check nodes. Unless otherwise noted, we assume that these two distributions are concentrated on two values for the degrees, K for the check nodes, and d for the variable nodes. If M is the number of check nodes, we have the relation $MK = dN$.

The actual sampling for the regular ensemble is done by using the *configuration model* method, as follows. For each variable node we create d *variable node sockets*, and for each check node K *check node sockets*. We then connect each variable node socket to a check node socket by a random permutation (note that the two types of sockets are equal in number). Note that this does not correspond to picking a Tanner graph with the prescribed degrees uniformly at

random. The latter is a slightly different ensemble, from which it is harder to sample.

1.8.4 MAP Decoding and the Gibbs measure

The probabilistic model of channel transmission allows us to compute the aposteriori probability of each codeword being sent. This is obtained by using Bayes' rule:

$$p_{\underline{X}|\underline{Y}}(\underline{x}|\underline{y}) = \frac{1}{p_{\underline{Y}}(\underline{y})} p_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x}) p_{\underline{X}}(\underline{x}). \tag{1.14}$$

MAP (maximum a posteriori) decoding is simply picking the codeword $\underline{X}$ with the highest $p_{\underline{X}|\underline{Y}}(\underline{x}|\underline{y})$. This is computationally hard in general, but

We will treat the expression above as a Gibbs measure. It already factorizes nicely, but we will transform it slightly so that the dependence on $\underline{x}$ becomes more explicit. For this we introduce the half-log-likelihood-ratios (HLLR) $h(y)$, defined as

$$h(y) = \frac{1}{2} \log \frac{p_{Y|X}(y|+1)}{p_{Y|X}(y|-1)}, \tag{1.15}$$

with the possibility of it taking infinite values. From $h(y)$ one can recover the posterior probability that the bit x was sent. The latter is easily seen to be proportional to $e^{h(y)x}$. In fact we have

$$\frac{p_{Y|X}(y|x)}{p_{Y|X}(y|1)} = e^{h(y)(x-1)}. \tag{1.16}$$

For symmetric channels, the HLLR are a sufficient statistic, meaning that any reasoning we do based on the posterior probability can actually be done by knowing just $h(Y)$ and not Y itself. This can be easily seen when we rewrite (1.14) as (1.18) below. We get the prior on $\underline{X}$ using the graph description:

$$p_{\underline{X}}(\underline{x}) = \frac{1}{|\mathscr{C}(G)|} \mathbb{1}(\underline{x} \in \mathscr{C}(G)) = \frac{1}{|\mathscr{C}(G)|} \prod_{a \in G} \frac{1}{2} \left(1 + \prod_{i \in \partial a} x_i\right). \tag{1.17}$$

One can easily check that the product $\prod_{a \in G}(1 + \prod_{i \in \partial a} x_i)/2$ is 1 when $\underline{\sigma}$ is any codeword, and 0 otherwise. Putting everything together we obtain

$$p_{\underline{X}|\underline{Y}}(\underline{x}|\underline{y}) = \frac{1}{p_{\underline{Y}}(\underline{y})|\mathscr{C}(G)|} e^{\sum_i h(y_i)(x_i-1)} p_{\underline{Y}|\underline{X}}(\underline{y}|\underline{1}) \prod_{a \in G} \frac{1}{2} \left(1 + \prod_{i \in \partial a} x_i\right). \tag{1.18}$$

The plan is to get rid of $\underline{Y}$ completely in (1.18) and use just the HLLR. We use this opportunity to also change notation to the one used by physicists. The posterior will be given by the Gibbs measure μ depending on the vector of HLLR h. For conformity, bits will from now on use spin

notation σ instead of x (and still take values ± 1). Furthermore, we introduce the shorthand $\sigma_a = \prod_{i \in \partial a} \sigma_i$, as such products (arising from the parity checks) will be very common. We have

$$\mu(\underline{\sigma}) = \frac{e^{\underline{h} \cdot (\underline{\sigma} - \underline{1})} \prod_{a \in G} (1 + \sigma_a)/2}{Z}, \tag{1.19}$$

where $\cdot$ signifies the scalar product between vectors and the *partition function* Z is given by

$$Z = \sum_{\underline{\sigma} \in \mathscr{X}^V} e^{\underline{h} \cdot (\underline{\sigma} - \underline{1})} \prod_{a \in G} \frac{1 + \sigma_a}{2}.$$

Note that the scaling provided by shifting $\underline{\sigma}$ by 1 downward helps to keep the weights involved finite in the case $h = +\infty$. We will see shortly that the case $h = -\infty$ will never occur in our calculations, since by symmetry we can assume the codeword sent is the all-+1 codeword.

We have denoted the above probability measure by μ in order to distinguish it from other randomized parameters that appear, notably the channel and the randomness in the graph G. Note that μ depends on both G and the HLLRs $\underline{h}$, and when this is not clear we will make it explicit by adding G or $\underline{h}$ as a subscript: $\mu_{G,\underline{h}}$, $Z(G, \underline{h})$. Note that the Gibbs measure is a random quantity, as it depends on the channel and the random code.

The average with respect to the measure μ will appear quite often in the rest of the paper, and we use the *Gibbs brackets* $\langle \cdot \rangle$ to indicate it. In other words,

$$\left\langle f(\underline{\sigma}) \right\rangle = \sum_{\underline{\sigma} \in \mathscr{X}^V} f(\underline{\sigma}) \mu(\underline{\sigma}).$$

Regarding notation, the same subscript conventions, as for μ, apply for the bracket.

There are three types of randomness that are involved in our construction: (i) the random graph which is picked from an LDPC ensemble; (ii) the randomness induced by the channel and (iii) the Gibbs measure. The expectation in the first case is denoted by $\mathbb{E}_{G:\mathscr{G}}[\cdot]$, where $\mathscr{G}$ denotes the ensemble. The expectation with respect to the channel is written as $\mathbb{E}_h[\cdot] = \int \cdot \, d\mathsf{c}(h)$. As seen before, the average with respect to the Gibbs measure is denoted by angular brackets. The symbols $\mathbb{E}_{G:\mathscr{G}}$ and $\mathbb{E}_h$ commute, since the graph and the channel are independent. The angular bracket, however, depends on both h and the graph G and thus does not commute with the $\mathbb{E}$ symbols. In the language of Statistical Physics, the graph and the channel are said to be quenched.

Because of symmetry, the channel is fully characterized by the distribution $\mathsf{c}(h)$ of the HLLR computed from the output of the channel by (1.15) assuming the input of the channel is set to +1. We will view this distribution as a measure c on $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$, which due to channel symmetry has the property

$$d\mathsf{c}(-h) = d\mathsf{c}(h) e^{-2h},$$

i.e., a HLLR h is e^{2h} times more likely to occur than its negative. For this reason we call this property *symmetry of measures*, we denote all symmetric measures on $\overline{\mathbb{R}}$ by $\mathscr{X}$ and we identify $\mathscr{X}$ with the set of BMS channels. As a side note, observe that the mass sitting at $-\infty$ in any symmetric c must be 0. Because the HLLR is a sufficient statistic, all channels that share the same HLLR distribution are all equivalent, in the sense that the statistics of the posterior are the same. For this reason we can and will assume without loss of generality that the HLLR itself is the output of the channel.

A fundamental role will be played by the conditional entropy $H(\underline{X}|\underline{Y})$, defined as $-\mathbb{E}_{\underline{X},\underline{Y}} \log p(\underline{X}|\underline{Y})$. This measures the uncertainty in the input given the observation of the channel output, and thus it characterizes the ability to decode. In our notation inspired from statistical physics, the conditional entropy is in fact equal to the partition function, as expressed by the following lemma. We will still use the notation $H(\underline{X},\underline{Y})$ to convey information-theoretic intuition when needed.

Lemma 1. *For a linear binary code of block length N represented by a graph G, we have*

$$H(\underline{X}|\underline{Y}) = \mathbb{E}_{\underline{h}} \log Z(G,\underline{h}).$$

Proof. We use successively: (a) the definition of entropy, (b) the fact that a priori all codewords are equally likely to be sent and the symmetry of the channel, which ensures that all terms in the sum are identical, (c) the fact that the log-likelihood is a sufficient statistic, so $p(\underline{\sigma}|\underline{y}) = p(\underline{\sigma}|\underline{h})$, and the latter is nothing else than the probability measure $\mu_{G,\underline{h}}$, and the fact that the distribution of the HLLR is given by the distribution c and (d) the fact that $\mu(\underline{1}) = Z^{-1}$:

$$\begin{aligned} H(\underline{X}|\underline{Y}) &\overset{(a)}{=} -\sum_{\underline{\sigma}\in\mathscr{C}(G)} p(\underline{\sigma}) \int \mathrm{d}\underline{y} \prod_i p_{Y|X}(y_i|\sigma_i) \log p_{\underline{X}|\underline{Y}}(\underline{\sigma}|\underline{y}) \\ &\overset{(b)}{=} -\int\int \mathrm{d}\underline{y} \prod_i p_{Y|X}(y_i|1) \log p_{\underline{X}|\underline{Y}}(\underline{1}|\underline{y}) \\ &\overset{(c)}{=} -\int\int \prod_i \mathrm{dc}(h_i) \log \mu_{G,\underline{h}}(\underline{1}) \\ &\overset{(d)}{=} \mathbb{E}_{\underline{h}} \log Z(G,\underline{h}), \end{aligned}$$

where $\mathscr{C}(G)$ is the set of codewords. □

1.8.5 Smooth families of channels and thresholds

There is a partial ordering, called *degradation*, defined on $\mathscr{X}$ which expresses the fact that one channel is better or worse with respect to another one. We say that a channel c_1 is degraded w.r.t. a channel c_2 and write $\mathsf{c}_1 \succ \mathsf{c}_2$ if there exists a third channel that can transform the output of c_2 (the better channel) into the output of c_1 (the worse channel).

In the case of random CSPs the parameter of the problem is the clause-to-spin ratio α, and

with respect to this we investigate the occurrence of phase transitions. In the case of LDPC codes, the equivalent quantity is the rate of the code, but unlike random CSPs, we keep this quantity fixed (through the values of d and K). The parameter that we vary in the coding case is given by the channel HLLR distribution c. In contrast to CSPs, this parameter is not one-dimensional, but infinitely dimensional, since it lives in the space $\mathscr{X}$. In this context, it is not a priori clear how one should define thresholds. One could think of them as surfaces in $\mathscr{X}$ that separate easy regions from hard regions. This view is not so easy to formalize. The view we take here is to fix a path in the space $\mathscr{X}$ that fulfills certain properties, one of which being degradation along the path, and investigate where (and if) thresholds occur on that path. Thus the parameter of the problem is again one-dimensional, and a good choice for this parameter is the entropy of the channel $H(\mathsf{c})$ (which for BMS channels is just 1 minus the capacity) as we will see below.

If we turn around the hard upper bound on the rate coming from Shannon's theorem of channel coding, we get that for all channels with capacity lower than the fixed rate, decoding is hopeless. Thus the rate itself is a theoretic threshold of hardness (let us call it *Shannon threshold*), regardless of the choice of code and the exact type of channel under consideration. This means that by choosing as parameter of the path a quantity directly related to the capacity of the channel, not only we use a "universal" parameterization, but also the Shannon threshold occurs at the same place. For this reason we choose to parameterize the channel by the linear functional

$$H(\mathsf{c}) = \int \log_2(1+e^{-2h})\,\mathrm{d}\mathsf{c}(h), \tag{1.20}$$

which has the property that it is monotone with respect to degradation, i.e. $H(\mathsf{c}) > H(\mathsf{c}')$ for $\mathsf{c} \succ \mathsf{c}'$. It can be easily checked that this is actually the conditional entropy $H(X|Y)$ for any symmetric channel characterized by the HLLR distribution c. The capacity of any of those channels is given by $1 - H(\mathsf{c})$. We illustrate this picture with an example.

Example 2. *A common LDPC code is the* $(3,6)$*-code, i.e. the one where* $d = 3$ *and* $K = 6$*. The rate of this code is* $1 - d/K = 1/2$*, and the best possible codes with this rate will be able to work on any channels with capacity as low as* $1/2$*, or, equivalently, channel entropy at most* $1/2$*. Now the* $(3,6)$*-code in particular will not perform that well. If we use the optimal decoder (which is still computationally expensive), we find out we can only decode up to a lower value of the channel entropy, the MAP threshold, and this value is channel-dependent. For example, on the BEC, this value is around* 0.488*, while for BSC and BAWGNC it is slightly smaller. However, as it will follow from the results of Chapter 2, it is true that as the degrees* d, K *increase, the MAP threshold approaches the Shannon threshold of* $1/2$*.*

We denote the parameterization by h, so a path through the space $\mathscr{X}$ is expressed by a *family of channels* $\{\mathsf{c}_\mathrm{h}\}$ with $\mathrm{h} \in [\underline{\mathrm{h}}, \overline{\mathrm{h}}]$ with the property that $H(\mathsf{c}_\mathrm{h}) = \mathrm{h}$. The families of channels considered will can have the following properties:

- *smoothness*: for all continuously differentiable functions $f : \overline{\mathbb{R}} \to \mathbb{R}$ such that $f(h)e^h$ is bounded, the expectation $\int f(h)\,\mathrm{dc_h}(h)$ exists and is continuously differentiable with respect to h in $[\underline{\mathrm{h}}, \overline{\mathrm{h}}]$.
- *ordering by degradation*: $\mathrm{c_h} \succ \mathrm{c_{h'}}$ whenever $\mathrm{h} < \mathrm{h}'$.
- *completeness*: the family is defined for all $\mathrm{h} \in [0,1]$.

Note that since we are concerned only with binary channels, we have $0 \le \mathrm{h} \le 1$ and furthermore $\mathrm{h} = 0$ and $\mathrm{h} = 1$ occur only for the perfect channel, hereafter denoted by Δ_∞, which places all mass at $h = \infty$, and for the useless channel, denoted by Δ_0, which places all mass at $h = 0$. The author apologizes for the notation clash between the channel parameter, denoted by an upright h, and the HLLR, denoted by a slanted h.

Example 3. *The BEC(ϵ), BSC(p) and BAWGNC(σ) are all examples of smooth complete families of channels ordered by degradation. The relations between the usual parameters ϵ, p and σ and the natural parameter* h *are given by*

$$\begin{aligned}
&\mathrm{h}(\epsilon) = \epsilon, && \text{for } \epsilon \in [0,1],\\
&\mathrm{h}(p) = -p\log_2 p - (1-p)\log_2(1-p), && \text{for } p \in [0,1/2],\\
&\mathrm{h}(\sigma) = \int_{-1}^{1} \frac{\sigma}{\sqrt{2\pi}(1-y^2)} e^{-\frac{(1-\sigma^2\tanh^{-1}(y))^2}{2\sigma^2}} \log_2(1+y), && \text{for } \sigma \in [0,+\infty].
\end{aligned}$$

At this point we are able to define the location of the MAP threshold as follows. Given a smooth family of channels ordered by degradation and parameterized by h in the whole interval $[0,1]$, there exists a value $\mathrm{h_{MAP}}$ (called the *MAP threshold*) such that for channel parameters below this value, the scaled average conditional entropy (in other words $\frac{1}{N}\mathbb{E}_{G,h}\log Z(G,h)$) converges to zero in the infinite block length limit, while above this value it is positive. Formally,

$$\mathrm{h_{MAP}} = \inf\left\{\mathrm{h} : \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{G:\mathrm{LDPC}(N,\Lambda,K)} H(\underline{X}|\underline{Y}) > 0\right\}.$$

1.8.6 Belief Propagation and the Bethe Approximation

Belief propagation is the name given to the message passing equations (1.6) that we presented for the RS approach, when used to approximate the posterior distribution (1.19). This name stems from the interpretation of messages $\mu^{a\to i}$ and $\mu^{i\to a}$ as describing beliefs about the true value of σ_i. In (1.19) there are two types of factors: the ones arising from the channel observations, of the form $e^{h_i(\sigma_i-1)}$ and those arising from the parity check constraints, of the form $1+\prod_{i\in\partial a}\sigma_i$. This would imply that one would in principle have to deal with two types of function nodes (observation nodes and check nodes), and more types of messages. However, simplifications can be made: the messages from the channel observation nodes to the variable nodes are constant (in the sense that they do not depend on other messages), while those that travel from variable nodes to the observation nodes are irrelevant.

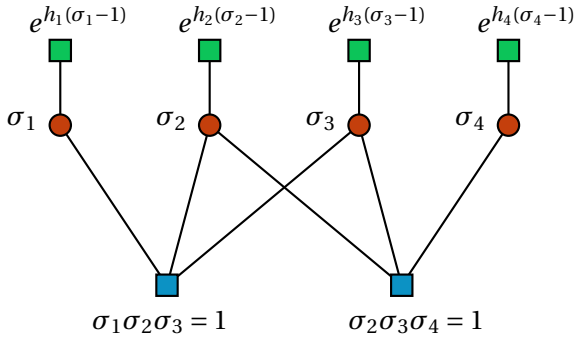


Figure 1.1 – Example of a factor graph of an LDPC code. The observation nodes are depicted green and the check nodes blue.

Also, since the messages represent distributions on a set of cardinality two, they can be characterized by one number. Thus, instead of $\mu^{i\to a}(+1)$ and $\mu^{i\to a}(-1)$ we will use $\nu^{i\to a} = \frac{1}{2}\log\frac{\mu^{i\to a}(+1)}{\mu^{i\to a}(-1)}$ and likewise $\nu^{i\to a}$. Thus the messages take the form of HLLRs, which will enable us to write the equations in a compact form. We have (see Appendix B.1)

$$\begin{aligned}\nu^{i\to a} &= h_i + \sum_{b\in\partial i\setminus a} \nu^{b\to i},\\ \nu^{a\to i} &= \tanh^{-1}\left(\prod_{j\in\partial a\setminus i} \nu^{j\to a}\right).\end{aligned} \tag{1.21}$$

The Bethe functional (1.8) corresponds to

$$\begin{aligned}N\Phi_{\mathrm{BP}} = &\sum_i \log\left(1+e^{-2(h_i+\sum_{a\in\partial i}\nu^{a\to i})}\right) - \sum_a \log\left(1+e^{-2\tanh^{-1}\prod_{j\in\partial a}\tanh\nu^{j\to a}}\right) + \\ &+ \sum_{i\sim a}\log(1+e^{-2\nu^{i\to a}}) + \sum_{i\sim a}\log(1+e^{-2(\nu^{i\to a}+\nu^{a\to i})}).\end{aligned} \tag{1.22}$$

While there are many possible ways ("schedules") to update the messages, the way in which equations (1.21) are most often used in practice is to initialize all messages $\nu_0^{a\to i} = \nu_0^{i\to a} = 0$ and at each time step t use the messages $\nu_{t-1}^{\cdot\to\cdot}$ to compute the messages $\nu_t^{\cdot\to\cdot}$.

In the case of the BEC, everything becomes simple: the messages can only take one of two values: either 0 (not yet determined) or $+\infty$ (determined). The processing at the variable nodes becomes a logical disjunction while at the check node it becomes a disjunction. The schedule becomes unimportant in the case of the BEC and belief propagation reduces to the *peeling decoder*.

For any degraded family of BMS channels, we define the *BP threshold* (informally) as the channel parameter h_{BP} up to which running the BP equations will result in decoding, i.e. all messages will take value $+\infty$ asymptotically almost surely (in the large N limit, for random factor graph and channel realization). Typically the BP threshold is much lower than the MAP threshold. The main feature of spatially coupled codes, as we will see soon in more detail, is

that the BP threshold moves to the value of the MAP threshold, thereby enabling us to decode in an efficient manner using just BP.

1.8.7 Density evolution

Density evolution is used to analyze the belief propagation equations (1.21) in the large N limit and on a code chosen at random. Then we can assume that a message picked at random at time t, going from variable nodes to check nodes comes from a distribution x, while one going from check nodes to variable nodes comes from a distribution y. It can be easily seen that both x and y are symmetric distributions, so $\mathsf{x}, \mathsf{y} \in \mathscr{X}$. In this context, we will refer to the objects from $\mathscr{X}$ as *densities*.

To reflect the types of operations seen in the belief propagation equations (1.21), we introduce two operations on $\mathscr{X}$ denoted by $\circledast$ and $\boxtimes$, defined as follows. The measure $\mathsf{z}_1 \circledast \mathsf{z}_2$ is the distribution of the sum of two independent random variables $h_1 + h_2$ with laws $h_1 \sim \mathsf{z}_1$ and $h_2 \sim \mathsf{z}_2$, respectively; in fact it is just the usual convolution

$$(\mathsf{z}_1 \circledast \mathsf{z}_2)(B) = \int \mathrm{d}z_1(h_1)\, \mathrm{d}z_2(h_2)\, \mathbb{1}[h_1 + h_2 \in B],$$

for any measurable set B. Likewise, we define the measure $\mathsf{z}_1 \boxtimes \mathsf{z}_2$ as the distribution of $\tanh^{-1}(\tanh h_1 \tanh h_2)$, where $h_1 \sim \mathsf{z}_1$ and $h_2 \sim \mathsf{z}_2$ are independent random variables, i.e.

$$(\mathsf{z}_1 \boxtimes \mathsf{z}_2)(B) = \int \mathrm{d}z_1(h_1)\, \mathrm{d}z_2(h_2)\, \mathbb{1}[\tanh^{-1}(\tanh h_1 \tanh h_2) \in B].$$

It can be easily seen that both $\mathsf{z}_1 \circledast \mathsf{z}_2$ and $\mathsf{z}_1 \boxtimes \mathsf{z}_2$ are symmetric measures. In fact, the operations can be generalized straightforwardly to apply to any symmetric finite signed measures, not just probability measures.

These two operations are, each taken separately, commutative and associative. Moreover, when combined with the addition of measures (in the finite signed measure setting) each of them turns the space of symmetric finite measures into a unital algebra over the reals. The distribution Δ_0 serves as a unit for $\circledast$, while Δ_∞ is a unit for $\boxtimes$. However, the two operations $\circledast$ and $\boxtimes$ do not "mix" well among themselves, so the two quantities $z_1 \circledast (z_2 \boxtimes z_3)$ and $(z_1 \circledast z_2) \boxtimes z_3$ are in principle different.

Using these two operations, and assuming incoming messages for a random node are independent, the two BP equations become

$$\mathsf{x} = \mathsf{c} \circledast \underbrace{\mathsf{y} \circledast \cdots \circledast \mathsf{y}}_{d \text{ times}}, \tag{1.23}$$

$$\mathsf{y} = \underbrace{\mathsf{x} \boxtimes \cdots \boxtimes \mathsf{x}}_{K \text{ times}}. \tag{1.24}$$

To express products as above in a more compact form, we introduce the notation $z^{\circledast n} \equiv z \circledast \cdots \circledast z$ where z appears n times. More generally, given a polynomial $\lambda(u) = \sum_{n=0}^{\deg\lambda} \lambda_n u^n$, we define $\lambda^{\circledast}(z)$ as $\sum_{n=0}^{\deg\lambda} \lambda_n z^{\circledast n}$. Note that if $z \in \mathscr{X}$ and λ has positive coefficients with $\lambda(1) = 1$ then also $\lambda^{\circledast}(z) \in \mathscr{X}$. The definitions of $z^{\boxtimes n}$ and $\lambda^{\boxtimes}(z)$ are similar.

By replacing averaging over nodes and edges with averaging over the distributions x and y, the Bethe functional 1.22 becomes

$$\Phi_{\mathrm{BP}}(c,x,y) = H(c \circledast y^{\circledast d}) - \frac{d}{K} H(x^{\boxtimes K}) + d\,H(x) - d\,H(x \circledast y). \tag{1.25}$$

Using the duality formula (B.3) (see Appendix B.2 for the derivation), and (1.24), we can express the Bethe functional as

$$\Phi_{\mathrm{BP}}(c,x) = H(c \circledast (x^{\boxtimes(K-1)})^{\circledast d}) + (d + \frac{d}{K}) H(x^{\boxtimes K}) - d\,H((x)^{\boxtimes(K-1)}). \tag{1.26}$$

Iterating the density evolution equations (1.23) and (1.24) starting from $x = \Delta_0$ (no information about bits) will lead to a fixpoint $x = c \circledast (x^{\boxtimes(K-1)})^{\boxtimes(d-1)}$. The fixpoint corresponds to a local minimum of $\Phi_{\mathrm{BP}}(c,x)$. Decoding using BP is successful if the fixpoint reached is $x = \Delta_\infty$.

In Chapter 2, one of the main results is to show that for standard LDPC codes in the regime $h \in [0, h_{\mathrm{BP}}] \cup [h_{\mathrm{MAP}}, 1]$, the actual free entropy $\mathbb{E}[\Phi]$ matches $\Phi_{\mathrm{BP}}(c,x)$

1.9 Spatial coupling

Spatially coupled LDPC codes have the property that BP decoding works all the way up to the MAP threshold, which is the same as for standard LDPC codes. This phenomenon, the BP threshold moving to the MAP threshold, goes under the name of *threshold saturation*. One characteristic of threshold saturation in the coding case (and compressed sensing, for instance) is that the BP threshold is algorithmic in nature. In other types of models, such as in random CSPs, threshold saturation occurs for quantities that are not obviously algorithmic in nature, as we will see in Chapter 3.

In this section we will review the spatial coupling paradigm in general, and then briefly illustrate the occurrence of threshold saturation in the case of coding over the binary erasure channel.

1.9.1 The spatial coupling paradigm

We present first the general picture, in terms of factor graphs. Coupling introduces an extra "spatial" dimension, whereby the nodes of the factor graph are assigned a *position*, which is

typically an integer. Edges are allowed between two nodes if their positions are spatially close to each other. An example is presented in Figure 1.2.

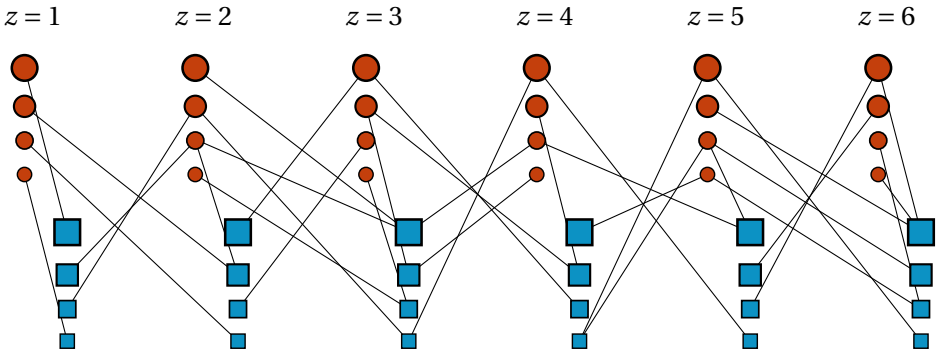


Figure 1.2 – An example of a factor graph whose structure is "spatially" constrained. Note that function nodes are only connected to variable nodes lying at either the same position or one position to the left or to the right. Positions are indexed from 1 to 6.

The whole construction has two parameters, which we call L and W. L is the *length of the chain*, i.e. the number of positions, which are typically indexed from 1 to L. The parameter W is the *window size*, which characterizes the allowed offset between the positions of neighboring nodes. In the example of Figure 1.2, the allowed edges were from a function node at position z to variable nodes at positions $z-1$, z, and $z+1$, which corresponds to a window size of $W = 3$. In what follows, we will assume without loss of generality that an edge links a function node at z with a variable node at a position in $\{z, z+1, \dots, z+w-1\}$.

The random generation of a spatially coupled graph proceeds in a similar fashion as for the standard version. For convenience, we distinguish here two cases:

- *Poisson-distributed degrees* for the variable nodes. Assuming the average variable-node degree is α, we allot for each position a number N of variable nodes and $N\alpha/K$ function nodes. For each function node at position z, we sample the links to the variable nodes independently, choosing uniformly at random among the NW variable nodes at positions $z, \dots, z+W-1$.
- *Arbitrary degree distributions* for the variable nodes, including regular graphs. These can be obtained using the configuration method mentioned in the previous section on coding. There is not a single obvious way in which this can be achieved, and the exact details in the coding case will be presented in the next chapter. The main idea is that we associate a number of sockets for each variable and function node, which corresponds to a target degree. Then we pick a random matching between variable-node sockets and function-node sockets, in such a way that the spatial (windowing) constraints are not violated. While choosing this matching sometimes we may tolerate a number of unmatched sockets as long as this number is sub-linear in N.

We left open the question of what happens at the ends of the chain, since this warrants

additional discussion. The power of spatially coupled structures comes from phenomena that happen at the boundaries. If we were to consider an infinite chain, the neighborhood of a node would not differ at all from the neighborhood of nodes in the uncoupled structure. If we choose to terminate the chain, we will typically obtain modified degree distributions at the boundaries, in a fashion that will be exemplified below for LDPC codes. These modified degree distributions will typically make the problem much easier at the boundaries, allowing for the computation of good marginals, i.e. stronger beliefs, which in turn propagate towards the center of the chain.

The way in which the problem is made easier at the boundary may depend on the model. However, the general recipe is the following. We start by sampling an infinite-length chain, with positions ranging over all the integers. We keep only the variables at positions $1,\dots,L$, together with the function nodes that link to them, regardless of their positions. This means that function nodes will exist also at negative positions, or positions beyond L. These may, in turn, link to variables that are outside positions $1,\dots,L$, let us call them *pseudo-variables*. In the case of LDPC codes, we set the pseudo-variables to $+1$. In the case of coloring or K-SAT, we should set the pseudo-variables to some "special" value which satisfies automatically each constraint that it takes part in; this is equivalent to deleting outright the constraints that involve pseudo-variables.

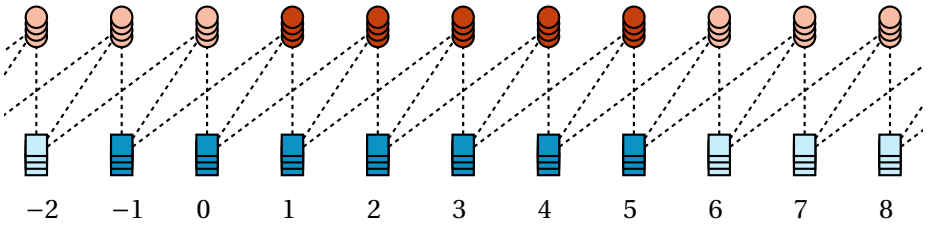


Figure 1.3 – A schematic illustration of the construction of a spatially coupled random model with $L = 5$ and $W = 3$. Possible edge locations are shown with dashed lines, and the pseudo-nodes are shown with light colors.

Of course, message passing takes the same form (1.6) on coupled and uncoupled structures, since coupling only affects the random model that generates the structures. However, if we are interested in a density-evolution/population-dynamics type of analysis, we need to keep track of densities at each position. This is because the shapes of the typical neighborhoods depend on the distance to the chain boundary. An example of this is shown below for LDPC codes.

1.9.2 Threshold saturation for LDPC codes

To convey the feeling of how spatial coupling enhances belief propagation, we first write the density evolution equations in the spatially-coupled scenario and then illustrate the operation of the decoder in the case of transmission over the BEC. At each position z we will keep track of densities $\mathrm{x}_z^{(t)}$ and $\mathrm{y}_z^{(t)}$ at each time step t. We set all the pseudo-variables to $+1$, so $\mathrm{x}_z^{(t)} = \Delta_\infty$ for all $z \le 0$ or $z > L$.

The density evolution equations then take the form

$$\mathsf{x}_z^{(t+1)} = \mathsf{c} \circledast \left(\sum_{w=0}^{W-1} \mathsf{y}_{z-w} \right)^{\circledast d}, \text{ for } z \in \{1, \dots, L\}, \tag{1.27}$$

$$\mathsf{y}_z^{(t+1)} = \left(\sum_{w=0}^{W-1} \mathsf{x}_{z+w} \right)^{\boxtimes K}, \text{ for all } z. \tag{1.28}$$

Check nodes close to the boundary provide more information to their neighbors than do check nodes in the interior of the chain. This is because the pseudo-variables that appear are fixed to +1 and so the check constraints are effectively smaller. This extra information provides an extra edge that allows for the determination of variables close to the boundary. These values in turn help decode bits further inside the chain, creating a "decoding wave". This phenomenon is illustrated for the BEC in Figure 1.4, where the first 120 iterations of density evolution are shown for three different values of the channel parameter.

Note that in the coupled scenario, the rate of the code is smaller than in the uncoupled case. This happens because instead of a d/K check-to-variable ratio, coupled LDPC codes have around $\frac{L+W-1}{L}\frac{d}{K}$ checks per variable. The extra amount of check nodes that lies at the boundary is crucial in providing the seed that gets the decoding wave started. Because of the rate penalty, however, LDPC codes become effective in the limit where $L/W \to \infty$.

1.9.3 Historical note

The first example of spatially coupled codes was introduced by Felstrom and Zigangirov [FZ99] under the name of convolutional LDPC codes. However, the threshold saturation property was not obvious, since the chain considered was circular and so had no boundary. Only later it was observed that terminating the chain dramatically improves the performance [SLCJZ04]. For the BEC, threshold saturation was observed and proved in [KRU11]. Independently, the BP threshold for the coupled codes was computed in [LSCJZ10]; the observation that this in fact coincides with the MAP threshold was subsequently presented in [LF10], where the authors give credit for the observation to G. Liva. The generalized result applicable not just for the BEC, but for any smooth family of BMS channels was presented in [KRU12].

More generally, spatial coupling can be used as a paradigm to build graphical models on which belief-propagation algorithms perform essentially optimally. As such, it has found application not just in coding, but also in the field of compressed sensing [DJM13], where the underlying factor graph is complete, and the algorithm used is Approximate Message Passing (AMP).

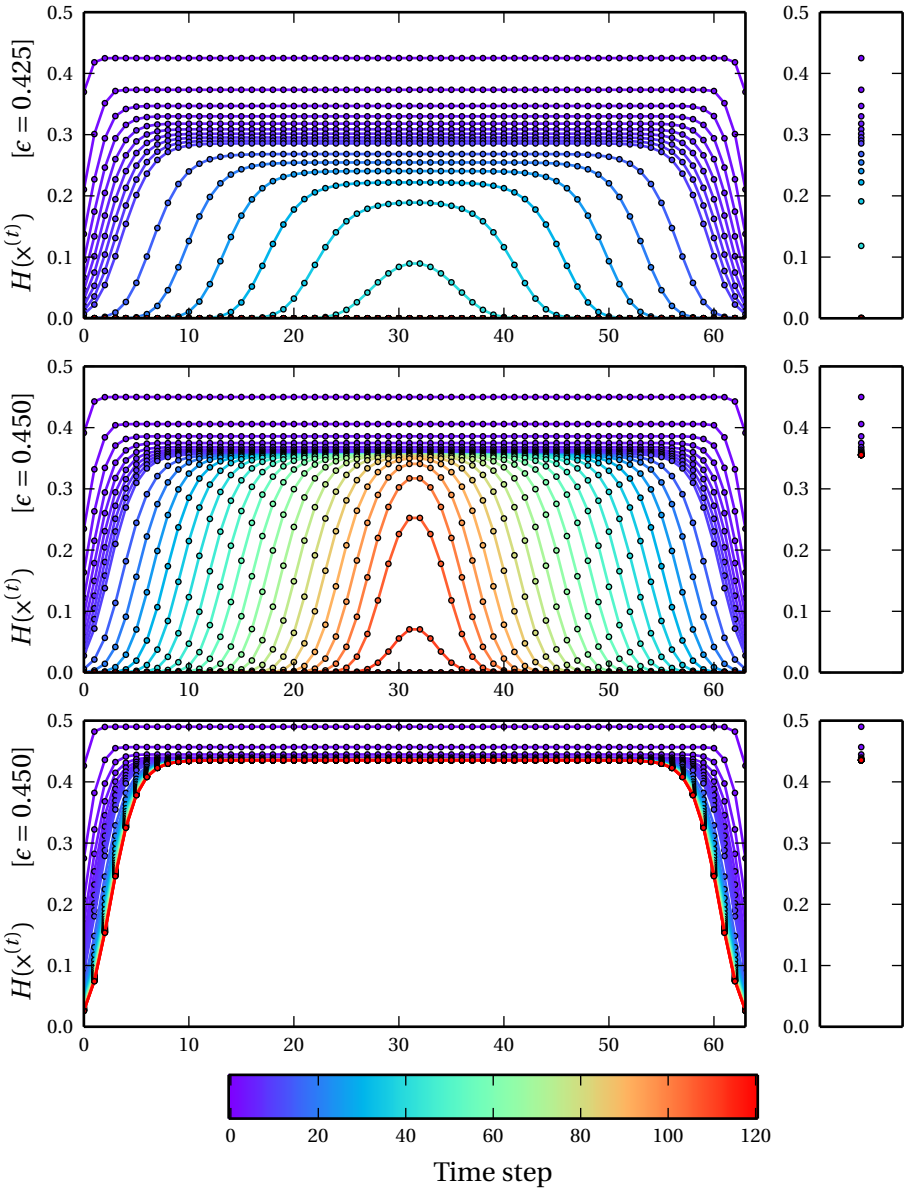


Figure 1.4 – We present here the results of density evolution on a spatially coupled LDPC code with $d = 3$, $K = 6$, $L = 64$ and $W = 4$. DE is run for three values of the erasure probability, for which qualitatively different behaviour of DE is observed. The plots on the left correspond to the coupled ensemble. The value of $H(x_z)$, i.e. the probability that a message leaving a variable node encodes uncertainty is shown for each position z. The plots on the right correspond to same quantity for the standard (uncoupled). DE was iterated 120 times, with time encoded as color, progressing from blue to red.
(i) The top corresponds to the regime $\epsilon < \epsilon_{BP}$. Here both the standard and the coupled codes are able to decode fast.
(ii) The middle corresponds to $\epsilon_{BP} < \epsilon < \epsilon_{MAP}$. Here the standard code gets stuck, but the coupled code manages to decode using the information at the boundary. Note the decoding wave propagating towards the interior of the chain.
(iii) The bottom corresponds to $\epsilon_{MAP} < \epsilon$. In this regime neither of the two codes can decode.

2 LDPC codes achieve capacity: Spatial coupling as a proof technique

The main use of spatial coupling so far was to produce better codes. We will show here how spatial coupling can also become useful in a different way: as a theoretical tool that improves understanding of uncoupled systems. More specifically, sometimes it is easier to prove that (i) a property of a graphical model holds under spatial coupling than for the uncoupled version. If that is the case, and if (ii) the coupled and the uncoupled scenarios are equivalent with respect to that property, then we obtain a proof that the uncoupled graphical system has the said property.

In this chapter we prove a statement of type (ii) in the case of LDPC codes. [1] Namely, we prove Theorem 4 below which states that the conditional entropy in the infinite blocklength limit is the same for the coupled and uncoupled versions of the code. This enables us to derive the equality of the MAP thresholds for coupled and uncoupled codes (Corollary 5). We then present three applications of this result. The first one - Equation 2.2 - is a proof of the Maxwell construction (see [RU08] Chap 4, Sec. 4.12, p. 257): we already know that this conjecture holds for coupled ensembles [KRU12] (a result of type (i)) and here we deduce that it also holds for the uncoupled systems. Then, using the freshly-proven Maxwell construction conjecture, we derive two more results, namely Theorems 7 and 9. The first one states the equality of the BP and MAP GEXIT curves above the MAP threshold (see conjecture 1 in [MMRU09] and Sec III.B [Mac07] for a related discussion) and the second implies the exactness of the replica-symmetric formula for the conditional entropy (see conjecture 1 in [Mon05] and Sec III.B in [Mac07]). Our treatment is general enough to provide a potential recipe for similar results for many types of graphical models.

[1] The content of this chapter has been submitted for publication in [GMU15], and an arXiv preprint can be obtained. A *proof of concept* was presented at ISIT 2012 [GMU12] for ensembles with Poisson-distributed degrees, whose range of applicability in coding is limited. This is due to the occurrence of nodes of very small degrees in significant proportions, which limits the performance. Subsequently, this technical barrier was removed, which allowed for a wide choice of degree distributions, including regular graphs. However, the restrictions (see [GMU12]) that the check node degrees have to be even and that the channel must be symmetric are still necessary. The core of the proof rests on the interplay of symmetry and evenness. A summary of the proof of the main theorem 4 and the application to the proof of the Maxwell construction appeared in ISIT 2013 [GMU13].

Note that the replica-symmetric formula for error correcting codes on general channels was first derived by non-rigorous methods in the statistical mechanics literature [KMS00, MKSV00, Mon01, FLMRT02]. The Maxwell construction and equality of BP and MAP GEXIT curves can also be informally derived from this formula, which in the statistical physics literature plays the role of a "more primitive" object. Progress towards a proof of this formula (for general channels) was then achieved in the form of a lower bound [Mon05, Mac07, KM09] and proofs were found that work in low/high noise regimes [KM01] or for the special case of the binary erasure channel [MMU08, KKM07].

Our proof uses the interpolation method, which was introduced in statistical physics by Guerra and Toninelli for the Sherrington-Kirkpatrick spin glasses [GT04] and gradually found its way to constraint satisfaction problems [FL03, FLT03, BGT10] and coding theory [Mon05, KM09]. The version we use here employs a discrete interpolation between the coupled and two versions of the uncoupled scenarios. An error-tolerating version of the superadditivity lemma is also borrowed from Bayati *et al.* [BGT10] to show that the conditional entropy has a limit for large blocklengths (the equivalent of *thermodynamic limit* in physics terminology).

The rest of this chapter is organized as follows: In section 2.1 we revisit the coupled ensembles and introduce circular coupling. Section 2.2 states the main results and their implications. Next, in Section 2.3 we introduce some prior results, one of which being the Nishimori identity. The main core of the argument resides in Sections 2.4 and 2.5: there we introduce a configuration model that approximates the standard LDPC ensemble and on which we can cleanly perform the interpolation technique. Sections 2.6 and 2.7 are fairly technical. The former describes how to transfer the result from the configuration model to the LDPC ensemble, while in the latter we need to deal with the limit $N \to \infty$.

2.1 Preliminaries

2.1.1 Simple ensembles

We start by describing the simple (i.e. uncoupled) ensemble of codes, which we denote by LDPC(N, Λ, K), where N is the number of variable nodes, $\Lambda(x) = \sum_{d \geq 0} \Lambda_d x^d$ is the probability generating function (PGF) of the variable-node degree distribution, and the integer K is the fixed check-node degree. This is essentially the code ensemble introduced in Section 1.8.3, but generalized to accept a large class of variable-side degree distributions Λ. Previously we considered only the regular case, where Λ is concentrated on one integer d. The distribution Λ must be supported on a finite subset of the positive integers. The average with respect to this distribution will be denoted by $\bar{d}$. For each of the N variable nodes, the *target degree* is drawn i.i.d. from Λ, and each variable node is labeled with that many *sockets*. The purpose of a socket is to receive at most one edge from a check node, and all edges must be connected to sockets on the variable-node side. The number of sockets D will thus be a random variable which concentrates around $N\bar{d}$.

The check nodes and the connections are placed in the following way: As long as there are at least K free sockets (initially all sockets are free), add one new check node connected to K free sockets chosen uniformly at random, without replacement. The chosen sockets then become occupied. The final number of check nodes that are added is exactly $\lfloor D/K \rfloor$. Note that there could be at most $K-1$ unconnected sockets at the end of this process, so the resulting variable node degrees will not in general match the target degrees. However, we will be interested in the limit $N \to \infty$, where the distribution of the resulting degrees matches Λ.

2.1.2 Coupled ensembles

Intuitively, a coupled ensemble LDPC(N, L, W, Λ, K) consists of a number L of copies of a simple ensemble, with interaction between copies allowed, in the sense that a check node can be connected to nodes in neighboring copies. In this chapter it will be more convenient to use a circular chain of positions, as illustrated in Figure 2.1. More precisely, the variable nodes are distributed into L groups, which lie on a *closed circular chain*. The positions are indexed by integers modulo L, and we employ the set of representatives $\{1, \dots, L\}$. Later we will also refer to open-ended chains (i.e. those introduced in Section 1.8.3, and which are actually useful in practice).

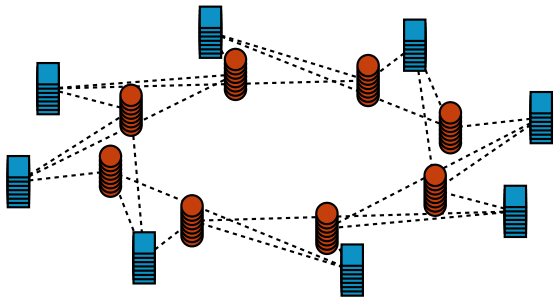


Figure 2.1 – Schematic illustration of the construction of a circular spatially coupled random model with $L = 8$ and $W = 3$. Possible edge locations are shown with dashed lines.

Just as for simple ensembles, each node is assigned a number of sockets drawn i.i.d. from the distribution Λ. The check nodes, however, are restricted in the following way: they are only allowed to connect to sockets whose positions lie inside an interval - called *window* - of length W somewhere on the chain, i.e. there exists a position z such that all edges are connected to nodes at positions $z, z+1, \dots, z+W-1$. As before, check nodes have degree K, and they are sampled as follows: first choose a window uniformly at random, then for each edge, choose a position uniformly and i.i.d. inside that window, and then choose uniformly a free socket at that position. In case there are no free sockets in the chosen position, the process stops. Note that it is possible to stop with a lot of empty sockets in the chain: for example in a very unlucky case, the same position might be picked all the time. However, with high probability, only a small number of sockets will be free at the end of the process, and it is easy to see that in the limit where $N \to \infty$ the rate of the code only depends on $\bar{d}$ and K. The steps in this process will be described in more detail in Section 2.4.

Note that the ensembles described so far are built in two stages: first the vertices are allotted a number of empty sockets, which is determined by sampling from the distribution Λ, thereby establishing the *configuration pattern*; in the second stage, the edges of the graph are connected to free sockets in the configuration pattern. It will be sometimes helpful to separate the two stages and start at the place where the configuration pattern is already given.

This is a good place to observe that the cases where $W = 1$ and $W = L$ yield instances of the single ensemble in the following ways: for $W = 1$, there are L different, non-interacting copies of LDPC(N, Λ, K), whereas for $W = L$, the whole ensemble is equivalent to LDPC(NL, Λ, K), up to $O(\sqrt{N})$ missing check nodes.

The reader will notice that the ensemble we have just constructed is circular and thus the coupling chain has no boundaries. It is a boundary that is responsible for all the useful properties of LDPC codes like threshold saturation. We simply find it easier to work with the circular ensemble and we shall see later that we can add a boundary condition with little cost.

2.1.3 Graphical notation

Traditionally, the Tanner graph is pictured as a bipartite graph, with edges linking the variable nodes to the check nodes. Here we will consider an equivalent rendering, namely as a hypergraph, where the variable nodes are the only nodes, and check nodes correspond to K-ary hyperedges, i.e., K-tuples of variable nodes.

The check constraints have fixed even degree K, and for each check constraint a we denote by $a_1, \ldots, a_K$ the variables involved in the constraint (the ordering is not important, since we are using this notation to describe a single graph). Notation that captures more details will be introduced in Section 2.4 in order to specify exactly the ensemble of codes. For the moment, it suffices to describe a code by listing all of its check constraints, which in turn encode which variables they bind. Thus, abusing a bit the standard terminology, we will say that a graph G is just a K-tuple of check constraints of the kind $a = \{a_1, \ldots, a_K\}$. Note that this notation now allows for repetitions of variables inside check constraints. In general we will use the letters a, b, c, ... to describe check constraints, u, v, ... to describe variable nodes, and G, $\tilde{G}$ G', ... to describe graphs.

2.2 Outline of the results

2.2.1 Comparison of entropies for coupled and simple ensembles

We will set up the machinery of the interpolation method and direct it at proving the following theorem (for the proof, see Section 2.7), which states that the entropies of the simple LDPC(N, Λ, K) and coupled LDPC(N, L, W, Λ, K) ensembles are asymptotically the same in the large N limit.

Theorem 4. *Let L, W, K be integers such that $L \geq W \geq 1$ and K is even and let Λ be a degree distribution with finite support. Then for a fixed BMS channel we have*

$$\lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{G:\mathrm{LDPC}(N,\Lambda,K)} H(\underline{X}|\underline{Y}) = \lim_{N\to\infty} \frac{1}{LN} \mathbb{E}_{G:\mathrm{LDPC}(N,L,W,\Lambda,K)} H(\underline{X}|\underline{Y}), \tag{2.1}$$

and in particular the two limits exist.

Given a smooth family of channels ordered by degradation and parameterized by h in the whole interval $[0,1]$, there exists a value $\mathrm{h}_{\mathrm{MAP}}$ (called the *MAP threshold*) such that for channel parameters below this value, the scaled average conditional entropy (quantities of the kind appearing on both sides of (2.1)) converges to zero in the infinite block length limit, while above this value it is positive.

More formally, for the two kinds of LDPC ensembles, we define the MAP threshold in the following manner:

$$\mathrm{h}_{\mathrm{MAP}} = \inf\left\{\mathrm{h} : \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{G:\mathrm{LDPC}(N,\Lambda,K)} H(\underline{X}|\underline{Y}) > 0\right\},$$
$$\mathrm{h}^{L,W}_{\mathrm{MAP}} = \inf\left\{\mathrm{h} : \lim_{N\to\infty} \frac{1}{NL} \mathbb{E}_{G:\underset{(N,L,W,\Lambda,K)}{\mathrm{LDPC}}} H(\underline{X}|\underline{Y}) > 0\right\}.$$

These definitions usually employ lim inf and are meaningful even when the existence of limits is not guaranteed. However, in our case, the existence of limits is part of the result of Theorem 4. The theorem further implies that these two thresholds are equal.

Corollary 5. *With the same assumptions as in Theorem 4, we have* $\mathrm{h}_{MAP} = \mathrm{h}^{L,W}_{MAP}$.

2.2.2 Proof of the Maxwell construction

As our first application of the equality of MAP thresholds for the coupled and uncoupled ensembles, we will prove the Maxwell conjecture for a large class of degree distributions in the uncoupled case.

Let us recall the statement of the conjecture. The BP-GEXIT function characterizes asymptotically in the large N limit an ensemble of codes over a smooth and degraded family of channels and thus is a function of the channel parameter h (see (2.6) for a definition). Supposing now that h varies from 0 to 1, we define the area threshold $\mathrm{h}_{\mathrm{Area}}$ as that value where the integral of the BP-GEXIT curve over the interval $[\mathrm{h}_{\mathrm{Area}}, 1]$ equals the design rate $1 - \bar{d}/K$. The *Maxwell construction* conjectures that

$$\mathrm{h}_{\mathrm{Area}} = \mathrm{h}_{\mathrm{MAP}}. \tag{2.2}$$

For more details see [RU08] (Chap 4, Sec. 4.12, pp. 257).

The following was recently proved in [KRU12]. For a large class of LDPC ensembles, if we

consider the corresponding coupled ensemble, then the BP threshold (and hence, by threshold saturation, the MAP threshold) is very well approximated by $\mathrm{h}_{\mathrm{Area}}$ (of the simple ensemble) in the following sense:

$$\mathrm{h}_{\mathrm{Area}} - O(\frac{1}{W^{1/2}}) \leq \mathrm{h}_{\mathrm{BP}}^{L,W,\mathrm{open}} \leq \mathrm{h}_{\mathrm{MAP}}^{L,W,\mathrm{open}} \leq \mathrm{h}_{\mathrm{Area}} + O(\frac{W}{L}). \tag{2.3}$$

The threshold $\mathrm{h}_{\mathrm{MAP}}^{L,W,\mathrm{open}}$ is the one of an *open* coupled chain, which is constructed such that the positions on the chain are from $\{1,\dots,L\}$, but the windows do not "wrap around". Instead we add pseudo-variable nodes at positions $-W+2,\dots,-1,0$ and $L+1,\dots,L+W-1$, whose input bits will always be fixed to $+1$. The windows are of the form $\{z,\dots,z+W-1\}$, where $z=-W+2,\dots,L$.

The only difference in the average conditional entropy of the open and closed chains comes from the check nodes that lie at the boundary of the chain. The proportion of these check-nodes is $O(W/L)$. We will later prove in Lemma 12 that the contribution of a single check constraint to the conditional entropy is $O(1)$, and so by a repeated application, the difference of the entropies obtained by removing all check constraints on the boundary is $O(W/L)$, which goes to 0 as $L\to\infty$. As a consequence,

$$\lim_{L\to\infty} \mathrm{h}_{\mathrm{MAP}}^{L,W,\mathrm{open}} = \lim_{L\to\infty} \mathrm{h}_{\mathrm{MAP}}^{L,W}.$$

Thus by (2.3) and Corollary 5, we deduce that in fact $\mathrm{h}_{\mathrm{MAP}}$ equals $\mathrm{h}_{\mathrm{Area}}$, by first taking the limit $L\to\infty$ and then $W\to\infty$. This completes the proof that the Maxwell construction is indeed correct for all those LDPC ensembles for which (2.3) is known.

2.2.3 Proof of the equality of the MAP- and the BP-GEXIT curves above the MAP threshold

Using the equality of the MAP and area thresholds for uncoupled ensembles, we can derive more properties of uncoupled codes. The ensemble over which we average in the rest of this section will be exclusively LDPC(N,Λ,K). We first prove the following lemma establishing continuity in the channel parameter for the average per-bit conditional entropy as $N\to\infty$. Also, in order to make clear that the channel output depends on the channel entropy parameter h, we will write the former as $Y(\mathrm{h})$.

Lemma 6. *Given an ensemble* LDPC(N,Λ,K) *as in Theorem 4 and a smooth family of BMS channels ordered by degradation and parameterized by* h*, the quantity* $\lim_{N\to\infty}\frac{1}{N}\mathbb{E}_G\big[H(\underline{X}|\underline{Y}(\mathrm{h}))\big]$ *is a convex function of* h *and is Lipschitz continuous with Lipschitz constant* 1.

Proof. That the limit exists and the function is well defined is a consequence of Theorem 4. We use the fact that for any binary linear code the function $\frac{1}{N}H(\underline{X}|\underline{Y}(\mathrm{h}))$ is differentiable and

its derivative is increasing with values between 0 and 1 [MÓ6, Theorem 5.2, Corollary 5.1], so it is convex and Lipschitz continuous with Lipschitz constant 1. Taking the average over the code ensemble preserves these two properties. Passing to the limit $N \to \infty$, Lipschitz continuity and convexity are also preserved, because they are both defined by non-strict inequalities, which are maintained under the pointwise limit. □

The MAP-GEXIT function g^{MAP} is defined [MMRU09, Definitions 3 and 6] as

$$g^{\mathrm{MAP}}(\mathrm{h}) = \limsup_{N\to\infty} \frac{1}{N} \mathbb{E}_G \left[\frac{\mathrm{d}}{\mathrm{dh}} H(\underline{X}|\underline{Y}(\mathrm{h})) \right]. \tag{2.4}$$

We lower bound the area below g^{MAP} above the MAP threshold as follows:

$$\begin{aligned}
&\int_{\mathrm{h}_{\mathrm{MAP}}}^{1} g^{\mathrm{MAP}}(\mathrm{h})\,\mathrm{dh} = \\
&= \int_{\mathrm{h}_{\mathrm{MAP}}}^{1} \left(\limsup_{N\to\infty} \frac{1}{N} \mathbb{E}_G \left[\frac{\mathrm{d}}{\mathrm{dh}} H(\underline{X}|\underline{Y}(\mathrm{h})) \right] \right) \mathrm{dh} \\
&\overset{(a)}{\geq} \limsup_{N\to\infty} \int_{\mathrm{h}_{\mathrm{MAP}}}^{1} \frac{1}{N} \mathbb{E}_G \left[\frac{\mathrm{d}}{\mathrm{dh}} H(\underline{X}|\underline{Y}(\mathrm{h})) \right] \mathrm{dh} \\
&\overset{(b)}{=} \lim_{N\to\infty} \left(\frac{1}{N} \mathbb{E}_G H(\underline{X}|\underline{Y}(1)) - \frac{1}{N} \mathbb{E}_G H(\underline{X}|\underline{Y}(\mathrm{h}_{\mathrm{MAP}})) \right) \\
&\overset{(c)}{=} R - 0 = R,
\end{aligned} \tag{2.5}$$

where in step (a) we use the Fatou Lemma (note that the integrand on the r.h.s. is bounded), in step (b) we integrate and then use the existence of limits provided by Theorem 4 to replace lim sup with lim, and in step (c) we observe the following. For the first term, since at h = 1 the channel is completely useless, we have that $H(\underline{X}|\underline{Y}(1)) = H(\underline{X})$, which when scaled by N is nothing else than the rate of the code; in the large blocklength limit, the average of this over the ensemble coincides with the design rate $R = 1 - \bar{d}/K$. For the second term, note that $\lim_{N\to\infty} \frac{1}{N} \mathbb{E}_G\big[H(\underline{X}|\underline{Y}(\mathrm{h}))\big] = 0$ which follows from the of continuity in h obtained in Lemma 6.

The BP-GEXIT curve is defined [MMRU09, Definition 6] by

$$g^{\mathrm{BP}}(\mathrm{h}) = \lim_{\ell\to\infty} \limsup_{N\to\infty} \frac{1}{N} \mathbb{E}_G \left[\sum_v g^{\mathrm{BP}}_{G,v}(\mathrm{h}) \right], \tag{2.6}$$

$$g^{\mathrm{BP}}_{G,v}(\mathrm{h}) = \frac{\partial H(X_v|Y_v(\mathrm{h}_v), \Phi_v^{\ell}(\mathrm{h}))}{\partial \mathrm{h}_v}\Big|_{\mathrm{h}_v=\mathrm{h}}, \tag{2.7}$$

where $\Phi_v^{\ell}(\mathrm{h})$ is the BP estimate of X_v based on a computation tree of depth ℓ. An equivalent form is given by Equation (C.4) in Appendix C.2.

It is known that (see Lemma 9 in [MMRU09])

$$g^{\mathrm{MAP}}(\mathrm{h}) \le g^{\mathrm{BP}}(\mathrm{h}), \text{ for all } \mathrm{h} \in [0,1]. \tag{2.8}$$

The area threshold mentioned before is defined as the solution $\mathrm{h}_{\mathrm{area}}$ to the equation

$$\int_{\mathrm{h}_{\mathrm{area}}}^{1} g^{\mathrm{BP}}(\mathrm{h})\,\mathrm{dh} = R. \tag{2.9}$$

Using then the equality of the MAP and area thresholds established in the previous subsection for the above-mentioned class of LDPC codes and using (2.5) and (2.9) we obtain

$$\int_{\mathrm{h}_{\mathrm{MAP}}}^{1} (g^{\mathrm{BP}}(\mathrm{h}) - g^{\mathrm{MAP}}(\mathrm{h}))\,\mathrm{dh} \le R - R = 0. \tag{2.10}$$

The positivity of the integrand (cf. (2.8)) entails the following result.

Theorem 7. *Given an* LDPC(N,Λ,K) *ensemble and a smooth family of channels indexed by the entropy parameter* h*, the two curves* g^{MAP} *and* g^{BP} *are equal almost everywhere above the MAP threshold, as long as the MAP threshold is at least* $\bar{\mathrm{h}}$ *defined in Lemma 10 below.*[2]

The discussion of (2.5) also entails the following result, which will be useful subsequently. Among others, this allows us to exchange the lim inf with lim in the expression for the MAP threshold.

Proposition 8. *The limit* $\lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mathrm{LDPC}(N,\Lambda,K)} H(\underline{X}|\underline{Y}(\mathrm{h}))$ *exists for all values of* h*, and furthermore*

$$\int_{\mathrm{h}_0}^{1} g^{MAP}(\mathrm{h})\,\mathrm{dh} = R - \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mathrm{LDPC}(N,\Lambda,K)} H(\underline{X}|\underline{Y}(\mathrm{h}_0)),$$

where $R = 1 - \Lambda'(1)/K$ *is the rate of the code.*

2.2.4 Exactness of the replica-symmetric formula

The previous result, namely the equality of the BP and MAP GEXIT curves, allows us to settle another conjecture. We can prove that under certain conditions (above the MAP threshold) the potential functional [KYMP12], [KYMP14], also called replica-symmetric functional, is in fact equal to the conditional entropy $H(\underline{X}|\underline{Y})$. Note that while the former is a quantity derived by message passing, the latter is related to combinatorial optima. Also, unlike GEXIT curves, these quantities make sense already without considering the channel as part of a smooth family and thus in a sense appear to be more natural.

[2] The value $\bar{\mathrm{h}}$ will always be under the MAP threshold as long as degree are large enough.

In order to define the potential functional (or replica-symmetric functional), we need to introduce the density evolution operations. The beliefs that are transmitted during BP have distributions that are symmetric measures.

We restrict ourselves now to regular LDPC ensembles with left and right degrees d_l and d_r, respectively. However, since the derivation holds more generally, we will work with the polynomials Λ, P and λ, ρ as left and right degrees from the node and from the edge perspective, respectively. For us, they take the simple forms $\lambda(u) = u^{d_l-1}$, $\rho(u) = u^{d_r-1}$, $\Lambda(u) = u^{d_l}$ and $P(u) = u^{d_r}$.

The density evolution (DE) equation can then be written as $\mathsf{x}^{\ell+1} = \mathsf{c} \circledast \lambda^{\circledast}(\rho^{\boxtimes}(\mathsf{x}^{\ell}))$. The fixed point that can be reached by starting with $\mathsf{x}^0 = \Delta_0$ will be called *forward DE fixed point* and will be denoted by x_{c}.

We are now ready to define the replica-symmetric functional, which depends on the channel c and the message density x as

$$\begin{aligned}\Phi(\mathsf{x},\mathsf{c}) = &-\frac{L'(1)}{R'(1)} H(R^{\boxtimes}(\mathsf{x})) - L'(1) H(\rho^{\boxtimes}(\mathsf{x})) \\ &+ L'(1) H(\mathsf{x} \boxtimes \rho^{\boxtimes}(x)) + H(\mathsf{c} \circledast L^{\circledast}(\rho^{\boxtimes}(\mathsf{x}))).\end{aligned} \tag{2.11}$$

For a more complete exposition of this formalism, the identity of the potential functional and the replica symmetric functional properties, and various properties of the two operations $\circledast$ and $\boxtimes$, please refer to [KYMP14] (note that $\Phi(\mathsf{x},\mathsf{c})$ is equal to minus the function $U(\mathsf{x},\mathsf{c})$ of reference [KYMP14]).

The replica-symmetric formula conjectures that

$$\lim_{N\to+\infty} \frac{1}{N} \mathbb{E}_{\mathrm{LDPC}(N,d_l,d_r)} H(\underline{X}|\underline{Y}(\mathsf{c})) = \sup_{\mathsf{x}\in\mathscr{X}} \Phi(\mathsf{x},\mathsf{c}). \tag{2.12}$$

We prove this conjecture for standard regular LDPC codes with large enough, but fixed, d_l, d_r and also require even d_r. The proof of this conjecture is a consequence of Theorem 9 below.

This theorem states that in a region of channels above the MAP threshold characterized by a regularity condition, this functional evaluated at the right fixed point (which is algorithmic in nature as it comes from message passing) is equal to the conditional entropy, which is combinatorial in nature.

To express the regularity constraint, we first define the region of channels above the MAP threshold:

$$\mathscr{C}_0 = \{\mathsf{c} \in \mathscr{X} : \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathrm{LDPC}(N,d_l,d_r)} H(\underline{X}|\underline{Y}(\mathsf{c})) > 0\}.$$

Ideally, we would like our result to hold in the whole of this region, but, unfortunately, we need

to add a Lipschitz type of restriction. Let

$$\mathscr{C}_1 = \big\{\mathsf{c}_0 \in \mathscr{X} : \text{there is } \delta > 0 \text{ s.t. for all } \mathsf{c}, \mathsf{c}' \in [\mathsf{c}_0, \Delta_0] \\ \text{we have that } \left|\frac{\mathscr{B}(\mathsf{x}_\mathsf{c} - \mathsf{x}_{\mathsf{c}'})}{\mathscr{B}(\mathsf{c} - \mathsf{c}')}\right| \le \frac{1}{\delta}\big\}, \tag{2.13}$$

where $\mathscr{B}(\cdot)$ is the Bhattacharyya functional defined by (C.5), and

$$[\mathsf{c}_0, \Delta_0] = \{\mathsf{c} : \mathsf{c} = p\mathsf{c}_0 + (1-p)\Delta_0, \text{ for some } p \in [0,1]\}.$$

Note that the regions $\mathscr{C}_0$ and $\mathscr{C}_1$ depend on the parameters of the code.

Theorem 9. *Given the regular ensemble* LDPC(N, d_l, d_r) *with even* d_r*, for any channel* $\mathsf{c} \in \mathscr{C}_0 \cap \mathscr{C}_1$ *we have that*

$$\Phi(\mathsf{x}_\mathsf{c}, \mathsf{c}) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathrm{LDPC}(N,d_l,d_r)} H(\underline{X}, \underline{Y}(\mathsf{c})).$$

As the proof is fairly technical, we defer it to Appendix C.2.

We show now that for large degree pairs, $\mathscr{C}_0 \subseteq \mathscr{C}_1$, i.e. the theorem holds everywhere above the MAP threshold. This is made precise by Lemma 18 from [KRU12], reproduced below, which states that all channels with entropy above a value that goes to 0 as the right degree increases are in $\mathscr{C}_1$.

Lemma 10. *Let* d_l *and* d_r *be fixed numbers. There is a constant*[3] $\bar{\mathrm{h}}$ *depending only on the degrees* d_l *and* d_r *satisfying*

$$\bar{\mathrm{h}} < \frac{e^{1/4}\sqrt{2}}{d_r^{1/4}} \tag{2.14}$$

such that $\{\mathsf{c} \in \mathscr{X} : H(\mathsf{c}) > \bar{\mathrm{h}}\} \subseteq \mathscr{C}_1$.

We can readily see that for large degrees the right hand side of condition (2.14) approaches 0. Also, for large degrees, the MAP threshold approaches capacity and is bounded away from 0 uniformly for all channel families. This implies that $\mathscr{C}_0 \subseteq \mathscr{C}_1$ and hence $\mathscr{C}_0 \cap \mathscr{C}_1 = \mathscr{C}_0$.

We believe that the theorem remains true without this technical condition. Proving that this is indeed the case is an interesting open problem.

Let us conclude this paragraph by remarking that the above considerations imply the replica-symmetric formula (2.12) for large enough d_l, d_r and where d_r is an even number. From

[3] An expression for $\bar{\mathrm{h}}$ can be found in Lemma 18 of [KRU12].

[Mon05] we know that (for any BMS channel and d_r even)

$$\lim_{N\to\infty}\frac{1}{N}\mathbb{E}_{\mathrm{LDPC}(N,d_l,d_r)}H(\underline{X}|\underline{Y}(\mathsf{c}))\geq\sup_{\mathsf{x}\in\mathscr{X}}\Phi(\mathsf{x},\mathsf{c}). \tag{2.15}$$

Note first that for $\mathsf{c}\notin\mathscr{C}_0$ we have by definition $\lim_{N\to\infty}\frac{1}{N}\mathbb{E}_{\mathrm{LDPC}(N,d_l,d_r)}H(\underline{X}|\underline{Y}(\mathsf{c}))=0$. Thus

$$0\geq\sup_{\mathsf{x}\in\mathscr{X}}\Phi(\mathsf{x},\mathsf{c})\geq\Phi(\Delta_\infty,\mathsf{c})=0, \tag{2.16}$$

so (2.12) is satisfied for $\mathsf{c}\notin\mathscr{C}_0$. Now consider $\mathsf{c}\in\mathscr{C}_0$. Whenever $\mathscr{C}_0\cap\mathscr{C}_1=\mathscr{C}_0$ (e.g when d_l,d_r are large enough) Theorem 9 implies

$$\Phi(\mathsf{x}_c,\mathsf{c})=\lim_{N\to\infty}\frac{1}{N}\mathbb{E}_{\mathrm{LDPC}(N,d_l,d_r)}H(\underline{X}|\underline{Y}(\mathsf{c}))\geq\sup_{\mathsf{x}\in\mathscr{X}}\Phi(\mathsf{x},\mathsf{c})\geq\Phi(\mathsf{x}_c,\mathsf{c}) \tag{2.17}$$

and hence again (2.12) holds for $\mathsf{c}\in\mathscr{C}_0$.

2.3 Some useful lemmas

We present in this section two results that are quite general in nature, meaning that they are true for any linear code. They already appear in [Mon05, Mac07], but we reproduce short proofs here in order to make the exposition self-contained. The symmetry of the channel is a property that seems indispensable for the proofs in the rest of this paper, and we will need it in the form of the Nishimori Identity. The channel used for transmission needs to be BMS, symmetry being the crucial ingredient.

Lemma 11 (Nishimori Identity)**.** *Fix a graph G (no constraints on the check node degrees needed here) and a channel $\mathsf{c}\in\mathscr{X}$. For any odd positive integer m we have*

$$\mathbb{E}_h\big[\langle\sigma_b\rangle^m\big]=\mathbb{E}_h\big[\langle\sigma_b\rangle^{m+1}\big], \tag{2.18}$$

where $b=(b_1,\dots,b_J)$ is a vector of variable nodes (which need not belong a check constraint) of arbitrary length, and $\sigma_b=\sigma_{b_1}\cdots\sigma_{b_J}$.

Proof. We will assume here that the measure c does not contain mass at infinity. Extending to the general case can easily be done by considering the point mass at $+\infty$ separately. Because of channel symmetry, the measure defined by $d\mathsf{s}(\mathsf{h})=e^{-h}d\mathsf{c}(h)$ has the property $d\mathsf{s}(h)=d\mathsf{s}(-h)$. Using the memoryless property of the channel, the l.h.s. of (2.18) can be written as

$$\mathbb{E}_h\big[\langle\sigma_b\rangle^m\big]=\int\langle\sigma_b\rangle^m\prod_{v\in V}e^{h_v}\,\mathrm{d}\mathsf{s}(h_v). \tag{2.19}$$

We now observe that due to channel symmetry the above quantity is preserved under the

transformation $h_v \mapsto h_v\tau_v$, $\sigma_v \mapsto \sigma_v\tau_v$, if τ is a codeword. As a matter of fact, the transformed HLLRs $h_v\tau_v$ are those received when the codeword τ was transmitted, instead of the all-+1 codeword.

We now perform an average over all codewords τ, obtaining

$$\mathbb{E}_h\big[\langle\sigma_b\rangle^m\big] = \frac{1}{|\mathscr{C}(G)|}\sum_{\tau\in\mathscr{C}(G)}\int \langle\sigma_b\tau_b\rangle^m \prod_{v\in V} e^{h_v\tau_v}\,\mathsf{ds}(h_v),$$

where $\mathscr{C}(G)$ is the set of all codewords.

Note that the Gibbs bracket above averages over σ, and thus we can safely take τ_b out of the bracket. Since m is odd, $\tau_b^m = \tau_b$. Next we use the definition of Gibbs measure (equation (1.19)) to replace $\sum_{\tau\in\mathscr{C}(G)} e^{\underline{h}\cdot(\underline{\tau}-1)}\tau_b$ with $Z(G)\,\langle\tau_b\rangle$. We obtain

$$\mathbb{E}_h\big[\langle\sigma_b\rangle^m\big] = \frac{1}{|\mathscr{C}(G)|}\int Z(G)\,\langle\sigma_b\rangle^{m+1}\prod_{v\in V}\mathsf{ds}(h_v). \tag{2.20}$$

Expanding $Z(G)$ into $\sum_{\lambda\in\mathscr{C}(G)} e^{h\cdot\lambda}$ we get

$$\mathbb{E}_h\big[\langle\sigma_b\rangle^m\big] = \frac{1}{|\mathscr{C}(G)|}\sum_{\lambda\in\mathscr{C}(G)}\int \langle\sigma_b\rangle^{m+1}\prod_{v\in V} e^{h_v\lambda_v}\,\mathsf{ds}(h_v).$$

A second gauge transformation $h_v \mapsto h_v\lambda_v$, $\sigma_v \mapsto \sigma_v\lambda_v$ allows us to cancel all λ factors, since $\lambda_v^2 = 1$. All $|\mathscr{C}(G)|$ terms in the sum are equal, so the expression simplifies to

$$\mathbb{E}_h\big[\langle\sigma_b\rangle^m\big] = \int \langle\sigma_b\rangle^{m+1}\prod_{v\in V} e^{h_v}\,\mathsf{ds}(h_v), \tag{2.21}$$

and thus the claim follows. □

The next result quantifies the effect on $\log Z$ of one extra check node added to some general linear code. This is the main reason why we chose to work with $\log Z$ instead of the conditional entropy.

Lemma 12. *Given any graph G and an additional check constraint b, we have that*

$$\mathbb{E}_h\big[\log Z(G\cup b) - \log Z(G)\big] = -\log 2 + \sum_{r\in 2\mathbb{Z}_+}\frac{\mathbb{E}_h\big[\langle\sigma_b\rangle_G^r\big]}{r^2-r}.$$

In particular, $-\log 2 \le \log Z(G\cup b) - \log Z(G) \le 0$.

The second part of the statement shows that the contribution of one extra check node gives only a finite variation in $\log Z$, and it turns out to be very useful for the cases where we need to

show that two similar ensembles have log-partition functions that are asymptotically identical.

Proof. Using the definition of the partition function $Z(G \cup b)$, we are able to write

$$Z(G \cup b) = \sum_{\underline{\sigma} \in \mathscr{X}^V} e^{\underline{h} \cdot (\underline{\sigma} - \underline{1})} \frac{1+\sigma_b}{2} \prod_{a \in G} \frac{1+\sigma_a}{2} = Z(G) \left\langle \frac{1+\sigma_b}{2} \right\rangle_G .$$

Then $\log Z(G \cup b) - \log Z(G) = -\log 2 + \log(1 + \langle \sigma_b \rangle)$. Expanding the logarithm into power series, we obtain

$$\log(1 + \langle \sigma_b \rangle) = \sum_{j \geq 1} \frac{(-1)^{j+1}}{j} \langle \sigma_b \rangle^j . \tag{2.22}$$

We now use the Nishimori Identities (Lemma 11) with $\mathbb{E}_h\big[\langle \sigma_b \rangle^{j-1}\big] = \mathbb{E}_h\big[\langle \sigma_b \rangle^j\big]$, for even j. This allows us to merge each odd-index term with the following term, proving the claim. □

Let us now analyze the terms of the form $\langle \sigma_b \rangle_G^r$ that appear in the last lemma. For this purpose, we will work with the product measure $\mu^{\otimes r}$. The measure space here is the one of r-tuples $(\sigma^{(1)}, \ldots, \sigma^{(r)})$, where $\sigma^{(j)} \in \mathscr{X}^V$. Because the product measure is just the measure of r independent copies of the measure (henceforth called *replicas*), it is easy to check that

$$\langle \sigma_b \rangle_G^r = \left\langle \sigma_b^{(1)} \cdots \sigma_b^{(r)} \right\rangle_G^{\otimes r} .$$

The $\otimes r$ sign at the top right of the bracket is just to remind us that we deal with the product measure $\mu^{\otimes r}$. Since this is evident from context, we will drop this sign in the future. We are then able to restate the last lemma as follows.

Corollary 13. *Given any graph G and an additional check constraint b, we have that*

$$\mathbb{E}_h\big[\log Z(G \cup b) - \log Z(G)\big] = -\log 2 + \mathbb{E}_h \sum_{r \in 2\mathbb{Z}_+} \frac{\left\langle \sigma_b^{(1)} \cdots \sigma_b^{(r)} \right\rangle_G}{r^2 - r}. \tag{2.23}$$

2.4 The configuration model

In this section we introduce the language needed to describe and dissect all the kinds of ensembles that we need.

We assume that the configuration pattern introduced in Section 2.1.2 is already fixed, i.e., it has been properly sampled at an earlier stage, and there are at least $N\bar{d}(1 - N^{-\eta})$ and at most $N\bar{d}(1 + N^{-\eta})$ sockets at every position of the chain. By a straightforward application of a Azuma-Hoeffding type of inequality and the union bound for all positions, this happens with

high probability[4] in the first stage, as long as $0 < \eta < \frac{1}{2}$. The fixed underlying configuration pattern is always of the coupled kind, i.e., there are L groups of N variable nodes each; the simple kind will arise from the conditions $W = 1$ and $W = L$.

Given the fixed configuration pattern, each variable node v has a *target degree* $d(v)$, and exactly $d(v)$ *sockets* numbered from 1 to $d(v)$. Given a socket s, let $\mathrm{var}(s)$ denote the variable node that it is part of; by σ_s we understand $\sigma_{\mathrm{var}(s)}$. Let $\mathrm{pos}(v)$ denote the position of the variable v, with the notation extending to sockets in the obvious manner: $\mathrm{pos}(s) = \mathrm{pos}(\mathrm{var}(s))$. We also set S to be the set of all sockets and put $S_z = \{s \in S : \mathrm{pos}(s) = z\}$, i.e. the set of sockets at a particular position.

Check nodes will connect to sockets, so a check node a will have the form of a K-tuple $(a_1, \ldots, a_K)$, where the components a_j are sockets. Note that the ordering of the edges leaving the check-node matters, so the check also "stores" this information. We say that a check node a has *type* $\alpha = (\alpha_1, \ldots, \alpha_K)$ if $\alpha_j = \mathrm{pos}(a_j)$, for all $1 \le j \le K$. In other words, the type records the positions of the variable nodes to which the check node a connects.

We now consider random types, of which there are three kinds that are important to us:

- **The connected random type.** This random type is uniformly distributed over the set of all L^K possible types. We denote this distribution by **conn**.
- **The disconnected random type.** This type is uniformly distributed over the set of all types whose entries are all equal, i.e., types of the form $(z, z, \ldots, z)$. We denote this distribution by **disc**.
- **The coupled random type.** We choose a position z uniformly at random and the result is a type uniformly distributed over the set of all types whose entries lie in the set $\{z, \ldots z + W - 1\}$. We denote this distribution by **coup**.

We now define the *positional occupation vector* occ_α of a type α to be a vector whose z entry counts the number of occurrences of position z in type α. As an example, if $K = 6$ and $\alpha = (1, 3, 2, 5, 1, 3)$ and assuming there are $L = 5$ positions, then $\mathrm{occ}_\alpha = (2, 1, 2, 0, 1)$.

Given a multiset of types Γ (a set of types where duplicates can appear), we extend the definition of the positional occupation vector to $\mathrm{occ}_\Gamma = \sum_{\alpha \in \Gamma} \mathrm{occ}_\alpha$.

We call a multiset of types *m-admissible* if $\mathrm{occ}_\Gamma(z) \le |S_z| - m$, for all positions z. In other words, an m-admissible set of types Γ ensures that there exists a graph G whose check constraints match one-to-one the types in Γ (we say that G is *compatible* with Γ), and in addition, there are at least m sockets at each position that remain free. We will also use the word *admissible* to mean 0-admissible. One should think about the multiset of types as being a kind of "pre-graph", where only the positions of the edges are decided, but not yet the actual sockets.

[4] By *with high probability* we mean that the event in question happens with probability $1 - o(1/\mathrm{poly}(N))$. The parameters L and W are considered constant for this purpose.

The random graph generated by an admissible multiset of types Γ is simply given by the uniform measure over all graphs that are compatible with Γ. To sample this random graph, the algorithm is as follows: start with the empty graph; for each type $\alpha = (\alpha_1, \dots, \alpha_K)$ in the multiset Γ (the order is immaterial), pick *distinct* a_i uniformly at random from the free sockets at position α_i, and add check constraint $(a_1, \dots, a_K)$ to the graph. We will use this check-generating procedure often, so we will say that check constraint a is chosen according to distribution $\nu(\alpha, G)$ that depends on the type α, and the part G of the graph that is already in place. Let B_α be the set of check constraints that are compatible with α and are connected to free sockets (sockets that do not appear in G). Note that a socket must never be used twice, so they are chosen without replacement. Then $\nu(\alpha, G)$ is the uniform measure on B_α.

We also trivially extend this definition to the case of a random graph generated by a *random* multiset of types. This latter random object will be typically a list of independent random types of one of the three kinds *connected*, *disconnected* and *coupled*. For the sake of precision, in case the multiset of types is not admissible (by this we mean m-admissible, where m will be fixed later), we define the generated random graph to be the empty one.

We now introduce a quantity inspired from statistical physics that plays an important role in what comes next, namely the *positional overlap functions*. Fix a configuration graph G, a channel realization h, and the number r of replicas of the measure $\mu_{G,h}$. Let $F_z \subseteq S_z$ be the set of free sockets at position z (free sockets being those that do not appear in any check constraint of G). The *positional overlap functions* Q_z, indexed by a position z, are defined by

$$Q_z(\sigma^{(1)}, \dots, \sigma^{(r)}) = \frac{1}{|F_z|} \sum_{s \in F_z} \sigma_s^{(1)} \cdots \sigma_s^{(r)}. \tag{2.24}$$

The next statement describes the link between the overlap functions and the replica averages introduced by Lemma 12.

Lemma 14. *Given a number $m > K^2$, a fixed channel realization, a fixed graph G whose associated type set is m-admissible and fixed type α, we have*

$$\mathbb{E}_{a:\nu(\alpha,G)} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle_G = \left\langle \prod_{j=1}^{K} Q_{\alpha_j}(\sigma^{(1)}, \dots, \sigma^{(r)}) \right\rangle + O\left(\frac{1}{m}\right). \tag{2.25}$$

Proof. The left hand side is nothing else than the average over all possible a that are compatible with the type α and connect to free sockets. In other words,

$$\frac{1}{|B_\alpha|} \sum_{a \in B_\alpha} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle. \tag{2.26}$$

The goal is to somehow factorize the sum, but the fact that sockets are not replaced makes it a bit harder. Suppose that, contrary to our current model, free sockets are allowed to be

chosen with replacement, that is, it is possible to have $a_i = a_j$ for $i \neq j$. Let B'_α be the set of all (pseudo-)check constraints that are compatible with α, and where sockets are allowed to appear multiple times. Then B'_α can be written as a product:

$$B'_\alpha = F_{\alpha_1} \times \ldots \times F_{\alpha_K},$$

where the set F_z is the set of free sockets at position z. The idea is now that we can replace B_α with B'_α in the average (2.26) without losing too much, while gaining the ability to factorize the sum.

The relation between the two, which is proven in Appendix C.1, is

$$\frac{1}{|B_\alpha|} \sum_{a \in B_\alpha} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle = \frac{1}{|B'_\alpha|} \sum_{a \in B'_\alpha} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle + O\left(\frac{1}{m}\right). \tag{2.27}$$

Now we are in a better position, since on the r.h.s. any entry a_i is chosen independently of the others. We rewrite the sum over B'_α in the following way:

$$\frac{1}{|F_{\alpha_1}|} \sum_{a_1 \in F_{\alpha_1}} \cdots \frac{1}{|F_{\alpha_K}|} \sum_{a_K \in F_{\alpha_K}} \langle \sigma_{a_1}^{(1)} \cdots \sigma_{a_K}^{(1)} \cdots \sigma_{a_1}^{(r)} \cdots \sigma_{a_K}^{(r)} \rangle.$$

Taking the bracket outside and factorizing, we obtain

$$\left\langle \left(\frac{1}{|F_{\alpha_1}|} \sum_{a_1 \in F_{\alpha_1}} \sigma_{a_1}^{(1)} \cdots \sigma_{a_1}^{(r)} \right) \cdots \left(\frac{1}{|F_{\alpha_K}|} \sum_{a_K \in F_{\alpha_K}} \sigma_{a_K}^{(1)} \cdots \sigma_{a_K}^{(r)} \right) \right\rangle,$$

which we can identify as the bracketed product of positional overlap functions on the right hand side of (2.25). □

Lemma 15. *Let G be a graph whose type multiset is m-admissible, and fix the channel realization h. Then the following inequalities hold:*

$$\mathbb{E} \sum_{\substack{\alpha:\mathbf{conn} \\ a:\nu(\alpha,G)}} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle_G \leq \mathbb{E} \sum_{\substack{\alpha:\mathbf{coup} \\ a:\nu(\alpha,G)}} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle_G + O(1/m), \tag{2.28}$$

$$\mathbb{E} \sum_{\substack{\alpha:\mathbf{coup} \\ a:\nu(\alpha,G)}} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle_G \leq \mathbb{E} \sum_{\substack{\alpha:\mathbf{disc} \\ a:\nu(\alpha,G)}} \langle \sigma_a^{(1)} \cdots \sigma_a^{(r)} \rangle_G + O(1/m). \tag{2.29}$$

Proof. The claim follows by Lemma 14 if we manage to show the following two inequalities:

$$\mathbb{E}_{\alpha:\mathbf{conn}} \langle Q_{\alpha_1} \cdots Q_{\alpha_K} \rangle \leq \mathbb{E}_{\alpha:\mathbf{coup}} \langle Q_{\alpha_1} \cdots Q_{\alpha_K} \rangle, \tag{2.30}$$

$$\mathbb{E}_{\alpha:\mathbf{coup}} \langle Q_{\alpha_1} \cdots Q_{\alpha_K} \rangle \leq \mathbb{E}_{\alpha:\mathbf{disc}} \langle Q_{\alpha_1} \cdots Q_{\alpha_K} \rangle, \tag{2.31}$$

where the dependence of the positional overlap functions on the spin systems $\sigma^{(j)}$ has been dropped in order to lighten notation.

We rewrite the quantities above as follows:

$$\mathbb{E}_{\alpha:\mathbf{conn}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \frac{1}{L^K}\sum_{\substack{(\alpha_1,\ldots,\alpha_K)\\ \in[L]^K}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \left\langle\left(\frac{1}{L}\sum_{z\in[L]}Q_z\right)^K\right\rangle, \tag{2.32}$$

$$\begin{aligned}\mathbb{E}_{\alpha:\mathbf{coup}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle &= \frac{1}{L}\sum_{z'\in[L]}\frac{1}{W^K}\sum_{\substack{(\alpha_1,\ldots,\alpha_K)\\ \in\{z',\ldots,z'+W-1\}^K}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle \\ &= \left\langle\frac{1}{L}\sum_{z'\in[L]}\left(\frac{1}{W}\sum_{z=z'}^{z'+W-1}Q_z\right)^K\right\rangle,\end{aligned} \tag{2.33}$$

$$\mathbb{E}_{\alpha:\mathbf{disc}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \frac{1}{L}\sum_{z\in[L]}\langle Q_z\cdots Q_z\rangle = \left\langle\frac{1}{L}\sum_{z\in[L]}Q_z^K\right\rangle. \tag{2.34}$$

In the above expressions we assume Q_z is defined for all integer z using the relation $Q_{z'} = Q_{z''}$ whenever $z' \equiv z''(\mod L)$. Both inequalities (2.30) and (2.31) are proved by an application of Jensen's Inequality using the convexity of the function $x \mapsto x^K$, for even K. □

2.5 The interpolation

We now move a bit further and consider random ensembles of graphs. These are obtained in the following way: first we prescribe the numbers of random types of each kind that we want, i.e. how many types should be connected, disconnected and coupled. Afterwards, the random types are sampled according to the distributions prescribed. Finally the graph is chosen uniformly to match the multiset of types, in the spirit of the previous section.

We use the notation $G : \left\{\begin{smallmatrix} t_1\times\mathbf{coup}\\ t_2\times\mathbf{disc}\end{smallmatrix}\right\}$ to say that G is sampled in the way outlined above, where t_1 and t_2 are the number of random types of the coupled kind and disconnected kind, respectively. Of course, we could specify any combination of the three kinds, **conn** included.

Now we need to set the number of check nodes in the ensemble. There are two conflicting constraints we would like to satisfy: first, the set of types needs to be admissible with high probability — so that the sampled graph exists in the form we want; second, the number of free sockets that remain should be small, in the sense that the proportion of free sockets needs to vanish in the limit.

The average amount of check nodes needed to use all available sockets is (ideally) $NL\bar{d}/K$. However, there is a fluctuation ($\pm N^{1-\eta}\bar{d}$ at each position) of the amount of available sockets and it might not be possible to connect actual check nodes to all sockets (for example, because of window constraints). As a consequence, we choose the actual size of the graph (by this we mean the number of multi-edges, i.e. check nodes) to be $T = NL\bar{d}(1-N^{-\gamma})/K$, so in case the graph is admissible there will be $O(N^{1-\gamma})$ free sockets left at each position. The exponent γ is arbitrary, as long as $0 < \gamma < \eta$. The next lemma confirms that by using this value for T, the resulting set of types is admissible with high probability.

Lemma 16. *Let $\alpha^1, \ldots, \alpha^T$ be random types, each drawn from a distribution that is either* **conn**, **disc** *or* **coup** *(could be different for each type). Then with high probability (more precisely $1 - O(\exp(-\kappa N^{1-2\gamma}))$, for some positive constant κ the resulting multiset of types is $\bar{d}N^{1-\gamma}/2$-admissible.*

Proof. The plan is the following: fix a position z, and show that the number of appearances of z as entries of $\alpha^1, \ldots, \alpha^T$ exceeds $TK/L + \bar{d}N^{1-\gamma}/2$ with a very small probability. Next, by the union bound over all positions z, we upper bound the probability that the graph is not $\bar{d}N^{1-\gamma}/2$-admissible and the lemma is proved.

We concentrate on the above claim, and define X_t to be the number of entries in α^t equal to z, for $1 \le t \le T$. Clearly the X_t are independent, bounded and their expectation equals K/L (the choice of distribution of α^t is immaterial as long as it is one of **conn**, **disc** or **coup**). Then by Hoeffding's Inequality, the probability that $\sum X_t$ deviates from its expectation TK/L decays very fast. More exactly,

$$\mathbb{P}\left[\sum_{t=1}^{T} X_t \ge \frac{TK}{L} + \frac{1}{2}\bar{d}N^{1-\gamma}\right] \le \exp\left(-\frac{\bar{d}^2 N^{2-2\gamma}}{2K^2 T}\right), \tag{2.35}$$

which proves the claim. □

The previous lemma essentially allows us to take the expectation over an ensemble of graphs without caring too much about non-admissibility. This enables us to prove a the following key lemma.

Lemma 17. *The following two inequalities hold:*

$$\mathbb{E}_{h,G:\{T\times\mathbf{conn}\}} \log Z(G) \le \mathbb{E}_{h,G:\{T\times\mathbf{coup}\}} \log Z(G) + O\left(N^{\gamma}\right), \tag{2.36}$$

$$\mathbb{E}_{h,G:\{T\times\mathbf{coup}\}} \log Z(G) \le \mathbb{E}_{h,G:\{T\times\mathbf{disc}\}} \log Z(G) + O\left(N^{\gamma}\right). \tag{2.37}$$

Proof. We only discuss the first of the two inequalities, since the proof of the other is identical. We will set up a chain of inequalities, at the ends of which sit the two quantities that we need to compare. This is the main idea of the *interpolation method*: finding a sequence of objects that transition "smoothly" between two objects that can differ significantly. In our case, it is easily seen that the claim follows if we are able to show that

$$\mathbb{E}_{h,G:\left\{\substack{(t+1)\times\mathbf{conn}\\(T-t-1)\times\mathbf{coup}}\right\}} \log Z(G) \le \mathbb{E}_{h,G:\left\{\substack{t\times\mathbf{conn}\\(T-t)\times\mathbf{coup}}\right\}} \log Z(G) + O\left(N^{\gamma-1}\right). \tag{2.38}$$

The two ensembles involved in inequality (2.36) lie at the endpoints of a chain of T inequalities of the form above, with t moving from 0 to $T-1$. The crucial observation here is that the two ensembles $\left\{\substack{(t+1)\times\mathbf{conn}\\(T-t-1)\times\mathbf{coup}}\right\}$ and $\left\{\substack{t\times\mathbf{conn}\\(T-t)\times\mathbf{coup}}\right\}$ can both be obtained by sampling a graph $\widetilde{G}$ from their common part, $\left\{\substack{t\times\mathbf{conn}\\(T-t-1)\times\mathbf{coup}}\right\}$ and in case G is not null, adding an extra random check

constraint sampled according to **conn** and **coup**, respectively. The plan is to show that the inequality (2.38) holds also when $\widetilde{G}$ is fixed, and then to average over $\widetilde{G}$.

Let us fix $m = \bar{d}N^{1-\gamma}/2$, and let us first deal with the case when the realization of the ensemble $\left\{{}^{t\times\mathbf{conn}}_{(T-t-1)\times\mathbf{coup}}\right\}$ is not m-admissible. This event occurs with a very small probability, sub-exponential according to Lemma 16. Since $\log Z(G) = O(N)$ (according to Lemma 12), the error obtained by not considering this case is extremely small and fits in the tolerated term $O\left(\frac{1}{N^{1-\gamma}}\right)$.

Otherwise, $\widetilde{G}$ is such that there are at least m free sockets at every position, and we need to show that

$$\mathbb{E}_h\mathbb{E}\underset{a:\nu(\alpha,\widetilde{G})}{{}_{\alpha:\mathbf{conn}}}\log Z(\widetilde{G}\cup a) \le \mathbb{E}_h\mathbb{E}\underset{a:\nu(\alpha,\widetilde{G})}{{}_{\alpha:\mathbf{coup}}}\log Z(\widetilde{G}\cup a).$$

We subtract $\log Z(\widetilde{G})$ on both sides and then use Lemma 12 to write the difference of log partition functions as a linear combination of brackets of the form $\langle\sigma^{(1)}\cdots\sigma^{(r)}\rangle_{\widetilde{G}}$, after which we can readily apply Lemma 15 and the claim follows. □

2.6 Retrieving the original LDPC ensembles

We will now investigate further the connection between the ensembles $\{T\times\mathbf{conn}\}$ and $\{T\times\mathbf{disc}\}$. In fact, they are both variants of the uncoupled ensembles introduced in the beginning of Section 2.1. The first one is very similar to LDPC(NL,Λ,K), and the second one is similar to L copies of LDPC(N,Λ,K). The only differences that occur are related to the case where there is a large deviation in the number of sockets generated in the first stage, or when the multisets of types generated by $\{T\times\mathbf{conn}\}$ and $\{T\times\mathbf{disc}\}$ are not admissible. Also since the first stage of the ensemble generation, where we obtain the configuration pattern, is the same in all cases, we condition on the event that the configuration pattern is known and that it satisfies the condition stated at the beginning of Section 2.4, namely that the number of sockets at each position is $N\bar{d}/K \pm O(N^{\eta})$.

We can easily see that the ensemble $\{T\times\mathbf{disc}\}$, conditioned on the fact that its realization is admissible, can be extended to L copies of the simple (i.e. uncoupled) ensemble on N variable nodes by adding $O(N^{1-\gamma})$ extra check constraints. Thus the scaled log partition function is the same up to a sub-linear term.

Can we say the same about the ensemble $\{T\times\mathbf{conn}\}$ and the simple ensemble on NL variable nodes? Yes, but it requires a lengthier argument. Let us look closer at the latter. This ensemble is not generated using types (since positions play no role here), but we can still count the occurrences of various types that appear in it. There are exactly L^K different types, and the next proposition estimates the probability that a particular random check constraint in the simple ensemble LDPC(NL,Λ,K) has a certain type. To see the crux of the problem, in the

$\{T \times \mathbf{conn}\}$ ensemble, the types are generated uniformly. Whereas in the simple ensemble, a position with considerably more occupied sockets than other positions has a lesser chance to be picked.

We will proceed by transforming the ensemble LDPC(NL, Λ, K) (the *simple* ensemble) into $\{T \times \mathbf{conn}\}$ (the *connected* ensemble) through only a small amount of check additions and deletions. Let X_α be the number of check nodes of type α that occur in a realization of the simple ensemble. For every type α, let Y_α be a random variable sampled according to $\mathrm{Bin}(T, L^{-K})$. If $X_\alpha > Y_\alpha$, then exactly $X_\alpha - Y_\alpha$ check nodes of type α selected uniformly at random from the existing ones are deleted from the simple ensemble. Otherwise, exactly $Y_\alpha - X_\alpha$ check nodes of type α are chosen uniformly at random from all possible combinations of compatible free sockets and inserted in the graph without replacement. All insertions of check nodes must occur after all deletions have been performed (the order of the types is important). If at any stage there are no free sockets at a particular position to choose from, it just means the underlying multiset of types (which here is given by the numbers Y_α) is not T-admissible, and we produce the trivial code.

In order to bound the number of check node insertions and deletions, we compute the first and second moments of $X_\alpha - Y_\alpha$. The total number of check nodes M in the simple ensemble is fixed for our purposes (depends only on the configuration pattern), so we can write $X_\alpha = \sum_a R_\alpha^a$, where R_α^a is the indicator random variable of the event that check node a has type α, and the sum ranges over all M check nodes.

Proposition 18. *The expectation and variance of $X_\alpha - Y_\alpha$ are given by*

$$\mathbb{E}[X_\alpha - Y_\alpha] = O(N^{1-\gamma}), \tag{2.39}$$

$$\mathrm{Var}[X_\alpha - Y_\alpha] = O(N^{2-\eta}). \tag{2.40}$$

Proof. We determine first the probability $\mathbb{E}R_\alpha^a$ that a check node a has type α. This event happens if and only if all sockets a_i to which a is connected are placed at positions α_i. For this, we need to evaluate the proportion of free sockets at each position (all sockets are free initially, because w.l.o.g. we can say that a is the first check node to be allocated). The number of sockets at any position is between $N\bar{d}(1-N^{-\eta})$ and $N\bar{d}(1+N^{-\eta})$; the number of occupied sockets is at most $K-1$ (from previous edges). Thus, the probability that $\mathrm{pos}(a_i) = \alpha_i$ is lower-bounded by

$$\frac{N\bar{d}(1-N^{-\eta}) - K}{NL\bar{d}(1+N^{-\eta})} = \frac{1}{L} - O(N^{-\eta}),$$

and, likewise, upper-bounded by

$$\frac{N\bar{d}(1+N^{-\eta})}{NL\bar{d}(1-N^{-\eta})} = \frac{1}{L} + O(N^{-\eta}).$$

It then follows that

$$\mathbb{E}R_\alpha^a = \left(\frac{1}{L} + O(N^{-\eta})\right)^K = \frac{1}{L^K} + O(N^{-\eta}). \tag{2.41}$$

For the second moments we need $\mathbb{E}\left[R_\alpha^a R_\beta^b\right]$, i.e. the probability that a and b have types α and β at the same time. The reasoning is essentially similar to the previous case, only now there are $2K$ edges to connect and at most $2K-1$ occupied sockets (by symmetry we can arrange that a and b are the first two check nodes to be allocated). Then we have

$$\mathbb{E}\left[R_\alpha^a R_\beta^b\right] = \left(\frac{1}{L} + O(N^{-\eta})\right)^{2K} = \frac{1}{L^{2K}} + O(N^{-\eta}). \tag{2.42}$$

By summing over all check nodes, we get $\mathbb{E}X_\alpha = \frac{M}{L^K} + O(N^{1-\eta})$ and after elementary calculations, $\mathrm{Var}X_\alpha = O(N^{2-\eta})$. Since Y_α is binomially distributed, and using $T = M + O(N^{1-\gamma})$, we have

$$\mathbb{E}Y_\alpha = \frac{T}{L^K} = \frac{M}{L^K} + O(N^{1-\gamma}),$$

and also

$$\mathrm{Var}Y_\alpha = T\frac{1}{L^K}\left(1 - \frac{1}{L^K}\right) = O(N),$$

which is much smaller than $\mathrm{Var}X_\alpha$. □

To show that the amount of inserted and deleted check nodes is small, we employ now the Chebyshev Inequality, which, for some value of the parameter ζ to be fixed shortly, reads

$$\mathbb{P}\left[\left|X_\alpha - Y_\alpha - O\left(N^{1-\gamma}\right)\right| \geq N^\zeta O\left(N^{1-\frac{\eta}{2}}\right)\right] \leq \frac{1}{N^{2\zeta}}.$$

We fix the values $\zeta = \frac{\eta}{4}$ and $\gamma = \frac{\eta}{2}$ (these choices are somewhat arbitrary), and simplifying we obtain

$$\mathbb{P}\left[|X_\alpha - Y_\alpha| \geq O\left(N^{1-\frac{\eta}{4}}\right)\right] \leq N^{-\frac{\eta}{2}}.$$

Using the union bound over all L^K possible types, the bound on the probability that the number of insertions and deletions is sub-linear in the way depicted above remains $O\left(N^{-\eta/2}\right)$. In case the the number of insertions and deletions is too large, we use the $O(N)$ we use the fact that $\log Z(G)$ is always $O(N)$ (see Lemma 12). This proves the following lemma.

Lemma 19. *Transmitting over a BMS channel, we have*

$$\mathbb{E}_{h,G:\mathrm{LDPC}(NL,\Lambda,K)}\log Z(G) \geq \mathbb{E}_{h,G:\{T\times\mathbf{conn}\}}\log Z(G) + O\left(N^{1-\frac{\eta}{4}}\right).$$

2.7 The large N limit

This section wraps up the proof of Theorem 4. The main ingredient is the content of Lemma 17, which can be written as

$$\begin{aligned}
&\mathbb{E}_{h,G:\{T\times\mathbf{conn}\}}\log Z(G) - O\left(N^{1-\gamma}\right) \\
&\quad\leq \mathbb{E}_{h,G:\{T\times\mathbf{coup}\}}\log Z(G) \\
&\quad\leq \mathbb{E}_{h,G:\{T\times\mathbf{disc}\}}\log Z(G) + O\left(N^{1-\gamma}\right).
\end{aligned} \tag{2.43}$$

Using the results from the previous section on the comparison with the simple ensembles and scaling everything by NL, we obtain

$$\begin{aligned}
&\frac{1}{NL}\mathbb{E}_{h,G:\mathrm{LDPC}(NL,\Lambda,K)}\log Z(G) - O\left(N^{-\gamma}\right) \\
&\quad\leq \frac{1}{NL}\mathbb{E}_{h,G:\{T\times\mathbf{coup}\}}\log Z(G) \\
&\quad\leq \frac{1}{N}\mathbb{E}_{h,G:\mathrm{LDPC}(N,\Lambda,K)}\log Z(G) + O\left(N^{-\gamma}\right).
\end{aligned} \tag{2.44}$$

The next step is to take the $N\to\infty$ limit, and in case it exists for the outer terms, which we are about to show, we can apply the "sandwich rule" to obtain Theorem 4. Note that the ensemble appearing in the middle is what we call $\mathrm{LDPC}(N,L,W,\Lambda,K)$ — we are of course not obliged to pick it as such: we could do another level of processing in the style of the previous section; however the current form is known to fulfill the Maxwell conjecture, so we need not go any further.

To show that the limit

$$\lim_{N\to\infty}\frac{1}{N}\mathbb{E}_{h,G:\mathrm{LDPC}(N,\Lambda,K)}\log Z(G)$$

exists, we use the following result, whose proof can be found in the Appendix of [BGT10].

Lemma 20 (The modified superadditivity theorem)**.** *Given $\alpha\in(0,1)$, suppose a non-negative sequence $\{a_{N,N\geq 1}\}$ satisfies*

$$a_{N_1+N_2} \geq a_{N_1} + a_{N_2} - O((N_1+N_2)^{\alpha}) \tag{2.45}$$

for every $N_1,N_2\geq 1$. Then the limit $\lim_{N\to\infty}\frac{a_N}{N}$ exists (it may be $+\infty$).

The claim then follows by setting the sequence a_N to be the negative of the sequence we study

(since $\log Z(G)$ are negative). It remains to be shown that superadditivity indeed holds.

Since this part is a somewhat simpler variation of the interpolation we have already seen, we only present the proof sketch. We consider a coupled ensemble consisting of only two positions($L = 2$) and interpolate between the cases $W = 1$ (disconnected case) and $W = 2$ (connected case). The novelty is that the number of variables at the first and second positions differ, they are N_1 and N_2, respectively. For the connected case, when edges from check nodes are connected, we do not pick the position at random, but rather weigh the choice by $\nu_1 = \frac{N_1}{N_1+N_2}$ and $\nu_2 = \frac{N_2}{N_1+N_2}$, respectively.

The only difference appears in the reasoning of Lemma 15, where the types are not uniformly distributed anymore. The types are now binary strings of length K, with the two symbols appearing denoting the position, one having weight ν_1, the other ν_2. The weight of the type is the product of the weights of the symbols it contains. If α is a type, let $\nu(\alpha)$ be the weight of that type. Then Equations (2.32) and (2.34) become

$$\mathbb{E}_{\alpha:\mathbf{conn}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \sum_{\alpha\in\{1,2\}^K} \nu(\alpha)\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \langle(\nu_1 Q_1+\nu_2 Q_2)^K\rangle,$$
$$\mathbb{E}_{\alpha:\mathbf{disc}}\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \sum_{z\in\{1,2\}} \nu_z\langle Q_{\alpha_1}\cdots Q_{\alpha_K}\rangle = \langle\nu_1 Q_1^K+\nu_2 Q_2^K\rangle,$$

and clearly the lemma remains true in this case as well.

2.8 Final remarks

The present analysis can be extended with almost no change to arbitrary check-node degree distributions whose generating polynomial $P(x) = \sum_{K\le 0}\rho_K x^K$ is convex for $x \in [-1,1]$. Experimental evidence suggests that even this condition can be relaxed, but new ideas seem to be required to extend the proofs. A possible route would be to show self-averaging properties for overlap functions, which would allow to use the convexity of $x \mapsto P(x)$ for $x \ge 0$, which holds for any degree distributions (see [KM09] for a related approach).

3 Threshold saturation in the coloring of random graphs

The purpose of this chapter is to investigate threshold saturation for random CSPs. We will concentrate on random Q-COL for various reasons: it is well studied, it exhibits a rich set of thresholds and it exhibits a phase where the problem is believed to be hard (unlike XOR-SAT for example). These traits are shared with random K-SAT, and much of what we present can be readily applied for that and other random CSPs. However, there are a number of differences between coloring and formula satisfiability which we will mention along the way.

We proceed by deriving the location of the dynamic, condensation, freezing and colorability thresholds using the 1-RSB cavity method. The values of the thresholds are typically computed using population dynamics to represent random samples of messages. This approach can be generalized to spatially coupled coloring, working in a similar manner to density evolution in the case of LDPC codes. We observe the following:

- The dynamic threshold α_d of coupled random coloring moves to the condensation threshold α_c. This can be interpreted as the disappearance of clustering for $\alpha < \alpha_c$.
- The freezing phenomenon does not occur for $\alpha < \alpha_s$ in coupled random coloring. For coupled planted coloring, the freezing threshold moves up, based on a one-dimensional coupled recursion.
- The freezing threshold in fact coincides with the coloring threshold for coupled random coloring. This is suggested also by a proof that the coupled version of the RSB equations at $m = 0$ (i.e. the survey propagation equations) do not have non-trivial fixed points for $\alpha < \alpha_s$, whereas they do for $\alpha > \alpha_s$.

This chapter is organized as follows: Sections 3.1, 3.2 are introductory in nature and serve to set up the general 1-RSB framework of the cavity method. The equations for the special cases $m = 0$ and $m = 1$ are presented in Sections 3.3 and 3.5. Up to here everything is part of the well-established picture that emerged in statistical physics. We use the occasion to also write the equations for coupled systems. The main contributions are located in Sections 3.4, 3.6 and

3.7. Section 3.4 presents the lower bound on the freezing threshold of the coupled planted graph. Section 3.6 shows how for the SP threshold (freezing at $m = 0$) saturates to the coloring threshold for the coupled 1-RSB equations. Finally, Section 3.7 presents numerical evidence of the placement of thresholds at general values of the Parisi parameter m. Section 3.8 presents the conclusions and some open problems.

3.1 Preliminaries and the replica-symmetric approximation

Since we will only be concerned with coloring in this chapter, we simplify the notation in that we identify the binary constraints with the edges of the graph G; moreover the vertices of this graph will be denoted by u, v,..., rather than i, j, ... Colors will be identified with the integers in $[Q] = \{1,\dots,Q\}$. We will typically denote individual colors with q, q', etc.

A constraint (u,v) is characterized by the factor $\psi_{u,v} = 1-(1-e^{-\beta})\mathbb{1}(\sigma_u = \sigma_v)$. Note that in the zero temperature limit $\beta \to \infty$ the constraints become "hard" and the partition function Z simply counts the number of valid colorings.

We will work with the expected free entropy $\mathbb{E}\frac{1}{N}\log Z$. In this context, using hard constraints may result in expected free entropy equal to $-\infty$ because of rare events. It is customary in the physics literature to first take $N \to \infty$ and then $\beta \to \infty$. In simulations one can safely set $\beta = \infty$; in some places we find it more convenient to work at $\beta = \infty$, but the results can be easily extended to arbitrary β. To make things clear, we will make β explicit as a parameter whenever we work at nonzero temperature.

Methods of statistical physics have revealed that for this problem there are a number of distinct phases, for different values of α. Some of these predictions have been further substantiated by rigorous mathematical proofs, as mentioned in the introduction. We summarize here the picture that has emerged from the cavity method of statistical physics applied to coloring.

- **The RS phase.** For $\alpha < \alpha_d$ there is an exponential number of valid colorings, which are distributed in the whole space $[Q]^N$ with no clear structure underneath. In this phase an MCMC process wandering in the space of valid colorings by flipping a constant number of colors at each time step would mix very fast (in time polynomial in N).
- **The dynamic-RSB phase.** For $\alpha_d < \alpha < \alpha_c$, the number of valid colorings is still exponential, and moreover the exponential order can be obtained by analytic continuation in α from the RS phase. However the valid colorings are clustered, in the sense that an MCMC process will get stuck in a set of valid colorings that we call a *cluster*. Note that this is a very informal definition of clusters. In fact the clusters are likely to not be completely separated form each other, but rather they are connected by thin bridges that slow the dynamics. Both the number of clusters and the number of valid colorings in a cluster are exponential (in N), and a coloring chosen uniformly at random is with high probability likely to be found in one of the exponentially many clusters of size of

the highest exponential order.

- **The static-RSB phase.** For $\alpha_c < \alpha < \alpha_s$, there are indications that the number of solutions is still exponential, however the clusters of maximal size are sub-exponential in number, and they contain almost all valid colorings. The exponential order of the overall number of solutions is no longer given by the analytic continuation from the RS phase.
- **The uncolorable phase.** For $\alpha_s < \alpha$, the graph has with high probability no valid coloring.

The values α_d, α_c, α_s are the *dynamic*, *condensation* and *sat/unsat* phase transitions, respectively. We first give now a rough overview of how they emerge.

The regime where the computation of Z is easiest is when $0 < \alpha < 1$. In this case the graph is w.h.p. a forest and the computation is straightforward: we have that $Z = Q^N(1-(1-e^{-\beta})/Q)^{\alpha N/2}$. Thus in this regime, the free entropy density is given by

$$\Phi = \log Q + \frac{\alpha}{2}\log(1-\frac{1-e^{-\beta}}{Q}), \tag{3.1}$$

which is analytic in α. In fact, the free entropy continues to have this expression well beyond $\alpha = 1$; using the cavity method it is apparent that the first point of non-analiticity is α_c, the condensation phase transition. In particular this formula remains true in the dynamic RSB phase, which exhibits clustering. Recently, this fact has been established in a mathematically rigorous manner [BCOH$^+$14].

We now describe the RS equation of the cavity method for coloring. In the case of a tree, a message $\boldsymbol{\mu}^{u\to v}$ is a Q-tuple $\{\boldsymbol{\mu}^{u\to v}(q)\}_{q\in[Q]}$, whose meaning is the following. Remove node v from the graph and consider the remaining subtree containing u. Then $\boldsymbol{\mu}^{u\to v}(q)$ represents the proportion of colorings of that subtree in which u is colored with q. The messages are thus elements of the $Q-1$-dimensional simplex, which we denote by $\boldsymbol{\Delta}_Q$. We will refer to elements of $\boldsymbol{\Delta}_Q$ as *marginals*, and we will use bold face font for them throughout this chapter. We denote the vertices of $\boldsymbol{\Delta}_Q$ by $\boldsymbol{\eta}_1,\dots,\boldsymbol{\eta}_Q$, so that $\boldsymbol{\eta}_q(q') = \delta_{qq'}$. The *uniform marginal* $\boldsymbol{\eta}_=$ is the one that assigns equal proportions to all colors, i.e. $\boldsymbol{\eta}_=(q) = 1/Q$. The recursive rule to compute the messages on a tree is given by

$$\boldsymbol{\mu}^{u\to v}(q) = \frac{\prod_{v'\in\partial u\setminus v}(1-(1-e^{-\beta})\boldsymbol{\mu}^{v'\to u}(q)}{z(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})}, \quad z(\{\boldsymbol{\mu}^i\}_i) = \sum_{q\in[Q]}\prod_i(1-(1-e^{-\beta})\boldsymbol{\mu}^i(q)). \tag{3.2}$$

From these messages one obtains the approximation to the local marginals of the Gibbs measure as

$$\boldsymbol{\mu}^u(q) = \frac{\prod_{v'\in\partial u}(1-(1-e^{-\beta})\boldsymbol{\mu}^{v'\to u}(q)}{z(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u})}. \tag{3.3}$$

Because colors can be permuted, the solution space exhibits color symmetry, so at least in the case of trees, all messages will be the equal to $\boldsymbol{\eta}_{=}$. We term this set of messages the *RS solution*. [1] Even when the graph is not a tree, the RS solution is still a valid one for the message passing equations in (3.2), though it might not be unique. The existence of other solutions sets in at the dynamic threshold α_d. Insight into this phenomenon is obtained using the one-step replica symmetry breaking method (1-RSB).

The Bethe formula gives the RS approximation of the free entropy

$$\Phi_{\mathrm{RS}}(\beta) = \frac{1}{N}\left\{\sum_u \log z(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u}) - \sum_{u\sim v}\log z_e(\boldsymbol{\mu}^{v\to u},\boldsymbol{\mu}^{u\to v})\right\}, \tag{3.4}$$

$$\text{with } z_e(\boldsymbol{\mu},\boldsymbol{\mu}') = 1-(1-e^{-\beta})\sum_{q\in[Q]}\boldsymbol{\mu}(q)\boldsymbol{\mu}'(q), \tag{3.5}$$

where $z(\cdot)$ is given in (3.2). Here $\sum_{u\sim v}$ is the sum over edges $\{u,v\}$ of the graph.

One can easily check that by plugging the RS fixed point in the above formula we retrieve (3.1).[2] Even though we did not really need them here, the two types of partition function components z and z_e will be useful when we introduce the 1-RSB approach.

3.2 The 1-RSB approach

Experimental evidence points to the fact that for $\alpha > \alpha_d$, valid colorings form clusters. There are at least two ways to think about these, and it is not yet rigorously shown that they correspond to the same structures.

- Consider the graph of valid colorings, and connect two colorings if they differ in $o(N)$ vertices. In this interpretation, clusters correspond to the connected components of this graph, or at least are internally highly connected subsets which are poorly connected between themselves. This way to picture clusters is the easier to formalize of the two, and parts of the picture have been made rigorous.

- Clusters correspond to fixed points of the message-passing equations (3.2). The size of each cluster would be given by the Bethe free entropy at that particular fixed point, which we call *internal free entropy* of the cluster. It is still helpful to think of the clusters as disjoint sets of valid colorings, in which case the internal free entropy would be just the logarithm of the number of colorings in the cluster. This picture is much harder to establish mathematically, but in practice it allows us to obtain numerical estimates for the number and size of clusters.

[1] The fact that the RS solution consists of all messages equal to the uniform marginal is not true in other types of random CSPs, such as k-SAT where each clause has a sign for every literal it contains.

[2] For problems that do not exhibit this type of spin symmetry such a closed form may not exist, and then the RS solution is estimated by population dynamics and plugged into (3.4).

To derive the 1-RSB equations, we will work with the second "picture" in mind. As N grows large and for fixed $\phi > 0$, the number of clusters with internal free entropy density around ϕ will typically be exponential, so let $\Sigma(\phi)$ be its exponential order, which we will call *complexity*. That $\Sigma(\phi)$ concentrates will naturally also be part of the assumptions.

3.2.1 Sampling clusters of the right size

Let us first describe the relationship between the total free entropy Φ_{RSB} and the complexity function. In our "picture" the number of clusters is an integer; so in case $\Sigma(\phi)$ is negative we consider that there are in fact no clusters at all with internal free entropy ϕ. One could think that in this case, the real value of $\Sigma(\phi)$ is $-\infty$, and let us assume that it is indeed so, in other words the $\Sigma(\phi)$ is either non-negative or $-\infty$. In this case we are able to write at finite N

$$e^{N\Phi_{\mathrm{RSB}}} \doteq \sum_{\text{clusters } C} e^{N\phi_C} \doteq \int_{\phi=0}^{\infty} e^{N(\phi+\Sigma(\phi))}\,\mathrm{d}\phi. \tag{3.6}$$

By taking the limit $N \to \infty$ we see that

$$\Phi_{\mathrm{RSB}} = \sup_{\phi\in[0,\infty)} \big(\phi + \Sigma(\phi)\big). \tag{3.7}$$

If we sampled from a distribution on clusters where each cluster, call it C, is weighed by its size i.e. by $e^{N\phi_C}$, then we would asymptotically almost surely sample a cluster of internal energy $\phi^\star$, where $\phi^\star$ is the value of ϕ that maximizes the supremum, and which we assume to be unique. Let us call such clusters *typical*. This distribution would be written down as a graphical model (here called the *1-RSB model*). Assuming that there are no large correlations [3], we would use message passing to compute the total and internal free energies. Let us call the messages of the 1-RSB model *meta-messages*, lest confusion arise between these and the much less complex RS-messages.

In practice, the situation is more complicated for the following reason. Because of the "picture" that we keep to the back of our minds, this distribution on clusters corresponds to a distribution on fixed points of the RS message passing equations. However, we are required to make a simplification of the way we do the meta-message passing, which we now briefly describe. It is computationally very expensive to simulate message passing on actual graphs in the 1-RSB model, because we would need graphs of very large size and because the meta-messages will be in themselves distributions on the $Q-1$-simplex. We are thus forced to choose a different approach, where we do not keep track of the messages going in and out of every node in the graph, but rather use the fact that the graph is random and so the neighborhood of a random node of the graph is asymptotically a tree. We then compute the distribution of meta-messages sent to a typical node whose neighborhood is chosen at random from the actual distribution of neighborhoods in a large graph. We have managed then to trade keeping meta-messages

[3] experimentally it was verified that on the meta-model such correlations do not happen for the values of α we are interested in

on a graph with keeping a distribution of meta-messages, i.e. working with distributions of distributions. In practice this will be dealt with using population dynamics.

However, this simplification comes at a price. When sampling the random graph, it might be the case that very rarely (exponentially rarely) clusters appear, which have a really high internal free energy. Then these high internal free energies, to which we would in the previous setting have assigned a complexity of $-\infty$, get a now a finite negative complexity. This unwanted negative complexity is able to change the supremum in (3.7) and the value of $\phi^\star$. What we want is to treat the case of negative $\Sigma(\phi)$ as if $\Sigma(\phi)$ were $-\infty$. To do this, we limit the scope of ϕ in (3.6) and (3.7) to the subset where $\Sigma(\phi)$ is non-negative, getting (3.7). Assuming the function Σ is convex, three cases arise:

- There exists $\phi^\star$ with $\Sigma(\phi^\star) > 0$ such that the supremum is attained at $\phi^\star$. In case the function Σ is differentiable, this means that $\frac{\mathrm{d}\Sigma}{\mathrm{d}\phi}|_{\phi^\star} = -1$; If we used the weighing of clustering by their size, as described before, we would sample clusters of internal energy $\phi^\star$. This case corresponds to the dynamic RSB phase, i.e. for $\alpha < \alpha_c$.
- There is no ϕ as above, but there are still ϕ for which $\Sigma(\phi)$ is positive. By running the population dynamics in the 1-RSB model we will effectively sample meta-messages that correspond to an internal energy ϕ with $\Sigma(\phi) < 0$, i.e. clusters that appear extremely rarely. In this case the real supremum $\phi^\star$ is attained at the value $\phi^\star$ for which $\Sigma(\phi^\star) = 0$, which we will need to find. This case corresponds to the static (condensed) RSB phase, i.e. for $\alpha_c < \alpha < \alpha_s$.
- For all ϕ, $\Sigma(\phi)$ is negative. Then the graph is uncolorable w.h.p., which happens for $\alpha > \alpha_s$. It may still be the case that clusters of solutions appear rarely, in a fashion governed by the now negative complexity function Σ.

There is in fact a method to compute the whole complexity curve, which proves especially useful in the static RSB phase. We have assumed so far a distribution of clusters in which all were weighted by their size. Looking at the exponential orders, all clusters of internal energy ϕ counted as $\phi + \Sigma(\phi)$, and we were seeking those that count most. Let us now imagine an experiment where clusters would count as $m\phi + \Sigma(\phi)$, where m is a parameter that we tune as we wish. Then the typical clusters that we sample from such a distribution would be those whose internal free entropy ϕ maximizes $m\phi + \Sigma(\phi)$. Changing the parameter m, which we will call *the Parisi parameter*, enables us to sample clusters of various sizes and compute the complexity for various values of ϕ.

- In the dynamic RSB phase ($\alpha_d < \alpha < \alpha_c$), using $m^\star = 1$ will ensure that we sample clusters of size $\phi^\star$.
- In the static RSB phase ($\alpha_c < \alpha < \alpha_s$), however, using $m = 1$ would sample clusters that appear very rarely; the correct value $m^\star \in (0,1)$ is such that those clusters of internal entropy $\phi^\star$ are sampled.

- Choosing $m = 0$ allows us to not take the size of clusters into consideration when sampling. The clusters will be sampled uniformly, but there is a value of the internal entropy for which the clusters are the most numerous, so asymptotically almost surely we will sample clusters with exactly that internal entropy. Thus choosing $m = 0$ enables us to compute to the highest value of Σ. If this value is positive, the graph is w.h.p. colorable, while if it is negative, it is w.h.p. uncolorable. Thus the value of α for which the maximum value of $\Sigma(\phi)$ is 0 marks the colorability threshold α_s.

3.2.2 Meta-message passing equations

We now exhibit the general way in which to sample clusters for different values of parameter m. The right distribution on clusters C is given by

$$\mathbb{\nu}^{(m)}(C) = \frac{e^{Nm\phi_C}}{Z^{(m)}_{\mathrm{RSB}}}, \tag{3.8}$$

where $Z^{(m)}_{\mathrm{RSB}}$ is a normalization factor. It can be easily checked that for $m^\star$ this is related to the total free entropy: $Z^{(m^\star)}_{\mathrm{RSB}} = e^{Nm^\star \Phi_{\mathrm{RSB}}}$. For this reason we introduce the notation $\Phi^{(m)}_{\mathrm{RSB}} = \frac{1}{Nm} Z^{(m)}_{\mathrm{RSB}}$.

We use our asserted equivalence of clusters with fixed points of message-passing (the "picture"). We begin first by deriving the meta-message-passing equations on a fixed graph. The clusters are characterized by the set of fixed point messages $\{\boldsymbol{\mu}^{u\to v}\}$ and the internal entropy ϕ_C is in fact given by the Bethe formula, i.e. (3.4). Writing down explicitly the internal entropy and the fixed point constraints, (3.8) becomes

$$\mathbb{\nu}^{(m)}(\{\boldsymbol{\mu}^{u\to v}\}) = \frac{1}{Z^{(m)}_{\mathrm{RSB}}} \frac{\prod_u z(\{\boldsymbol{\mu}^{v\to u}\}_{v\in\partial u})^m}{\prod_{u\sim v} z_e(\boldsymbol{\mu}^{v\to u}, \boldsymbol{\mu}^{u\to v})^m} \prod_{u\to v} \delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})), \tag{3.9}$$

where $\mathbf{f}\colon \boldsymbol{\Delta}_Q^* \to \boldsymbol{\Delta}_Q$ is the message processing rule of (3.2).

There are three type of factors in (3.9), which involve sets of variables of the form $\{\boldsymbol{\mu}^{v\to u}\}_{v\in\partial u}$, $\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v} \cup \{\boldsymbol{\mu}^{u\to v}\}$ and $\{\boldsymbol{\mu}^{v\to u}, \boldsymbol{\mu}^{u\to v}\}$, respectively. In the first phase, let us first write down meta-message passing rules that use joint marginals on pairs of messages that travel on the same edge in opposite direction. The meta-message passing rules will be derived in such a way that they are exact when the underlying graph is a tree, so let us assume for now that the graph is indeed a tree. Thus, $\mathbb{\nu}^{u\to v}(\boldsymbol{\mu}^{u\to v}, \boldsymbol{\mu}^{v\to u})$ is the meta-message from u to v, representing the marginal on the pair $(\boldsymbol{\mu}^{u\to v}, \boldsymbol{\mu}^{v\to u})$ when all the fixed point constraints at v and the z factor of incoming messages into v are removed.

We thus have

$$\mathbb{\nu}^{u\to v}(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u}) \simeq \int_{\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v}} \int_{\{\boldsymbol{\mu}^{u\to v'}\}_{v'\in\partial u\setminus v}} \prod_{v'\in\partial u\setminus v} \mathrm{d}\mathbb{\nu}^{v'\to u}(\boldsymbol{\mu}^{v'\to u},\boldsymbol{\mu}^{u\to v'})\cdot$$
$$\cdot\, z(\{\boldsymbol{\mu}^{w\to u}\}_{w\in\partial u})^m\, z_e(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})^{-m} \prod_{w\in\partial u} \delta(\boldsymbol{\mu}^{u\to w} = \mathbf{f}(\{\boldsymbol{\mu}^{w'\to u}\}_{w'\in\partial u\setminus w})), \tag{3.10}$$

where the sign $\simeq$ is used to denote the equality of the two measures on the two sides, up to a normalization constant which is chosen so that the left side is a probability measure.

It is possible to simplify these meta-messages by marginalizing further over the messages running in opposite directions, i.e. $\mathbb{\nu}^{u\to v}(\boldsymbol{\mu}^{u\to v}) = \int \mathrm{d}\boldsymbol{\mu}^{v\to u}\mathbb{\nu}^{u\to v}(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})$. This is because we can get rid of $\boldsymbol{\mu}^{v\to u}$ in the formula above by observing that

$$\begin{aligned} &z(\{\boldsymbol{\mu}^{w\to u}\}_{w\in\partial u})\, z_e^{-1}(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})\delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})) = \\ &= \frac{\sum_{q\in[Q]}\prod_{w\in\partial u\setminus v}(1-\boldsymbol{\mu}^{w\to u}(q))(1-\boldsymbol{\mu}^{v\to u}(q))}{1-\sum_{q\in[Q]}\boldsymbol{\mu}^{u\to v}(q)\boldsymbol{\mu}^{v\to u}(q)}\delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})) \\ &= z(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})\delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})). \end{aligned} \tag{3.11}$$

The other deltas that appear are neutralized by integration over the outgoing messages other than $\boldsymbol{\mu}^{u\to v}$, so that we obtain the much simpler form

$$\mathbb{\nu}^{u\to v}(\boldsymbol{\mu}^{u\to v}) \simeq \int \prod_{v'\in\partial u\setminus v} \mathrm{d}\mathbb{\nu}^{v'\to u}(\boldsymbol{\mu}^{v'\to u}) z(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})^m \delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})). \tag{3.12}$$

The total free entropy density Φ_{RSB} can be computed using the Bethe functional of the meta-messages where the Parisi parameter takes the value $m^\star$. More generally, the formula for $\Phi_{\mathrm{RSB}}^{(m)}$ is

$$\begin{aligned} \Phi_{\mathrm{RSB}}^{(m)} &= \frac{1}{Nm}\sum_u \log\int \prod_{v\in\partial u} \mathrm{d}\mathbb{\nu}^{v\to u}(\boldsymbol{\mu}^{v\to u}) z(\{\boldsymbol{\mu}^{v\to u}\}_{v\in\partial u})^m - \\ &- \frac{1}{Nm}\sum_{u\sim v}\log\int \mathrm{d}\mathbb{\nu}^{u\to v}(\boldsymbol{\mu}^{u\to v})\,\mathrm{d}\mathbb{\nu}^{v\to u}(\boldsymbol{\mu}^{v\to u}) z_e(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})^m. \end{aligned} \tag{3.13}$$

The internal entropy ϕ can be obtained in principle in two ways. If we had the possibility, we could sample a cluster/message-passing-fixed point at random from the distribution (3.9), and then compute the Bethe functional of that fixed point. This being beyond our means, we resort to something slightly different, namely compute the Bethe entropy using local pieces of typical fixed points: take each node, sample incoming messages using our graphical meta-model and compute the expected $\log z$ of those messages; similarly, sample messages that

travel in opposite directions on an edge and compute the expected $\log z_e$. We would obtain

$$\phi^{(m)} = \frac{1}{Nm}\sum_u \frac{\int \prod_{v\in\partial u} \mathrm{d}\mathbb{v}^{v\to u}(\boldsymbol{\mu}^{v\to u}) z(\{\boldsymbol{\mu}^{v\to u}\}_{v\in\partial u})^m \log z(\{\boldsymbol{\mu}^{v\to u}\}_{v\in\partial u})^m}{\int \prod_{v\in\partial u} \mathrm{d}\mathbb{v}^{v\to u}(\boldsymbol{\mu}^{v\to u}) z(\{\boldsymbol{\mu}^{v\to u}\}_{v\in\partial u})^m} - \\ - \quad \frac{1}{Nm}\sum_{u\sim v} \frac{\int \mathrm{d}\mathbb{v}^{u\to v}(\boldsymbol{\mu}^{u\to v})\,\mathrm{d}\mathbb{v}^{v\to u}(\boldsymbol{\mu}^{v\to u}) z_e(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})^m \log z_e(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})^m}{\int \mathrm{d}\mathbb{v}^{u\to v}(\boldsymbol{\mu}^{u\to v})\,\mathrm{d}\mathbb{v}^{v\to u}(\boldsymbol{\mu}^{v\to u}) z_e(\boldsymbol{\mu}^{u\to v},\boldsymbol{\mu}^{v\to u})^m}. \tag{3.14}$$

There is an alternative way to obtain the formula above. Let us view $\Phi^{(m)}_{\mathrm{RSB}}$ and $\phi^{(m)}$ as functions of m. Then taking the derivative of $m\Phi^{(m)}_{\mathrm{RSB}} = m\phi^{(m)} + \Sigma(\phi^{(m)})$ we get

$$\frac{\mathrm{d}}{\mathrm{d}m}\left(m\Phi^{(m)}_{\mathrm{RSB}}\right) = \phi^{(m)} + m\frac{\mathrm{d}\phi^{(m)}}{\mathrm{d}m} + \left.\frac{\mathrm{d}\Sigma}{\mathrm{d}\phi}\right|_{\phi^{(m)}} \frac{\mathrm{d}\phi^m}{\mathrm{d}m} = \phi^{(m)},$$

where in the last step used that $\left.\frac{\mathrm{d}\Sigma}{\mathrm{d}\phi}\right|_{\phi^{(m)}} = -m$. We can then check (3.14) by taking the derivative w.r.t. m in (3.13).

Values for the complexity function can then be obtained using the relation

$$\Sigma(\phi^{(m)}) = m(\Phi^{(m)}_{\mathrm{RSB}} - \phi^{(m)}).$$

All the meta-messages are elements of $\mathbb{M}(\mathbf{\Delta}_Q)$, the set of probability measures on $\mathbf{\Delta}_Q$, usually shortened to $\mathbb{M}$. We will typically denote the elements of $\mathbb{M}$ by blackboard-type fonts.

3.2.3 The mean-field form

We analyze equations such as (3.12) by considering meta-messages as random variables drawn from a distribution that we need to find.

To be able to write the distributional equation in a compact form, let $\mathbb{F}^{(m)} : \mathbb{M}^d \to \mathbb{M}$ be the 1-RSB message passing rule with Parisi parameter m defined by

$$\mathbb{F}^{(m)}(\mathbb{v}_1,\dots,\mathbb{v}_d)(B) = \frac{\int_{\mathbf{\Delta}_Q^d} \left(\prod_{i=1}^d \mathrm{d}\mathbb{v}_i(\boldsymbol{\mu}_i)\right) \mathbb{1}\{\mathbf{f}(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)\in B\} z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m}{\int_{\mathbf{\Delta}_Q^d} \left(\prod_{i=1}^d \mathrm{d}\mathbb{v}_i(\boldsymbol{\mu}_i)\right) z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m}, \tag{3.15}$$

for any measurable subset B of $\mathbf{\Delta}_Q$.

We are seeking distributional fixed points that are invariant under the 1RSB message passing rule, i.e., probability measures $\mathscr{P}$ on $\mathbb{M}$ that satisfy

$$\mathscr{P} = \mathbb{E}_d \int \mathrm{d}\mathscr{P}(\mathbb{v}_1)\cdots\mathrm{d}\mathscr{P}(\mathbb{v}_d)\delta_{\mathbb{F}^{(m)}(\mathbb{v}_1,\dots,\mathbb{v}_d)}. \tag{3.16}$$

In the case of general m, this fixed point equation is investigated using a two-level population

dynamics approach, in which we approximate the meta-messages by a large pool of sample marginals, themselves being collections of samples from $\mathbf{\Delta}_Q$. The fixed point equation (3.16) can be significantly simplified in the cases $m = 1$ and $m = 0$, in which case one level of population dynamics will suffice. We will treat these cases separately.

The averaged total RSB free energy density and the averaged free energy density per cluster can be computed using[4]

$$\Phi^{(m)}_{\mathrm{RSB}} = \frac{1}{m}\mathbb{E}_d \int \prod_{i=1}^{d} \mathrm{d}\mathscr{P}(\mathbb{v}_i) \log \int \prod_{i=1}^{d} \mathrm{d}\mathbb{v}_i(\boldsymbol{\mu}_i) z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m - \\ - \frac{\alpha}{2m} \int \mathrm{d}\mathscr{P}(\mathbb{v}_1)\,\mathrm{d}\mathscr{P}(\mathbb{v}_2) \log \int \mathrm{d}\mathbb{v}_1(\boldsymbol{\mu}_1)\,\mathrm{d}\mathbb{v}_2(\boldsymbol{\mu}_2) z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2)^m. \tag{3.17}$$

$$\phi^{(m)} = \frac{1}{m}\mathbb{E}_d \int \prod_{i=1}^{d} \mathrm{d}\mathscr{P}(\mathbb{v}_i) \frac{\int \prod_{i=1}^{d} \mathrm{d}\mathbb{v}_i(\boldsymbol{\mu}_i) z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m \log z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m}{\int \prod_{i=1}^{d} \mathrm{d}\mathbb{v}_i(\boldsymbol{\mu}_i) z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m} - \\ - \frac{\alpha}{2m} \int \mathrm{d}\mathscr{P}(\mathbb{v}_1)\,\mathrm{d}\mathscr{P}(\mathbb{v}_2) \frac{\int \mathrm{d}\mathbb{v}_1(\boldsymbol{\mu}_1)\,\mathrm{d}\mathbb{v}_2(\boldsymbol{\mu}_2) z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2)^m \log z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2)^m}{\int \mathrm{d}\mathbb{v}_1(\boldsymbol{\mu}_1)\,\mathrm{d}\mathbb{v}_2(\boldsymbol{\mu}_2) z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2)^m}. \tag{3.18}$$

In the case of spatially coupled graphs, there will be a distribution $\mathscr{P}_z$ at each position along the chain. The fixed point equation becomes

$$\mathscr{P}_z = \mathbb{E}_d \int \mathrm{d}\mathscr{Q}_z(\mathbb{v}_1)\cdots\mathrm{d}\mathscr{Q}_z(\mathbb{v}_d)\delta_{\mathbb{F}^{(m)}(\mathbb{v}_1,\dots,\mathbb{v}_d)}, \quad \text{where } \mathscr{Q}_z = \frac{1}{W^2}\sum_{w=0}^{W-1}\sum_{w'=0}^{W-1}\mathscr{P}_{z+w-w'}, \tag{3.19}$$

and where for z outside $\{1,\dots,L\}$ we set $\mathscr{P}_z$ to be concentrated on the meta-marginal with all mass on $\boldsymbol{\eta}_=$. This latter condition arises at the boundary, where some edges are missing. These are precisely the ones connecting to positions outside $\{1,\dots,L\}$. The meta-marginal $\mathbb{v}_=$ with all mass on $\boldsymbol{\eta}_=$ serves as a neutral element for $\mathbb{F}^{(m)}$, in the sense that $\mathbb{F}^{(m)}(\mathbb{v}_1,\dots,\mathbb{v}_d,\mathbb{v}_=) = \mathbb{F}^{(m)}(\mathbb{v}_1,\dots,\mathbb{v}_d)$. It then makes sense to ascribe the value $\mathbb{v}_=$ to all meta-messages coming in from positions outside $\{1,\dots,L\}$, justifying the value of $\mathscr{P}_z$ at these positions. This enables us to write a formula valid at all positions $z \in \{1,\dots,L\}$.

3.2.4 Freezing

Let $\mathring{\mathbb{M}}$ be the subset of $\mathbb{M}$ that consists of all probability measures on $\mathbf{\Delta}_Q$ that assign nonzero mass to the set $\{\boldsymbol{\eta}_1,\dots,\boldsymbol{\eta}_Q\}$. In other words, $\mathring{\mathbb{M}}$ contains all possible meta-marginals that assign a nonzero probability to the vertices of the simplex. We say that *freezing occurs* if there is a fixed point $\mathscr{P}$ of (3.16) such that $\mathscr{P}(\mathring{\mathbb{M}}) > 0$. Note that we can only talk about freezing in the zero temperature case ($\beta = +\infty$), for positive temperature freezing cannot occur. The *freezing threshold* $\alpha_f^{(m)}$ is defined as the point at which freezing sets in. Note that in the RS phase there is no freezing since the only fixed point of (3.16) is the one concentrated on the meta-marginal that has all its mass on $\boldsymbol{\eta}_=$.

[4] By an abuse of notation we shall be calling these averages also $\Phi^{(m)}_{\mathrm{RSB}}$ and $\phi^{(m)}$.

As we defined it, freezing is a property of Equation (3.16), and so one can have a freezing threshold for each value of the Parisi parameter m. Naturally, there is a "real" value of the freezing threshold, and it is given by α satisfying $\alpha = \alpha_f^{(m^\star(\alpha))}$.

According to the clustering "picture", freezing means that for each cluster there is a number of vertices that always keep the same color in all colorings of that cluster. In fact, the more usual way to define freezing is geometrical, in the sense that under a valid coloring a variable is said to be frozen if it cannot be changed by altering the coloring a small number ($o(N)$) of vertices at a time, repeatedly. This definition has the advantage that it makes no reference to clusters and so is used in proofs. The prediction of $\alpha_f^{(1)}$ can be verified rigorously under the planted model. However, no rigorous results are known for different values of m.

3.3 The special case $m = 1$

In order to simplify equation (3.12) we need some way to neutralize the $z(\cdot)$ that is integrated over. This disappears naturally if instead of meta-messages ν we use Q meta-messages $\mathbb{p}_q$, defined by

$$\mathbb{p}_q^{u\to v}(\boldsymbol{\mu}^{u\to v}) = q\boldsymbol{\mu}^{u\to v}(q)\mathbb{v}^{u\to v}(\boldsymbol{\mu}).$$

Because of color symmetry, $\mathbb{p}^{u\to v}$ is a probability distribution. To simplify notation, we introduce the quantity $\pi(q'|q) = \frac{1-(1-e^{-\beta})\delta_{qq'}}{Q-(1-e^{-\beta})}$. We have

$$z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_n)\mathbf{f}(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_n)(q) = (Q-1+e^{-\beta})\prod_i\sum_{q'}\pi(q'|q)\boldsymbol{\mu}_i(q').$$

Then the meta-message equations become

$$\begin{aligned}
\mathbb{p}_q^{u\to v}(\boldsymbol{\mu}^{u\to v}) &\simeq \int \prod_{v'\in\partial u\setminus v} \mathrm{d}\mathbb{v}^{u\to v}(\boldsymbol{\mu}^{u\to v})\boldsymbol{\mu}^{u\to v}(q)z(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})\delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})) \\
&\simeq \int \prod_{v'\in\partial u\setminus v} \mathrm{d}\mathbb{p}^{u\to v}(\boldsymbol{\mu}^{u\to v})\sum_{q'}\pi(q'|q)\boldsymbol{\mu}_i(q')\delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v})) \\
&\simeq \int \left(\prod_{v'\in\partial u\setminus v}\sum_{q'}\pi(q'|q)\,\mathrm{d}\mathbb{p}_{q'}^{u\to v}(\boldsymbol{\mu}^{u\to v})\right)\delta(\boldsymbol{\mu}^{u\to v} = \mathbf{f}(\{\boldsymbol{\mu}^{v'\to u}\}_{v'\in\partial u\setminus v}))
\end{aligned}$$

Note that there is no need for a normalization factor, because the final result is already a probability distribution. We write now the mean-field form for the above, assuming a distribution $\mathscr{P}_q(\mathbb{p}_q)$ on the meta-messages $\mathbb{p}_q$. However, because of the simple form taken by the equations, it is enough to work with the averages $\mathbb{p}_q(\boldsymbol{\mu}) = \int \mathrm{d}\mathscr{P}_q(\mathbb{p}_q)\mathbb{p}_q(\boldsymbol{\mu})$. Then we obtain

$$\mathbb{p}_q(\boldsymbol{\mu}) = \mathbb{E}_d\int\left(\prod_{i=1}^d\sum_{q'}\pi(q'|q)\,\mathrm{d}\mathbb{p}_{q'}(\boldsymbol{\mu}_i)\right)\delta(\boldsymbol{\mu} = \mathbf{f}(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)). \tag{3.20}$$

Because of the inherent symmetry, all $\mathbb{p}_q$ for $q = 1,\dots,Q$ are in fact the same up to permutation of colors in the following sense: if $\tilde{\pi}$ is a permutation on $[Q]$, then $\mathbb{p}_{\tilde{\pi}(q)} = \mathbb{p}_q \circ \tilde{\pi}^{-1}$.

3.3.1 Reconstruction on trees

A model for which equations (3.20) are known to be exact is that of *reconstruction on trees*. Suppose we are given a rooted tree T of depth D, with all vertices on level D being already assigned a fixed color. Let $\boldsymbol{\mu}(T)$ be the marginal over the root of the Gibbs measure over colorings of T. Now we randomize the tree, in that we start a Poisson-Galton-Watson process of parameter α at the root and stop the generation when it reaches depth D. Then we color the root with a fixed color q_0, and color all nodes recursively, so that if a parent is colored q then its children will be independently colored with colors q' drawn from $\pi(q'|q)$.We subsequently erase all colors except the ones on level D. We now ask what is the distribution of $\boldsymbol{\mu}(T)$ as $D \to \infty$ and T is sampled according to the model above. The answer turns out to be exactly $\mathbb{p}_{q_0}$ [MM06].

The reconstruction model can be easily extended to the spatially coupled scenario. Each node has an additional label, besides the color. This is a number between 1 and L, corresponding to the position. The root is assigned a fixed position z_0, and the positions z' of children are sampled independently once the parent position z is known by setting $z' = z + w - w'$, with w, w' drawn i.i.d. uniformly from $\{0,\dots,W-1\}$. Nodes with positions outside $\{1,\dots,L\}$ are deleted from the graph. Then the equivalent of (3.20) is

$$\mathbb{p}_{q;z}(\boldsymbol{\mu}) = \mathbb{E}_d \int \left(\prod_{i=1}^{d} \sum_{q'} \pi(q'|q)\, \mathrm{d}\mathbb{q}_{q';z}(\boldsymbol{\mu}_i) \right) \delta(\boldsymbol{\mu} = \mathbf{f}(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)), \tag{3.21}$$

$$\text{with} \quad \mathbb{q}_{q';z} = \sum_{w,w'=0}^{W-1} \mathbb{p}_{q';z+w-w'}. \tag{3.22}$$

and $\mathbb{p}_{q;z} = \boldsymbol{\eta}_=$ for all z outside $\{1,\dots,L\}$.

3.3.2 Free entropies and complexity

Because of color symmetry $\int \mathrm{d}\mathbb{v}(\boldsymbol{\mu})\boldsymbol{\mu}(q) = 1/Q$ for any q. Hence from (3.17) we obtain

$$\int \mathrm{d}\mathbb{v}_1(\boldsymbol{\mu}_1)\cdots\mathrm{d}\mathbb{v}_d(\boldsymbol{\mu}_d)\, z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d) = Q(1-\frac{1-e^{-\beta}}{Q})^d$$

$$\int \mathrm{d}\mathbb{v}_1(\boldsymbol{\mu}_1)\,\mathrm{d}\mathbb{v}_2(\boldsymbol{\mu}_2)\, z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2) = 1-\frac{1-e^{-\beta}}{Q}.$$

Using this we immediately see that the total RSB free entropy at $m = 1$ is equal to the RS free entropy:

$$\Phi^{(1)}_{\mathrm{RSB}} = \log Q + \frac{\alpha}{2}\log(1 - \frac{1-e^{-\beta}}{Q}).$$

The free entropy per cluster is obtained from (3.18) by expanding $z(\cdot)$ and $z_e(\cdot)$ and then replacing $\mathscr{P}(\mathbb{p})\mathbb{p}(\boldsymbol{\mu})\boldsymbol{\mu}(q)$ with $\mathbb{p}_q(\boldsymbol{\mu})$:

$$\begin{aligned}\phi^{(1)} = \mathbb{E}_d \frac{1}{Q}\sum_q \int \prod_{i=1}^{d}\left(\sum_{q'}\pi(q'|q)\,\mathrm{d}\mathbb{p}_{q'}(\boldsymbol{\mu}_i)\right)\log z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d) - \\ - \frac{\alpha}{2Q}\sum_{q,q'}\pi(q|q')\int \mathrm{d}\mathbb{p}_q(\boldsymbol{\mu}_1)\,\mathrm{d}\mathbb{p}_{q'}(\boldsymbol{\mu}_2)\log z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2).\end{aligned}$$

3.3.3 Freezing phenomenon

For $m = 1$, freezing can be characterized by a particularly simple equation. We need only keep track of the mass of $\mathbb{p}_q$ sitting at $\boldsymbol{\eta}_q$, since by construction of the measure $\mathbb{p}$ there will be no mass at any of the vertices of $\boldsymbol{\Delta}_Q$ except $\boldsymbol{\eta}_q$. Because of color symmetry, this mass will be in fact independent of q, so we denote it by a number x.

To derive the equation in terms of x, let us perform a random experiment. Given d, $q_1,\dots,q_d$, sample d independent random variables $A_1,\dots,A_d$ as follows: set $A_i = q_i$ with probability x and $A_i = *$ with probability $1-x$. Now we draw d from Poisson(α), and draw $q_1,\dots,q_d$ independently and uniformly from $\{1,\dots,Q\}\setminus\{q_0\}$, where q_0 is some color fixed apriori. Then from (3.21) we deduce

$$x = \mathbb{E}_d\mathbb{E}_{q_1,\dots,q_d}\Pr\left[\text{all in } \{1,\dots,Q\}\setminus\{q\} \text{ are represented among } A_1,\dots,A_d | q_1,\dots,q_d\right].$$

This can be simplified as follows. The same distribution on $d, q_1,\dots,q_d$ can be obtained in a different way. We assume for simplicity that $q_0 = Q$. For each $q \in \{1,\dots,Q-1\}$, we draw $d_q \sim \mathrm{Poisson}(\frac{d}{Q-1})$ independently. Set $d = d_1 + \dots d_{Q-1}$ and let $q_1,\dots,q_d$ be a random shuffle of d_1 times color 1, d_2 times color 2 and so on. In this setting, the probability event that all colors $\{1,\dots,Q-1\}$ are represented splits into independent factors, one for each color. Each factor is in fact the probability that $A_1,\dots,A_{d_i}$ are not all stars, so it is equal to $1-(1-x)^d_i$. Averaging this with respect to the Poisson distribution of d_i, we obtain $1-e^{\alpha x/(Q-1)}$, so finally we conclude that

$$x = \left(1 - e^{\alpha x/(Q-1)}\right)^{Q-1}. \tag{3.23}$$

The freezing threshold $\alpha_f^{(m=1)}$ is then the infimum over values of α for which the above equation has a solution in $(0,1)$. As we will see next, the freezing equation at $m = 1$ has a rigorous interpretation in case we are studying coloring on a planted random graph. The

equivalent equation for spatially coupled coloring is also treated in the next section.

3.4 Freezing on the planted graph

There are alternative ways to introduce freezing on a graph, and in fact these correspond to what was originally called freezing in the literature. The freezing phenomenon we discussed up to now is sometimes called *rigidity* [ZK07]. There is no consensus yet on the formal definition of freezing, which depends on the notion of "cluster of solutions" employed.

In [Mol12], the author takes the following approach: consider the auxiliary graph whose vertices are valid colorings and two valid colorings are connected if and only if they lie at Hamming distance at most ℓ, which is an arbitrary function of N. A vertex u is said to be ℓ-frozen with respect to some valid coloring σ if $\tau_u = \sigma_u$ for all valid colorings τ in the connected component containing σ of the auxiliary graph. For sufficiently large Q, it is shown that w.h.p., when picking a valid coloring uniformly at random (i) if $\alpha < \alpha_f^{(m=1)}$ then at most $o(N)$ vertices are $\omega(N)$ frozen, and (ii) if $\alpha > \alpha_f^{(m=1)}$ then a fixed proportion of vertices is $\Theta(n)$-frozen, while also a fixed proportion of vertices if not $\omega(N)$-frozen. Here $\omega(N)$ is a function tending to infinity arbitrarily slowly. The fixed proportion of vertices mentioned is related to a solution of (3.23). Thus, a property on the graph was found that has a threshold exactly at $\alpha_f^{m=1}$.

The property was first proved in a different setting, that of a *planted graph*. Based on a result of [ACO08], under certain conditions the planted graph model is "equivalent" to the original Erdős-Rényi model; thus certain properties valid for the planted model also hold for the original one. In [BCOH$^+$14] it was shown that for Q-COL (where Q is large enough) this equivalence is valid up to the condensation threshold α_c.

While the much of the argument provided in [Mol12] is quite technical, the lower bound on $\alpha_f^{(m=1)}$ (given by point (i) above) for the planted model is in fact quite easy to understand. We first present the planted model, show why $\alpha_f^{(m=1)}$ is indeed a lower bound on the real freezing threshold on planted graphs, and then generalize the argument to the coupled version of planted graphs.

The idea of the planted model is to generate a random graph that contains a specific assignment $\underline{\sigma}$ as a valid coloring, and otherwise looks similar to a random graph.

The planted graph $G(\underline{\sigma})$ is the random graph generated in the following manner. Let E be the set of *potential edges*, defined as all pairs (u, v) such that $\sigma_u \neq \sigma_v$. Add each potential edge to the graph independently with probability $\frac{\alpha Q}{N(Q-1)}$. This ensures that $\underline{\sigma}$ remains a valid coloring. If the assignment $\underline{\sigma}$ has the property that the number of vertices of each color is roughly the same, then the number of neighbors of a vertex u picked at random is asymptotically Poisson(α). There are $Q-1$ possible colors that each of these neighbors of u can have, and so the number of neighbors of u of each color is asymptotically Poisson($\frac{\alpha}{Q-1}$).

Let $T(N)$ be a function of N that tends to infinity arbitrarily slowly. We say that a vertex v is $T(N)$-*free* if there is an assignment $\underline{\tau}$ which differs from $\underline{\sigma}$ only on a neighborhood of size $T(N)$ around v. We show the following lemma.

Lemma 21. *If $\alpha < \alpha_f^{(m=1)}$ then with high probability a fraction of $1-o(N)$ of all vertices in the graph $G(\underline{\sigma})$ are $T(N)$-free.*

Note that this is equivalent to the lower bound result in [Mol12], but we present a proof that can be generalized to the coupled version of planted graphs.

Proof. The neighborhood $\mathcal{N}_v(G(N,\alpha;\sigma);t)$ is the subgraph induced by nodes at distance at most t from v. When t is a constant, this neighborhood will be w.h.p. a tree, according to Lemma 58 in the Appendix D.1. Moreover, from the same lemma we obtain that $\mathcal{N}_v(G(N,\alpha;\sigma);t)$ converges weakly to the Poisson Galton-Watson process $\mathcal{T}(\alpha,t)$ defined below.

The random $\mathcal{T}(\alpha,t)$ is obtained in the following manner. We color the root uniformly at random. For each leaf u colored q, we expand the tree by adding a number of children. For each color $q' \in [Q]\setminus\{q\}$ we draw a number $d_{u,q'}$ from Poisson$(\alpha/(Q-1))$ and create $d_{u,q'}$ new children of u and color them using color q'.

Let x_t be the probability of the event E_t that there exists no coloring $\underline{\sigma}'$ that coincides with $\underline{\sigma}$ on the set of nodes at distance t from the root, but on the root v itself, they differ. Clearly $x_0 = 1$, since in that case the color of the root is fixed. To compute recursively x_{t+1} from x_t we observe the following. The event $\overline{E}_{t+1}$ happens if and only if there is a coloring $\underline{\sigma}'$ such that all neighbors u of the root v colored $\underline{\sigma}'$ can change color in the subtree rooted at u subject to the condition that nodes situated at distance t from u keep their colors. Note, however, that the subtree rooted at any node of the tree is distributed as the entire tree. The probability that a child of the root be able to change color in the subtree with itself as root is then given by $1-x_t$.

We say that a color q *shifts* if the root has the property that all its children colored q can change color in the manner described above. The probability that q shifts is given by

$$\sum_{d\geq 0}\frac{1}{d!}e^{-\frac{\alpha}{Q-1}}\left(\frac{\alpha}{Q-1}\right)^d(1-x_t)^d = e^{-\frac{\alpha}{Q-1}x_t}. \tag{3.24}$$

Since there are $Q-1$ possible colors that can shift and they are independent, the probability that there exists none that shifts is given by

$$x_{t+1} = \left(1-e^{-\frac{\alpha}{Q-1}x_t}\right)^{Q-1}. \tag{3.25}$$

Since $\alpha < \alpha_f^{(m=1)}$, there are no fixed points of the equation above except 0. Thus, one can obtain an arbitrarily small x_t by increasing t. Since in the graph setting $x_t(1+o(1))$ translates

to the ratio of t-frozen vertices, it follows that there is no linear proportion of $T(N)$-frozen vertices. □

3.4.1 Freezing on the planted coupled graph

The planted coupled graph is defined as follows. There is a number L of positions, indexed from 1 to L, and N nodes are located at each position. As in the uncoupled case, we are given a planted coloring $\underline{\sigma}$. It is useful to think of the position of a vertex along the chain as also "planted". Thus, each vertex has two labels, a color and a position.

Edges are allowed only between nodes colored differently under $\underline{\sigma}$ and which additionally satisfy the condition that their positions satisfy the window constraint. Each such edge will appear in the graph independently with probability $\frac{\alpha Q(W-|w|)}{NW^2(Q-1)}$, where w is the distance in positions between the nodes. We call a coloring *balanced* if at each position $z \in [L]$ and the number of nodes colored with each color is $N/Q + o(N^{2/3})$.

We define the Galton-Watson tree $\mathscr{T}_z^{\text{coup}}(\alpha, L, W; t)$ in the following way. Create a root and label it with position z and a random color. At each generation, a leaf u labeled q' and z' generates for each $q'' \neq q'$ and $z'' \in \{z'-W+1, \dots, z'+W-1\} \cap \{1, \dots, L\}$ a number of children drawn from Poisson($\frac{\alpha(W-|z-z''|)}{W^2(Q-10}$).

Because of Lemma 59 in Appendix D.1, we only need examine the tree and not the random graph. We now maintain a separate probability $x_{t,z}$ at each position, which represents the probability that the root of $\mathscr{T}_z^{\text{coup}}(\alpha, L, W; t)$ will shift when the nodes at distance d have their color fixed. We obtain the coupled recursion

$$x_{t+1,z} = \left(1 - e^{-\frac{\alpha}{Q-1}\frac{1}{W^2}\sum_{w,w'=0}^{W-1} x_{t,z+w-w'}}\right)^{Q-1} \tag{3.26}$$

where we fixed $x_{t,z}$ to 0 for z outside $\{1, \dots, L\}$.

This is a type of scalar coupled recursion, and it can be analyzed with the tools developed in [YJNP12]. One can immediately check that the uncoupled recursion (3.25) is a *scalar admissible system* is the sense of Definition 1 of [YJNP12], with $g(x) = x$ and $f(x, \alpha)$ given by the right-hand-side of the recursion.[5] The potential, according to Definition 2 of [YJNP12], is given by $U(x) = \frac{x^2}{2} - x + \int_0^x f(x, \alpha)\, \mathrm{d}x$, which can also be written as

$$U^{(\alpha)}(x) = \frac{x^2}{2} - x + \frac{Q-1}{\alpha} \sum_{q=1}^{Q-1} \frac{1}{q} \left(1 - e^{-\frac{\alpha x}{Q-1}}\right)^q. \tag{3.27}$$

We define the coupled freezing threshold $\alpha_f^{\text{c}(m=1)}$ as $\sup\{\alpha : U^{(\alpha)}(x) > 0 \text{ for all } x \in (0,1]\}$. Then we use Theorem 1 of [YJNP12], which states that for $\alpha < \alpha_f^{\text{c}(m=1)}$ and $W > O(\frac{1}{\Delta E})$ there is no

[5] This definition seems to require $\alpha \in [0,1]$; however this is not crucial since we can always rescale.

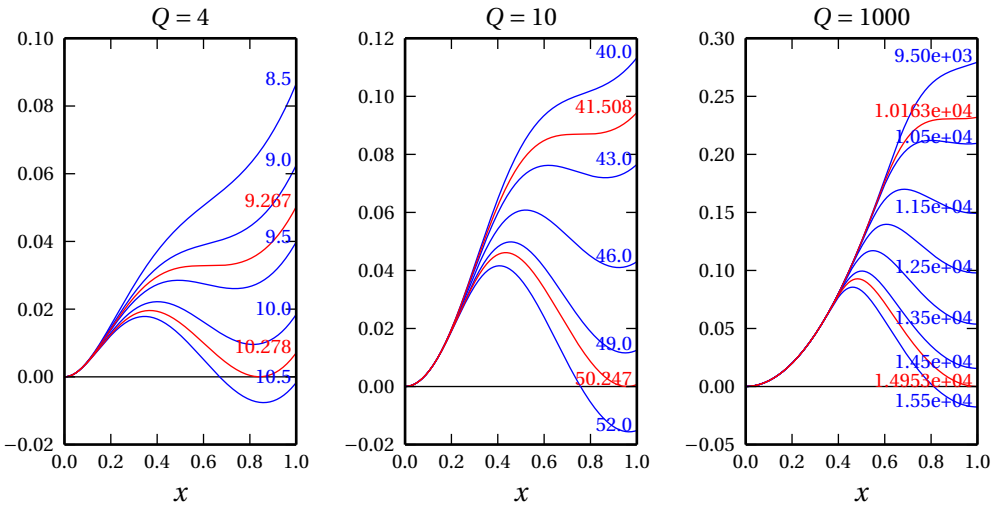


Figure 3.1 – The potential $U(x)$ for various values of Q.

fixed point of (3.26) except the trivial one. Here

$$\Delta E = \min_{x\in[u(\alpha),1]} U(x,\alpha), \qquad u(\alpha) = \sup\{x \in [0,1] : x > f(x,\alpha)\}.$$

We can then phrase our final result concerning freezing at $m = 1$.

Theorem 22. *For $\alpha < \alpha_f^{c(m=1)}$ and $W > O(\frac{1}{\Delta E})$, in the coupled Erdős-Rényi graph $G(N,\alpha,L,W;\underline{\sigma})$ with high probability a fraction of $1-o(N)$ of all vertices in the graph $G(\underline{\sigma})$ are $T(N)$-free. Here it is enough that $T(N) = \omega(1)$.*

See Figure 3.1 for the shape of the function $U(x)$ and the determination of thresholds and Figure 3.2 to get an intuition why there is not fixed point and no freezing below the coupled freezing threshold.

The asymptotic analysis of the freezing threshold for the coupled scenario can be found in Appendix D.2. We obtain

$$\alpha_f^{(m=1)c} = 2Q\log Q + 2\gamma Q - 2\log Q - 1 - 2\gamma - 2e^{-2\gamma} + o(1),$$

where $\gamma = 0.5772\ldots$ is the Euler-Mascheroni constant. For comparison, the asymptotic behaviour of the other thresholds is (see [ZK07]):

$$\alpha_f^{(m=1)} = Q\log Q + Q\log\log Q + 1 + o(1),$$
$$\alpha_s = 2Q\log Q - \log Q - 1 + o(1),$$
$$\alpha_c = 2Q\log Q - \log Q - 2\log 2 + o(1).$$

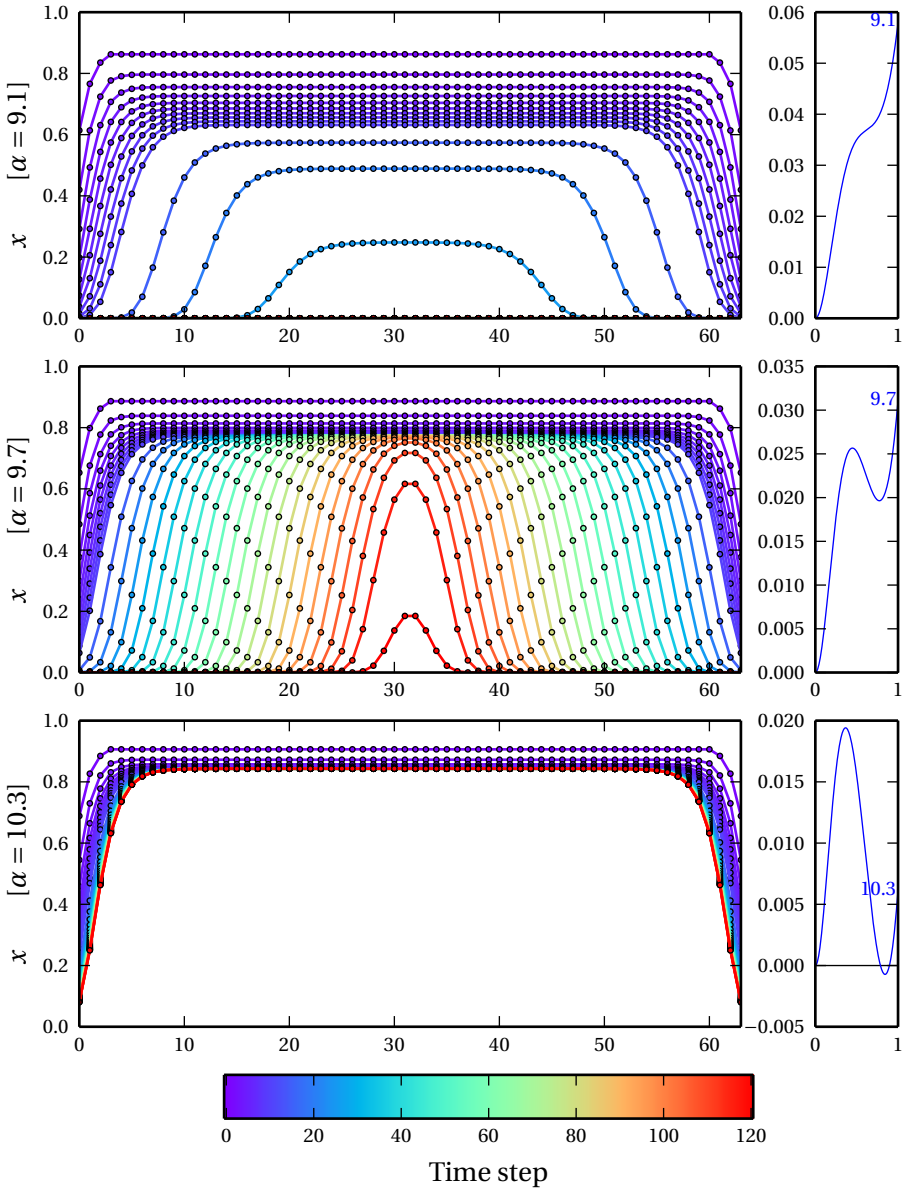


Figure 3.2 – We plot the $x_{t,z}$ for $Q = 4$, $\alpha = 9.1$, $\alpha = 9.7$ and $\alpha = 10.3$, with $L = 64$ and $W = 4$. The thresholds are located at $\alpha_f^{(m=1)} = 9.267$ and $\alpha_f^{c(m=1)} = 10.279$.
(i) The top corresponds to the regime $\alpha < \alpha_f^{(m=1)}$. Here neither the uncoupled graph or the coupled graph exhibits freezing.
(ii) The middle corresponds to $\alpha_f^{(m=1)} < \alpha < \alpha_f^{c(m=1)}$. Here only the uncoupled graph contains a proportion of frozen vertices.
(iii) The bottom corresponds to $\alpha_f^{c(m=1)} < \alpha$. In this regime both the coupled and uncoupled graphs exhibit freezing.

3.5 The special case $m = 0$

For the case $m = 0$, the 1RSB equations can be simplified considerably. We use the convention $0^0 = 0$,which we can justify by thinking of the limit $m \to 0$. Then the term $z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d)^m$ that appears in (3.15) reduces to $\mathbb{1}\{z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) > 0\}$. The meta-message passing equation can then be expressed using

$$\begin{aligned} \mathbb{F}^{(m=0)}(\mathbb{\nu}_1, \dots, \mathbb{\nu}_d)(B) &= \frac{\Pr_{\{\boldsymbol{\mu}_i \sim \mathbb{\nu}_i\}_{i\in[d]}}\left[\mathbf{f}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) \in B \wedge z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) > 0\right]}{\Pr_{\{\boldsymbol{\mu}_i \sim \mathbb{\nu}_i\}_{i\in[d]}}\left[z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) > 0\right]} \\ &= \Pr_{\{\boldsymbol{\mu}_i \sim \mathbb{\nu}_i\}_{i\in[d]}}\left[\mathbf{f}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) \in B | z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) > 0\right]. \end{aligned} \tag{3.28}$$

It is easy to check that all values of $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d$ which are not corners of the simplex $\boldsymbol{\Delta}_Q$ are treated in the same way. In other words, it only matters whether the messages $\boldsymbol{\mu}$ indicate frozen variables or not. As for the freezing setting, it only makes sense to consider $\beta = \infty$. All this means that instead of working with the meta-marginals $\mathbb{\nu}$ we only need to keep track of the mass of $\mathbb{\nu}$ resident at the Q corners of the simplex. Moreover, because of symmetry under color permutations, the masses that sit at each corner are all equal. Let x be the mass supported at any one corner.

The goal is to find a simplified version of (3.28). We project the quantities $\boldsymbol{\mu}$ to variables $\zeta \in \{1, \dots, Q, *\}$, by sending the corners of the simplex to $1, \dots, Q$ and the rest to $*$.

The (RS) message passing rule $\mathbf{f}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d)$ projects down nicely in this alphabet, because it can be written as

$$\mathbf{f}(\zeta_1, \dots, \zeta_d) = \begin{cases} q \in [Q], & \text{if for all } q' \in [Q] \setminus \{q\} \text{ there is } i \in [d] \text{ such that } \zeta_i = q', \\ *, & \text{otherwise.} \end{cases} \tag{3.29}$$

Also, the event that $z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) > 0$ can be expressed in terms of $\zeta_1, \dots, \zeta_d$:

$$\mathbb{1}\{z(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d) > 0\} = \begin{cases} 0, & \text{if for all } q \in [Q] \text{ there is } i \in [d] \text{ such that } \zeta_i = q, \\ 1 & \text{otherwise.} \end{cases} \tag{3.30}$$

We are now able to express $\mathbb{F}^{(m=0)}$ in terms of the probabilities $x_1, \dots, x_d$ that $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d$ are situated on one particular vertex of the simplex under $\mathbb{\nu}_1, \dots, \mathbb{\nu}_d$, respectively. The goal is to express the probabilities that arise in (3.28). We let the measurable set B consist of the corner corresponding to color 1. Then the denominator of (3.28) can be rewritten as

$$\Pr\left[\text{there is } q \in [Q] \text{ such that for all } i \in [d], \zeta_i \neq q\right],$$

where ζ_i is uniform on the colors with probability Qx_i and equal to $*$ with probability $1 - Qx_i$.

For $l = 0, \dots, Q-1$, the probability that elements from a fixed set of colors of size $l+1$ does not appear among $\zeta_1, \dots, \zeta_d$ is $\prod_{i=1}^{d}(1-(l+1)x_i)$. Using the inclusion-exclusion principle and the fact that there are $\binom{Q}{l+1}$ such sets, we have that the denominator of (3.28) is

$$g(x_1, \dots, x_d) = \sum_{l=0}^{Q-1} (-1)^l \binom{Q}{l+1} \prod_{i=1}^{d} (1-(l+1)x_i). \tag{3.31}$$

The numerator is computed similarly. It is the probability that one fixed color q does not appear among $\zeta_1, \dots, \zeta_d$, while all the other colors appear. The calculation is performed as before, except that now the sets of colors consist of one color that is fixed (q) and l others, which are chosen from the $[Q] \setminus \{q\}$ remaining colors. The numerator of (3.28) is then

$$f(x_1, \dots, x_d) = \sum_{l=0}^{Q-1} (-1)^l \binom{Q-1}{l} \prod_{i=1}^{d} (1-(l+1)x_i). \tag{3.32}$$

All this motivates us to define the function $\phi : [0, 1/Q]^* \to [0, 1/Q]$,

$$\phi(x_1, \dots, x_d) = \frac{\sum_{l=0}^{Q-1} (-1)^l \binom{Q-1}{l} \prod_{i=1}^{d} (1-(l+1)x_i)}{\sum_{l=0}^{Q-1} (-1)^l \binom{Q}{l+1} \prod_{i=1}^{d} (1-(l+1)x_i)}, \tag{3.33}$$

which enables us to simplify the 1-RSB equation (3.16) by tracking only the probability mass situated in the vertices of $\mathbf{\Delta}_Q$. Just as a side remark, the meta-message passing equations take the simple form

$$x^{u \to v} = \phi\left(\{x^{v' \to u}\}_{v' \in \partial u \setminus v} \right), \tag{3.34}$$

which go in the literature under the name of survey propagation (SP) equations.

The object that corresponds to $\mathscr{P}$ (a measure on the space of measures on $\mathbf{\Delta}_Q$) is now a mere probability measure supported on the interval $[0, 1/Q]$. In other words, we are seeking solutions $\mathbf{p} \in \mathbb{M}([0, 1/Q])$ to the equation

$$\mathbf{p} = \mathbb{E}_d \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1) \cdots \mathrm{d}\mathbf{p}(x_d) \delta_{\phi_d(x_1, \dots, x_d)}. \tag{3.35}$$

Note that the function ϕ takes any number of parameters. Sometimes it will be convenient to denote the parameters in a vectorized form, and then we will mention d as a subscript, so $\phi(x_1, \dots, x_d)$ could be written as $\phi_d(\underline{x})$. The same observation applies for the functions f and g above.

Regarding the total free entropy and the free entropy per cluster, we can quickly see that the quantity $m\phi^{(m)}|_{m=0}$ from (3.18) is 0 (using the convention $0 \log 0 = 0$). This is consistent with the intuition gained from the clustering picture, where all clusters are weighted by their size raised to the Parisi parameter. We can then compute the complexity Σ, which is now equal to

$m\Phi^{(m)}_{\mathrm{RSB}}|_{m=0}$. Making the dependence of Σ on $\mathbf{p}$ explicit, we obtain the functional

$$\Sigma(\mathbf{p}) = \mathbb{E}_d \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1)\cdots\mathrm{d}\mathbf{p}(x_d)\log g(x_1,\ldots,x_d) \\ -\frac{\alpha}{2}\int_{[0,1/Q]^2}\mathrm{d}\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x_2)\log(1-Qx_1x_2), \tag{3.36}$$

which is derived from (3.17) using the fact that $\Pr[z(\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_d)>0] = g(x_1,\ldots,x_d)$ and that $\Pr[z_e(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2)>0] = 1-Qx_1x_2$.

The freezing condition can then be succinctly described in the case $m=0$ as the existence of distributions $\mathbf{p}$ satisfying (3.35), which do not have all mass concentrated at $x=0$, i.e. there exist non-trivial fixed points of (3.35). The point at which non-trivial fixed points appear is in fact $\alpha_f^{(m=0)}$. In the literature it is also known as the *survey propagation* threshold α_{SP}.

3.5.1 Monotonicity properties of the functions f, g and ϕ

We present here a number of properties of ϕ_d that will be useful in our quest, but whose proofs we relegate to Appendix E.2.

Lemma 23. *The function $g_d : [0,1/Q]^d \to \mathbb{R}$ is decreasing in all parameters. Thus it attains the minimum at $g_d(\frac{1}{Q},\ldots,\frac{1}{Q})$. There is a constant K such that $g_d(\underline{x}) \geq K\left(1-\frac{1}{Q}\right)^d$ for all $d\geq 1$ and $\underline{x}\in[0,1/Q]^d$.*

Lemma 24. *The function $\phi(x_1,\ldots,x_d)$ is increasing in each of its parameters for $Q=3$.*

We conjecture that the previous lemma in fact holds for all $Q\geq 3$ but so far it was not possible to find a proof for this assertion. However, the rest of this exposition remains true under the assumption that the previous lemma holds in general. To show this, we will call this assumption the "increasing ϕ hypothesis". All the numerical evidence obtained so far support this hypothesis for $Q>3$.

Lemma 25. *Under the increasing ϕ hypothesis we have that*

$$0\leq \frac{\partial}{\partial x_1}\phi(x_1,\ldots,x_d) \leq \frac{2Q}{(Q-1)^2} \quad \textit{and} \quad -\frac{Q-1}{Q}\leq \log g(\underline{x})\leq 0,$$

for $(x_1,\ldots,x_d)\in[0,1/Q]^d$.

3.6 Proof of threshold saturation of the SP threshold to the colorability threshold

In this part we present a proof of threshold saturation. The way we define the thresholds themselves is not on actual Erdős-Rényi graphs; providing a rigorous proof that these thresholds

relate in some way to the graphs is still an open problem. We define the threshold positions in terms of equations (3.35) and the complexity functional (3.36). In this sense, our proofs are about fixed points of the type (3.35), the potentials that govern them and the relation to their spatially coupled versions. The technique of the proof is inspired by the potential method developed in [KYMP14] for LDPC codes.

The proof that we present holds for the case $Q = 3$. For larger Q, it holds under the increasing ϕ hypothesis. Experimentally, this hypothesis seems to be true for arbitrary Q as well, but its validity remains an open question.

3.6.1 Preliminaries

Let us denote by $\mathfrak{M}$ the space of probability measures on $[0, 1/Q]$. We call such measures *densities*, and let $\boldsymbol{\delta_0}$ and $\boldsymbol{\delta_{1/Q}}$ be the densities that have all mass on 0 and on $1/Q$, respectively. Then clearly for all $\mathbf{p} \in \mathfrak{M}$ it is true that $\boldsymbol{\delta_0} \preceq \mathbf{p} \preceq \boldsymbol{\delta_{1/Q}}$. We turn $\mathfrak{M}$ into a metric space by defining a distance $d_C : \mathfrak{M} \times \mathfrak{M} \to \mathbb{R}_+$. Moreover, a partial ordering (called *degradation*) is defined on $\mathfrak{M}$, satisfying $\boldsymbol{\delta_0} \preceq \mathbf{p} \preceq \boldsymbol{\delta_{1/Q}}$ for all $\mathbf{p} \in \mathfrak{M}$. The definition of the metric, the ordering and a number of topological properties of $\mathfrak{M}$ are described in Appendix E.1.

We introduce the operator $\mathscr{F} : \mathfrak{M} \to \mathfrak{M}$, $\mathscr{F}(\mathbf{p}) = \mathbb{E}_d \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1) \cdots \mathrm{d}\mathbf{p}(x_d) \delta_{\phi(x_1,\ldots,x_d)}$, so that the solutions of (3.35) are the fixed points of $\mathscr{F}$. It is easy to see that $\boldsymbol{\delta_0}$ is always a fixed point of $\mathscr{F}$, which we call *trivial fixed point*.

To be able to state the theorem, we need to introduce the coupled recurrence. In the coupled setting, we will have a distribution $\mathbf{p}_z$ (hereafter called *local density*) for each position z on the coupled chain. There are different ways in which spatial coupling can be done, and here we assume a model which turns out to simplify the exposition. As usual, the system is parameterized by two integers, L and W. In this model, both the vertices and the edges are assigned a position along a chain, in such a way that an edge at position z is connected only to vertices at positions $z - W + 1, \ldots, z$. The vertices are assigned positions between 1 and $2L - W + 1$.[6] Let us place also *dummy vertices* at positions ≤ 0 and $> 2L - W + 1$. We will allow edges to connect to such dummy variables, in which case the constraints on these edges will be thought to be always satisfied. These dummy edges are of course removable, but they are part of scheme that simplifies the exposition. We add edges in the following way: any edge at position z (which can be any integer for now) connects uniformly at random to vertices at positions $z - W + 1, \ldots, z$. Note that all edges at positions outside $\{1, \ldots, 2L\}$ will necessarily be dummy. Furthermore, some ratio of the edges at positions in $\{1, \ldots, W - 1, 2L - W + 1, \ldots, 2L\}$ will also be dummy, while all edges at the remaining positions are real (i.e. they connect two vertices). Also part of the scheme will be to work with *edge-based* local densities, so $\mathbf{p}_z$ is in fact associated to the edge position z. The objects of interest will thus be vectors of densities, denoted by $\underline{\mathbf{p}} \in \mathfrak{M}^{\{1,\ldots,2L\}}$ and called *coupled densities* or *profiles*.

[6] These numbers are chosen purely by convenience, to keep the exposition as uncluttered as possible.

We extend the notions of distance and ordering to profiles in $\mathfrak{M}^{\{1,\dots,2L\}}$ in a straightforward manner: we say that $\underline{\mathbf{p}} \preceq \underline{\mathbf{p}}'$ if for all indices $z \in \{1,\dots,2L\}$ we have $\mathbf{p}_z \preceq \mathbf{p}'_z$; we set $d_C(\underline{\mathbf{p}},\underline{\mathbf{p}}') = \sum_{z=1}^{2L} d_C(\mathbf{p}_z,\mathbf{p}'_z)$. Also, let $\underline{\boldsymbol{\delta}_{\mathbf{0}}}$ and $\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}}$ be profiles consisting of $2L$ copies of $\boldsymbol{\delta}_{\mathbf{0}}$ and $\boldsymbol{\delta}_{\mathbf{1/Q}}$, respectively. It will be helpful (and consistent with the previous discussion) to think of $\mathbf{p}_z$ as set to $\boldsymbol{\delta}_{\mathbf{0}}$ whenever the index z is outside $\{1,\dots,2L\}$.

The coupled version of the SP operator $\mathscr{F}$ is given by is $\mathscr{F}_{\mathrm{c}} : \mathfrak{M}^{\{1,\dots,2L\}} \to \mathfrak{M}^{\{1,\dots,2L\}}$,

$$\left[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})\right]_z = \frac{1}{W}\sum_{w'=0}^{W-1} \begin{cases} \mathscr{F}\left(\frac{1}{W}\sum_{w=0}^{W-1} \mathbf{p}_{z-w'+w}\right), & \text{if } z-w' \in \{1,\dots,2L-W+1\} \\ \boldsymbol{\delta}_{\mathbf{0}}, & \text{otherwise.} \end{cases} \tag{3.37}$$

To obtain an intuition of why the averaging over the window occurs twice, once before applying $\mathscr{F}$ and once afterwards, we should remember that $\mathbf{p}_z$ is a density of "messages" along *edges* situated at position z. The operation $\mathscr{F}$ is vertex-based, so in its inputs, the densities at index z need to be averages over edge-based densities coming from W positions, namely $z-W+1,\dots,z$. Also, the operation $\mathscr{F}$ is not performed at dummy vertices, which always "send" $\boldsymbol{\delta}_{\mathbf{0}}$. The output itself needs to be converted back from vertex-based densities to edge-based densities, so the edge-based density at z is formed as the average of the vertex-based densities at $z,\dots,z+W-1$.

We say that a profile $\underline{\mathbf{p}}$ is a fixed point of $\mathscr{F}_{\mathrm{c}}$ if $\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}) = \underline{\mathbf{p}}$. The fixed point is said to be *nontrivial* if $\underline{\mathbf{p}} \neq \underline{\boldsymbol{\delta}_{\mathbf{0}}}$.

We define the *coloring threshold* α_s by

$$\alpha_s = \inf\{\alpha \in [0,+\infty) : \inf_{\mathbf{p}\in\mathfrak{M}} \Sigma^{(\alpha)}(\mathbf{p}) < 0\}.$$

Note that all quantities that we work with depend on α. When we choose to make this dependence explicit, we will use a bracketed superscript.

It is now possible to state the main result regarding threshold saturation.

Theorem 26. *Under the increasing ϕ hypothesis, we have the following.*

- *For $\alpha < \alpha_s$, for a coupled system with sufficiently high length L and window size W, there is no nontrivial fixed point of the operator $\mathscr{F}_c$. Moreover, $\mathscr{F}_c^{(\infty)}(\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}}) = \underline{\boldsymbol{\delta}_{\mathbf{0}}}$.*
- *For $\alpha > \alpha_s$, for a coupled system with sufficiently high length L, there exists a nontrivial fixed point of $\mathscr{F}_c$. Moreover, $\mathscr{F}_c^{(\infty)}(\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}}) \neq \underline{\boldsymbol{\delta}_{\mathbf{0}}}$.*

The length and the window size of the coupled chain needed so that the theorem works are dependent on the complexity gap, defined below. Let $\mathfrak{T}$ be the subset of $\mathfrak{M}$ containing those $\mathbf{p}$ for which $\mathscr{F}^{(\infty)}(\mathbf{p})$ defined as $\lim_{n\to\infty} F^{(n)}(\mathbf{p})$ exists and is equal to $\boldsymbol{\delta}_{\mathbf{0}}$.

We define the *complexity gap* $\Delta\Sigma^{(\alpha)}$ as

$$\Delta\Sigma^{(\alpha)} = \inf_{\mathbf{p}\in\mathfrak{M}\setminus\mathfrak{T}^{(\alpha)}} \Sigma^{(\alpha)}(\mathbf{p}).$$

Note that this is only defined for the values of α for which $\mathfrak{T}^{(\alpha)} \neq \mathfrak{M}$. We will see soon that $\mathfrak{T}^{(\alpha)} = \mathfrak{M}$ is equivalent to $\mathscr{F}$ not having nontrivial fixed points.

Lemma 27. *Under the increasing ϕ hypothesis, for $\alpha < \alpha_s$ we have that $\Delta\Sigma^{(\alpha)} > 0$.*

Proof. Since $\mathfrak{M}\setminus\mathfrak{T}^{(\alpha)}$ is compact (Lemma 64 in Appendix E.4), there is $\mathbf{p} \in \mathfrak{M}\setminus\mathfrak{T}$ so that $\Sigma^{(\alpha)}(\mathbf{p}) = \Delta\Sigma^{\alpha}$. Suppose, for the purpose of contradiction, $\Delta\Sigma^{(\alpha)} \leq 0$, using Lemma 36 which gives us $\frac{\mathrm{d}}{\mathrm{d}\alpha}\Sigma^{(\alpha)}(\mathbf{p}) < 0$, we conclude that for some $\alpha' \in (\alpha, \alpha_s)$ we have $\Sigma^{(\alpha')}(\mathbf{p}) < \Sigma^{(\alpha)}(\mathbf{p}) \leq 0$. But this contradicts the definition of α_s. □

Proof of Theorem (26). The two claims of the theorem follow independently from Lemmas 45 and 49. These can be applied once we verify that $\Delta\Sigma > 0$ in the case $\alpha < \alpha_s$ and $\Delta\Sigma < 0$ for $\alpha > \alpha_s$, which follow from Lemmas 27 and 37, respectively. □

3.6.2 Properties of the operator $\mathscr{F}$

We state here some properties regarding monotonicity and continuity of $\mathscr{F}$ and then the existence of fixed points under some conditions. Proofs are provided in Appendix E.3.

Lemma 28. *Under the increasing ϕ hypothesis, if $\mathbf{p} \preceq \mathbf{p}'$, then $\mathscr{F}(\mathbf{p}) \preceq \mathscr{F}(\mathbf{p}')$.*

Lemma 29. *If $\alpha \leq \alpha'$ then $\mathscr{F}_\alpha(\mathbf{p}) \succeq \mathscr{F}_{\alpha'}(\mathbf{p})$.*

Lemma 30. *Under the increasing ϕ hypothesis, the operator $\mathscr{F}$ is continuous.*

Suppose a density $\mathbf{p}$ has the property that $\mathbf{p} \preceq \mathscr{F}(\mathbf{p})$. Then due to the monotonicity of $\mathscr{F}$, the sequence of densities $\mathbf{p}, \mathscr{F}(\mathbf{p}), \mathscr{F}^{(2)}(\mathbf{p}), \ldots$ is itself monotone and thus convergent. It then makes sense to speak of $F^{(\infty)}(\mathbf{p})$ as its limit. The same holds when $\mathbf{p} \succeq \mathscr{F}(\mathbf{p})$.

Lemma 31. *Let $\mathbf{p}$ be such that $\mathbf{p}$ and $\mathscr{F}(\mathbf{p})$ are degraded one with respect to the other. Then under the increasing ϕ hypothesis, $\mathscr{F}^{(\infty)}(\mathbf{p})$ is a fixed point of $\mathscr{F}$.*

It can also be seen immediately that $\mathscr{F}^{(\infty)}(\boldsymbol{\delta}_{\mathbf{1/Q}}) = \boldsymbol{\delta}_{\mathbf{0}}$ is equivalent to the existence of a nontrivial fixed point. If equality does not hold, a nontrivial fixed point is supplied by $\mathscr{F}^{(\infty)}(\boldsymbol{\delta}_{\mathbf{1/Q}})$. If it does hold, then by monotonicity of $\mathscr{F}$ we find that $\boldsymbol{\delta}_{\mathbf{1/Q}} \succeq \mathbf{p}$ implies $\boldsymbol{\delta}_{\mathbf{0}} \succeq \mathscr{F}^{(\infty)}(\mathbf{p})$; this leaves no space for a nontrivial fixed point.

3.6.3 Properties of the complexity functional

We present here the reason why the complexity functional serves as potential for the recursions given by the operator $\mathscr{F}$. This will be apparent once we compute the directional derivative of $\Sigma(\mathbf{p})$.

We start with some technical prerequisites.

Lemma 32. *The complexity functional Σ is continuous.*

The proof is presented in Appendix E.5.1.

The next step will be to show that fixed points of $\mathscr{F}$ correspond to stationary points of the complexity functional. The space $\mathfrak{M}$ is infinitely dimensional and even though one could in principle characterize stationarity using the Fréchet derivative, it is already enough for us to use simpler to express directional derivatives. These are defined as follows. Given two densities $\mathbf{p}, \mathbf{p}' \in \mathfrak{M}$, the directional derivative of Σ at $\mathbf{p}$ in the direction $\delta\mathbf{p} = \mathbf{p}' - \mathbf{p}$ is given by

$$\delta\Sigma(\mathbf{p})[\delta\mathbf{p}] = \lim_{t \searrow 0} \frac{\Sigma(\mathbf{p} + t\delta\mathbf{p}) - \Sigma(\mathbf{p})}{t}. \tag{3.38}$$

Note that this is not defined for any signed measure $\delta\mathbf{p}$, but for only those which have the property that $\mathbf{p} + \delta\mathbf{p} \in \mathfrak{M}$. These signed measures will be referred to as *directions*.

A stationary point is then a density $\mathbf{p}$ for which $\delta\Sigma(\mathbf{p})[\delta\mathbf{p}] = 0$ for all directions $\delta\mathbf{p}$.

One potential problem that we might have in computing the derivative could be that the underlying sum over d in $\mathbb{E}_d$ is infinite and the limit in the sum might commute with the derivation. The following lemma ensures that Σ on an interpolation path between $\mathbf{p}$ and $\mathbf{p}'$ is an absolutely convergent power series and so we can differentiate the sum it contains term by term.

Lemma 33. *Fix $\mathbf{p}, \mathbf{p}' \in \mathfrak{M}$. Then $\Sigma(\mathbf{p} + t(\mathbf{p}' - \mathbf{p}))$ as a real function of $t \in \mathbb{R}$ is analytic.*

The proof is presented in Appendix E.5.2. This lemma allows us to read the value of the directional derivative as the coefficient of t in the expansion as a power of t of $\Sigma(\mathbf{p} + t\delta\mathbf{p})$. Explicitly, we obtain

$$\begin{aligned}\delta\Sigma(\mathbf{p})[\delta\mathbf{p}] = \mathbb{E}_d d \int_{[0,1/Q]^d} \mathrm{d}\delta\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_d)\log g(x_1,\ldots,x_d) - \\ - \alpha \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x_2)\log(1 - Qx_1x_2).\end{aligned}$$

Since the first variable x_1 is "special", we separate it inside the logarithm, obtaining

$$\begin{aligned}
&\log g(x_1,\dots,x_d) = \\
&= \log\left(\sum_{l=0}^{Q-1}(-1)^l\binom{Q}{l+1}\prod_{i=1}^{d}(1-(l+1)x_i)\right) = \\
&= \log\left(\sum_{l=0}^{Q-1}(-1)^l\binom{Q}{l+1}\prod_{i=2}^{d}(1-(l+1)x_i) - x_1\sum_{l=0}^{Q-1}(-1)^l(l+1)\binom{Q}{l+1}\prod_{i=2}^{d}(1-(l+1)x_i)\right) \\
&= \log\left(\sum_{l=0}^{Q-1}(-1)^l\binom{Q}{l+1}\prod_{i=2}^{d}(1-(l+1)x_i) - Qx_1\sum_{l=0}^{Q-1}(-1)^l\binom{Q-1}{l}\prod_{i=2}^{d}(1-(l+1)x_i)\right) \\
&= \log\big(g(x_2,\dots,x_d) - Qx_1 f(x_2,\dots,x_d)\big) \\
&= \log\big(1-Qx_1\phi(x_2,\dots,x_d)\big) + \log g(x_2,\dots,x_d),
\end{aligned} \tag{3.39}$$

where $f(\cdot)$ and $g(\cdot)$ are the numerator and the denominator of $\phi(\cdot)$. Note now that the second term does not depend on x_1, and the variable x_1 is integrated with respect to the measure difference $\delta\mathbf{p}$. This causes the integral of the second term to be zero.

Also observe that $\mathbb{E}_d dF(d)$ can be rewritten as $\alpha\mathbb{E}_d F(d+1)$. This happens because the two are equal to the two quantities at the extremities in

$$\sum_{d\geq 0} d\frac{\alpha^d e^{-\alpha}}{d!}F(d) = \alpha\sum_{d\geq 1}\frac{\alpha^{d-1}e^{-\alpha}}{(d-1)!}F(d) = \alpha\sum_{d\geq 0}\frac{\alpha^d e^{-\alpha}}{d!}F(d+1). \tag{3.40}$$

Taking into account the two previous observations, we deduce that

$$\begin{aligned}
&\delta\Sigma(\mathbf{p})[\delta\mathbf{p}] = \\
&= \alpha\mathbb{E}_d\int_{[0,1/Q]^{d+1}}\mathrm{d}\delta\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_{d+1})\log\big(1-Qx_1\phi(x_2,\dots,x_{d+1})\big) - \\
&\qquad - \alpha\int_{[0,1/Q]^2}\mathrm{d}\delta\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x)\log(1-Qx_1x) \\
&= \alpha\int_{[0,1/Q]}\mathrm{d}\delta\mathbf{p}(x_1)\Big\{\mathbb{E}_d\int_{[0,1/Q]^d}\mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_{d+1})\log\big(1-Qx_1\phi(x_2,\dots,x_{d+1})\big) - \\
&\qquad - \int_{[0,1/Q]}\mathrm{d}\mathbf{p}(x)\log(1-Qx_1x)\Big\}.
\end{aligned} \tag{3.41}$$

Then the directional derivative becomes

$$\begin{aligned}
&\delta\Sigma(\mathbf{p})[\delta\mathbf{p}] = \\
&= \alpha\int_{[0,1/Q]}\mathrm{d}\delta\mathbf{p}(x_1)\Big\{\int_{[0,1/Q]}\mathrm{d}(\mathscr{F}(\mathbf{p}))(x)\log(1-Qx_1x) - \int_{[0,1/Q]}\mathrm{d}\mathbf{p}(x)\log(1-Qx_1x)\Big\} \\
&= \alpha\int_{[0,1/Q]}\mathrm{d}\delta\mathbf{p}(x_1)\int_{[0,1/Q]}\mathrm{d}(\mathscr{F}(\mathbf{p})-\mathbf{p})(x)\log(1-Qx_1x)\,.
\end{aligned} \tag{3.42}$$

Looking at this form of the directional derivative, we immediately see that a fixed point of $\mathscr{F}$ makes the directional derivative 0, a fact summarized in the following lemma.

Lemma 34. *Any density* $\mathbf{p} \in \mathfrak{M}$ *which satisfies* $\mathscr{F}(\mathbf{p}) = \mathbf{p}$ *is a stationary point of* Σ.

Some of the intermediary results are also worthy to be accounted separately, as they will be of use later.

Lemma 35. *Given* $\mathbf{p}_1, \mathbf{p}_1', \mathbf{p}_2, \ldots, \mathbf{p}_d \in \mathfrak{M}$ *and setting* $\delta\mathbf{p}_1 = \mathbf{p}_1' - \mathbf{p}_1$, *we have that*

$$\int_{[0,1/Q]^d} \mathrm{d}\delta\mathbf{p}_1(x_1)\,\mathrm{d}\mathbf{p}_2(x_2)\cdots\mathrm{d}\mathbf{p}_d(x_d)\log g(x_1,\ldots,x_d) = \\ = \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_1(x_1)\,\mathrm{d}(\mathscr{F}^{(d-1)}(\mathbf{p}_2,\ldots,\mathbf{p}_d))(x)\log(1-Qx_1x)\,.$$

If d *is Poisson*(α)*-distributed, then given* $\mathbf{p}_1, \mathbf{p}_1', \mathbf{p} \in \mathfrak{M}$ *and* $\delta\mathbf{p}_1$ *as above, we have*

$$\mathbb{E}_d \int_{[0,1/Q]^{d+1}} \mathrm{d}\delta\mathbf{p}_1(x_1)\,\mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_{d+1})\log g(x_1,\ldots,x_d) = \\ \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}(x_1)\,\mathrm{d}(\mathscr{F}(\mathbf{p}))(x)\log(1-Qx_1x)\,.$$

We show two more technical results, needed to ensure the existence of the threshold and of a nonzero complexity gap in the main theorem. The proofs are presented in the Appendix in Sections E.5.3 and E.5.4.

Lemma 36. *Let* $\mathbf{p}$ *be a fixed point of* $\mathscr{F}$. *Then*

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\Sigma^{(\alpha)}(\mathbf{p}) = \mathbb{E}_d \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1)\cdots\mathrm{d}\mathbf{p}(x_d)\log g(x_1,\ldots,x_d) \\ + \frac{1}{2}\int_{[0,1/Q]^2} \mathrm{d}\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x_2)\log(1-Qx_1x_2).$$

In particular, the derivative is strictly negative when $\mathbf{p}$ *is a nontrivial fixed point.*

Lemma 37. *Suppose that* $\inf_{\mathbf{p}\in\mathfrak{M}}\Sigma(\mathbf{p}) \le 0$ *and suppose there is* $\mathbf{p}_* \in \mathfrak{M}\setminus\{\boldsymbol{\delta}_\mathbf{0}\}$ *such that the infimum is achieved at* $\mathbf{p}_*$. [7] *Then* $\mathbf{p}_*$ *is a fixed point of* $\mathscr{F}$ *and, moreover,* $\Delta\Sigma = \inf_{\mathbf{p}\in\mathfrak{M}}\Sigma(\mathbf{p})$.

3.6.4 The coupled potential

All lemmas relating the ordering and the topology of $\mathfrak{M}$ naturally extend to $\mathfrak{M}^{2L}$. In particular, a monotonous (with respect to degradation) sequence of profiles always converges.

The properties given by Lemmas 28, 29 and 30 are straightforward generalizations to the

[7] The existence of a $\mathbf{p}_*$ is guaranteed in the case where the infimum is strictly negative, because $\mathfrak{M}$ is compact and $\Sigma(\boldsymbol{\delta}_\mathbf{0}) = 0$.

spatially coupled scenario, as summarized by the following statements. For this reason, the proofs are omitted.

Lemma 38. *Under the increasing ϕ hypothesis, if $\underline{\mathbf{p}} \preceq \underline{\mathbf{p}}'$ then $\mathscr{F}_c(\underline{\mathbf{p}}) \preceq \mathscr{F}_c(\underline{\mathbf{p}}')$.*

Lemma 39. *If $\alpha \leq \alpha'$ then $\mathscr{F}_{c(\alpha)}(\mathbf{p}) \succeq \mathscr{F}_{c(\alpha')}(\mathbf{p})$.*

Lemma 40. *Under the increasing ϕ hypothesis, for $\underline{\mathbf{p}} \succeq \underline{\mathbf{p}}'$, we have that $d_C(\mathscr{F}_c(\underline{\mathbf{p}}), \mathscr{F}(\underline{\mathbf{p}}') = O(d_C(\mathbf{p}, \mathbf{p}'))$ as $d_C(\mathbf{p}, \mathbf{p}') \to 0$.*

Note that in the previous lemma, the constant inside the O-notation absorbs quantities that depend on L and W.

Lemma 41. *Let $\underline{\mathbf{p}}$ be such that $\underline{\mathbf{p}}$ and $\mathscr{F}_c(\underline{\mathbf{p}})$ are degraded one with respect to the other. Then under the increasing ϕ hypothesis, $\mathscr{F}_c^{(\infty)}(\underline{\mathbf{p}})$ is a fixed point of $\mathscr{F}_c$.*

It will be desirable to work with coupled densities exhibiting an ordering between the local densities at each position, for example increasing until position L and decreasing afterwards. By just examining the transformation $\mathscr{F}_\mathrm{c}$ it is not clear that it maintains such a property, especially around the middle of the chain, at position L. For this reason we will apply the following trick. We will work with *one-sided profiles*. These are profiles for which the local densities at positions $L, L+1, \ldots, 2L$ are all equal. To enforce this constraint, we add a coupled operator which simply replicates the local density at L to all positions to the right, that is

$$[\mathscr{G}_\mathrm{c}(\underline{\mathbf{p}})]_z = \begin{cases} \mathbf{p}_z, & \text{if } z \leq L \\ \mathbf{p}_L, & \text{if } z > L. \end{cases} \tag{3.43}$$

We call a profile $\underline{\mathbf{p}}$ *position-monotone* if $\mathbf{p}_z \preceq \mathbf{p}_{z+1}$ for all $z = 1, \ldots, 2L-1$. Clearly $\underline{\boldsymbol{\delta}_\mathbf{0}}$ and $\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}}$ are both position-monotone. The purpose of introducing the operator $\mathscr{G}_\mathrm{c}$ is that even though $\mathscr{F}_\mathrm{c}$ might not preserve position-monotonicity, $\mathscr{G}_\mathrm{c} \circ \mathscr{F}_\mathrm{c}$ does. This fact, formalized by the following lemma, motivates us to consider the repeated application of $\mathscr{G}_\mathrm{c} \circ \mathscr{F}_\mathrm{c}$, which we will name the *one-sided recursion*.

Lemma 42. *If $\underline{\mathbf{p}}$ is position-monotone, then $\mathscr{G}_c \circ \mathscr{F}_c(\underline{\mathbf{p}})$ is as well, under the increasing ϕ hypothesis.*

Proof. It is enough to show that $\left[\mathscr{F}_\mathrm{c}(\underline{\mathbf{p}})\right]_z \preceq \left[\mathscr{F}_\mathrm{c}(\underline{\mathbf{p}})\right]_{z+1}$ for $1 \leq z < L$. This is obtained from the definition of $\mathscr{F}_\mathrm{c}$ by putting together the following facts: (i) if $\underline{\mathbf{p}}$ is position-monotone, then so is the profile given by $\sum_{w=0}^{W-1} \mathbf{p}_{z-w}$ (where as usual $\mathbf{p}_z$ is assumed to be equal to $\boldsymbol{\delta}_\mathbf{0}$ for $z \leq 0$); (ii) $\mathscr{F}$ is monotone (Lemma 28) and (iii) if $\underline{\mathbf{p}}'_z$ is position-monotone then at least for $1 \leq z \leq L$ the profile given by $\sum_{w=0}^{W-1} \mathbf{p}_{z+w}$ satisfies the position-monotonicity condition (in fact the only problem is at the right end of the profile). □

We now state the counterpart of Lemma 38 for the one-sided recursion.

Lemma 43. *Under the increasing ϕ hypothesis, if $\underline{\mathbf{p}} \preceq \underline{\mathbf{p}}'$ then $\mathscr{G}_c \circ \mathscr{F}_c(\underline{\mathbf{p}}) \preceq \mathscr{G}_c \circ \mathscr{F}_c(\underline{\mathbf{p}})$. As a consequence, the sequence $\{(\mathscr{G}_c \circ \mathscr{F}_c)^{(n)}(\underline{\boldsymbol{\delta}_{1/Q}})\}$ is monotonous and thus the limit $\{(\mathscr{G}_c \circ \mathscr{F}_c)^{(\infty)}(\underline{\boldsymbol{\delta}_{1/Q}})\}$ exists and is a fixed point of the operator $\mathscr{G}_c \circ \mathscr{F}_c$.*

The next step is to show that we need only look at the one-sided recursion initialized with $\underline{\boldsymbol{\delta}_{1/Q}}$.

Lemma 44. *If $(\mathscr{G}_c \circ \mathscr{F}_c)^{(\infty)}(\underline{\boldsymbol{\delta}_{1/Q}}) = \underline{\boldsymbol{\delta}_0}$, then $\mathscr{F}_c^{(\infty)}(\underline{\mathbf{p}}) = \underline{\boldsymbol{\delta}_0}$, for any $\underline{\mathbf{p}}$, and thus there are no non-trivial fixed points of $\mathscr{F}_c$.*

Proof. It is sufficient to prove by induction on n that $\mathscr{F}_{\mathrm{c}}^{(n)}(\underline{\mathbf{p}}) \preceq (\mathscr{G}_{\mathrm{c}} \circ \mathscr{F}_{\mathrm{c}})^{(n)}(\underline{\boldsymbol{\delta}_{1/Q}})$. Clearly in the case $n = 0$ this holds. For the induction step, we can verify the degradation at positions $1, \dots, L$ using Lemma 38, since $\mathscr{G}$ has no effect. For $z > L$, we first note that if $\mathscr{H}_{\mathrm{c}}$ is the operator that reflects the whole chain, i.e. $[\mathscr{H}_{\mathrm{c}}(\underline{\mathbf{p}})]_z = \mathbf{p}_{2L+1-z}$, then $\mathscr{F}_{\mathrm{c}}$ commutes with this reflection, so $\mathscr{H}_{\mathrm{c}} \circ \mathscr{F}_{\mathrm{c}}^{(n)}(\underline{\mathbf{p}}) = \mathscr{F}_{\mathrm{c}}^{(n)} \circ \mathscr{H}_{\mathrm{c}}(\underline{\mathbf{p}})$. This allows us to write

$$[\mathscr{F}_{\mathrm{c}}^{(n+1)}(\underline{\mathbf{p}})]_z = [\mathscr{F}_{\mathrm{c}}^{(n+1)} \circ \mathscr{H}_{\mathrm{c}}(\underline{\mathbf{p}})]_{2L+1-z} \preceq [(\mathscr{G}_{\mathrm{c}} \circ \mathscr{F}_{\mathrm{c}})^{(n)}(\underline{\boldsymbol{\delta}_{1/Q}})]_{2L+1-z} \preceq [(\mathscr{G}_{\mathrm{c}} \circ \mathscr{F}_{\mathrm{c}})^{(n)}(\underline{\boldsymbol{\delta}_{1/Q}})]_z,$$

where the second to last inequality follows from what we have already established in the $z \le L$ case, and the last inequality is given by Lemma 42. □

3.6.5 The main argument: $\alpha < \alpha_s$

Lemma 45. *Under the increasing ϕ hypothesis, if $\Delta\Sigma > 0$, if $W > \frac{2(\alpha^2+1)}{Q\Delta\Sigma}$ and $L > 4W$, the only fixed point of the coupled SP equation is $\underline{\boldsymbol{\delta}_0}$.*

The proof will require an estimate of the coupled version of the complexity functional and its first and second derivative in a certain direction. We first introduce the *coupled complexity functional* as $\Sigma_c : \mathfrak{M}^{\{1,\dots,2L\}} \to \mathbb{R}$,

$$\begin{aligned}\Sigma_{\mathrm{c}}(\underline{\mathbf{p}}) = &\sum_{z'=1}^{2L-W+1} \mathbb{E}_d \int_{[0,1/Q]^d} \prod_{i=1}^{d} \left(\frac{1}{W} \sum_{w=0}^{W-1} \mathrm{d}\mathbf{p}_{z'+w}(x_i) \right) \log g(x_1, \dots, x_d) \\ &+ \frac{\alpha}{2} \sum_{z=1}^{2L} \int_{[0,1/Q]^2} \mathrm{d}\mathbf{p}_z(x_1)\, \mathrm{d}\mathbf{p}_z(x_2) \log(1 - Q x_1 x_2), \end{aligned} \tag{3.44}$$

where we adopt the convention that $\mathbf{p}_z = \boldsymbol{\delta}_{\mathbf{0}}$ for all $z \notin \{1, \dots, 2L\}$. The derivatives will be computed with respect to the right-shift direction, obtained as follows. Let $\mathscr{H} : \mathfrak{M}^{\{1,\dots,2L\}} \to \mathfrak{M}^{\{1,\dots,2L\}}$ be defined by $[\mathscr{H}(\underline{\mathbf{p}})]_1 = \boldsymbol{\delta}_{\mathbf{0}}$ and $[\mathscr{H}(\underline{\mathbf{p}})]_z = \underline{\mathbf{p}}_{z-1}$ for $z = 2, \dots, 2L$. The directions that we consider in the remaining proofs are all of the form $\mathscr{H}(\underline{\mathbf{p}}) - \underline{\mathbf{p}}$.

Proof. Because of Lemma 44, it is enough to show that there is no nontrivial fixed point of $\mathscr{G}_{\mathrm{c}} \circ \mathscr{F}_{\mathrm{c}}$. Assume that $\underline{\mathbf{p}}$ is such a fixed point; from this we will derive a contradiction. We

consider the coupled complexity functional restricted to the convex combinations of $\underline{\mathbf{p}}$ and $\mathscr{H}(\underline{\mathbf{p}})$. These are parameterized by $t \in [0,1]$ as $(1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}})$.

As in the case of the non-coupled scenario, we can show that the function $\Sigma_c((1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}}))$ as a function of $t \in [0,1]$ is analytic. The next step is to write Taylor's theorem for this function with the remainder in Lagrange form: for a function $f : [0,1] \to \mathbb{R}$ twice continuously differentiable, we have that $f(1) = f(0) + f'(0) + \frac{1}{2}f''(x)$, for some $x \in (0,1)$. In our case,

$$\Sigma_c(\mathscr{H}(\underline{\mathbf{p}})) - \Sigma_c(\underline{\mathbf{p}}) = \frac{\mathrm{d}}{\mathrm{d}t}\Sigma_c((1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}}))\Big|_{t=0} + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\Sigma_c((1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}}))\Big|_{t=s}, \quad (3.45)$$

for some $s \in (0,1)$.

By Lemmas 46, 47 and 48 below we have the following bounds:

$$\begin{aligned}
&\left|\Sigma_c(\mathscr{H}(\underline{\mathbf{p}})) - \Sigma_c(\underline{\mathbf{p}})\right| = \Sigma(\mathbf{p}_{2L}),\\
&\frac{\mathrm{d}}{\mathrm{d}t}\Sigma_c((1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}}))\Big|_{t=0} = 0,\\
&\frac{1}{2}\left|\frac{\mathrm{d}^2}{\mathrm{d}t^2}\Sigma_c((1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}}))\Big|_{t=s}\right| \le \frac{2(\alpha^2+1)}{QW},
\end{aligned}$$

which when plugged back into (3.45) imply $\Sigma(\mathbf{p}_{2L}) \le \frac{2(\alpha^2+1)}{QW}$. Observe now that $\Sigma(\mathbf{p}_{2L}) \ge \Delta\Sigma$ (this holds whenever $\mathscr{F}(\mathbf{p}_{2L}) \neq \boldsymbol{\delta}_{\mathbf{0}}$, see definition of $\Delta\Sigma$; if this were not true, $\underline{\mathbf{p}}$ would not be a fixed point of $\mathscr{G}\circ\mathscr{F}$). If $W > \frac{2(\alpha^2+1)}{Q\Delta\Sigma}$ then the claim follows by contradiction. □

In the next three lemmas we assume that the coupled chain is such that $L > 4W$.

Lemma 46. *If $\underline{\mathbf{p}} \in \mathfrak{M}^{\{1,\dots,2L\}}$ is one-sided (i.e. $\mathbf{p}_{L+1} = \dots = \mathbf{p}_{2L}$), we have that*

$$\Sigma_c(\underline{\mathbf{p}}) = \Sigma_c(\mathscr{H}(\underline{\mathbf{p}})) + \Sigma(\mathbf{p}_{2L}).$$

Proof. Let us examine first the definition of Σ_c. Note that because of the one-sidedness of the profile, the terms corresponding to $z' \in \{L+W-1,\dots,2L-W+1\}$ and $z \in \{L,\dots,2L\}$ appearing in the two sums(3.44) are all equal, respectively. If we remove one such term from each of the sums, say the one corresponding to $z' = z = 2L-W+1$, the two removed terms combined give the (uncoupled) complexity functional of the local density that is repeated on the right half of the profile, namely $\Sigma(\mathbf{p}_{z'}) = \Sigma(\mathbf{p}_{2L})$. We now look at what remains in the sum. We perform a shift in the indices that does not modify the quantity. We shift all local densities in the profile at positions $< z'$ one position to the right (padding it with $\boldsymbol{\delta}_{\mathbf{0}}$ on the leftmost position), thereby obtaining the profile $\mathscr{H}(\underline{\mathbf{p}})$, while also shifting the corresponding summation indices z' and z one to the right. Also, since now the two sums start at $z' = z = 2$, we need to extend the sums to terms corresponding to $z' = z = 1$, but that is trivially done, as these terms are be 0. We can now identify what remains as $\Sigma_c(\mathscr{H}(\underline{\mathbf{p}}))$. □

Lemma 47. *Suppose that* $\underline{\mathbf{p}} \in \mathfrak{M}^{\{1,\dots,2L\}}$ *is a one-sided profile such that* $\underline{\mathbf{p}} = \mathscr{G} \circ \mathscr{F}(\underline{\mathbf{p}})$ *(i.e. it is a fixed point of* $\mathscr{G} \circ \mathscr{F}$*). Then* $\frac{\mathrm{d}}{\mathrm{d}t} \Sigma_c((1-t)\underline{\mathbf{p}} + t\mathscr{H}(\underline{\mathbf{p}}))\Big|_{t=0} = 0$.

Proof. We adopt the same tactic as for the uncoupled case (Lemma 34). What we want is essentially the directional derivative of Σ_c at $\underline{\mathbf{p}}$ in the direction $\delta\underline{\mathbf{p}} = \mathscr{H}(\underline{\mathbf{p}}) - \underline{\mathbf{p}}$, which we compute in a manner similar to the derivation of (3.42):

$$\begin{aligned}
&\delta\Sigma_{\mathrm{c}}(\underline{\mathbf{p}})[\delta\underline{\mathbf{p}}] \\
&= \sum_{z'=1}^{2L-W+1} \mathbb{E}_d d \int_{[0,1/Q]^d} \left(\frac{1}{W} \sum_{w=0}^{W-1} \mathrm{d}\delta\mathbf{p}_{z'+w}(x_i) \right) \prod_{i=2}^{d} \left(\frac{1}{W} \sum_{w=0}^{W-1} \mathrm{d}\mathbf{p}_{z'+w}(x_i) \right) \log g(x_1,\dots,x_d) \\
&\quad + \alpha \sum_{z=1}^{2L} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_z(x_1)\, \mathrm{d}\mathbf{p}_z(x_2) \log(1 - Q x_1 x_2). \qquad (3.46)
\end{aligned}$$

We follow the steps in the derivation of (3.42), so we use the fact that d is Poisson and the second part of Lemma 35:

$$\begin{aligned}
&\mathbb{E}_d d \int_{[0,1/Q]^{d-1}} \prod_{i=2}^{d} \left(\frac{1}{W} \sum_{w=0}^{W-1} \mathrm{d}\mathbf{p}_{z'+w}(x_i) \right) \log g(x_1,\dots,x_d) = \\
&\qquad = \alpha \int_{[0,1/Q]} \mathrm{d}\mathscr{F}\left(\frac{1}{W} \sum_{w=0}^{W-1} \mathrm{d}\mathbf{p}_{z'+w}(x_i) \right)(x) \log(1 - Q x_1 x).
\end{aligned}$$

We then obtain

$$\begin{aligned}
&\delta\Sigma_{\mathrm{c}}(\underline{\mathbf{p}})[\delta\underline{\mathbf{p}}] = \\
&= \sum_{z'=1}^{2L-W+1} \frac{1}{W} \sum_{w'=0}^{W-1} \alpha \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_{z'+w'}(x_1)\, \mathrm{d}\mathscr{F}\left(\frac{1}{W} \sum_{w=0}^{W-1} \mathrm{d}\mathbf{p}_{z'+w}(x_i) \right)(x) \log(1 - Q x_1 x) \\
&\quad + \alpha \sum_{z=1}^{2L} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_z(x_1)\, \mathrm{d}\mathbf{p}_z(x_2) \log(1 - Q x_1 x_2).
\end{aligned}$$

In the first sum we make the change of variables $z' + w' \to z$. We will sum z over $\{1,\dots,2L\}$, and w' over $\{0,\dots,W-1\}$. Since we sum over more terms now, we will take care that the extra terms, i.e. all those characterized by $z - w' \le 0$ or $z - w' > 2L - W + 1$, are all 0. This is readily apparent below, as $\int_{[0,1/Q]} \mathrm{d}\boldsymbol{\delta_0}(x) \log(1 - Q x_1 x) = 0$:

$$\begin{aligned}
\delta\Sigma_{\mathrm{c}}(\underline{\mathbf{p}})[\delta\underline{\mathbf{p}}] &= \alpha \sum_{z=1}^{2L} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_z(x_1) \frac{1}{W} \sum_{w'=0}^{W-1} \mathrm{d}\mathbf{u}_{z-w'}(x) \log(1 - Q x_1 x) \\
&\quad + \alpha \sum_{z=1}^{2L} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_z(x_1)\, \mathrm{d}\mathbf{p}_z(x_2) \log(1 - Q x_1 x_2),
\end{aligned}$$

where

$$\mathbf{u}_{z-w'} = \frac{1}{W}\sum_{w'=0}^{W-1}\begin{cases}\mathscr{F}\left(\frac{1}{W}\sum_{w=0}^{W-1}\mathbf{p}_{z-w'+w}\right), & \text{if } z-w' \in \{1,\dots,2L-W+1\}\\ \boldsymbol{\delta}_{\mathbf{0}}, & \text{otherwise.}\end{cases} = \left[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})\right]_z.$$

Thus we can summarize our calculation as

$$\delta\Sigma_{\mathrm{c}}(\underline{\mathbf{p}})[\delta\underline{\mathbf{p}}] = \alpha\sum_{z=1}^{2L}\int_{[0,1/Q]}\mathrm{d}\delta\mathbf{p}_z(x_1)\int_{[0,1/Q]}\mathrm{d}\left(\left[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})\right]_z - \mathbf{p}_z\right)(x)\log(1-Qx_1x)\,. \tag{3.47}$$

What we obtained in (3.47) holds in fact in a more general setting, in which $\underline{\mathbf{p}}$ is any profile (not necessarily one-sided) and $\delta\underline{\mathbf{p}}$ is any difference of profiles. We will find this observation useful later.

We now note that fixed points $\underline{\mathbf{p}}$ of $\mathscr{G}\circ\mathscr{F}$ satisfy (i) for $1\le z\le L$, $\left[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})\right]_z = \mathbf{p}_z$ and (ii) for $L+1\le z\le 2L$, $[\mathscr{H}(\underline{\mathbf{p}})]_z = \mathbf{p}_{z-1} = \mathbf{p}_z$ and so $\delta\mathbf{p}_z = 0$. Thus, each of the $2L$ terms in (3.47) is zero and the claim follows. □

Lemma 48. *Suppose that $\underline{\mathbf{p}}\in\mathfrak{M}^{\{1,\dots,2L\}}$ is a one-sided profile such that $\underline{\mathbf{p}} = \mathscr{G}\circ\mathscr{F}(\underline{\mathbf{p}})$ (i.e. it is a fixed point of $\mathscr{G}\circ\mathscr{F}$). Then for $s\in(0,1)$ we have $\left|\frac{\mathrm{d}^2}{\mathrm{d}t^2}\Sigma_c((1-t)\underline{\mathbf{p}}+t\mathscr{H}(\underline{\mathbf{p}}))\big|_{t=s}\right| \le \frac{4(\alpha^2+1)}{QW}$.*

Proof. Let $\underline{\mathbf{p}}' = (1-t)\underline{\mathbf{p}}+t\mathscr{H}(\underline{\mathbf{p}})$. Then the second derivative at $t=s$ ca be rewritten as $\frac{\mathrm{d}^2}{\mathrm{d}s^2}\Sigma_{\mathrm{c}}(\underline{\mathbf{p}}'+s\delta\underline{\mathbf{p}})\Big|_{s=0}$, where $\delta\underline{\mathbf{p}}$ is the same as in the calculation of the first derivative in the previous lemma, namely $\mathscr{H}(\underline{\mathbf{p}})-\underline{\mathbf{p}}$.

Repeating the same kind of reasoning as before, this is done by extracting the coefficient of $\frac{1}{2}s^2$ in the power series expansion of $\Sigma_{\mathrm{c}}(\underline{\mathbf{p}}'+s\delta\underline{\mathbf{p}})$ around $s=0$. We obtain

$$\begin{aligned}&\frac{\mathrm{d}^2}{\mathrm{d}s^2}\Sigma_{\mathrm{c}}(\underline{\mathbf{p}}'+s\delta\underline{\mathbf{p}})\Big|_{s=0} =\\ &=\sum_{z'=1}^{2L-W+1}\mathbb{E}_d\,d(d-1)\int_{\left[0,\frac{1}{Q}\right]^d}\prod_{i=1}^{2}\left(\frac{1}{W}\sum_{w=0}^{W-1}\mathrm{d}\delta\mathbf{p}_{z'-w}(x_i)\right)\prod_{i=3}^{d}\left(\frac{1}{W}\sum_{w=0}^{W-1}\mathrm{d}\mathbf{p}'_{z'-w}(x_i)\right)\log g(x_1,\dots,x_d)\\ &\quad+\alpha\sum_{z=1}^{2L}\int_{[0,1/Q]^2}\mathrm{d}\delta\mathbf{p}_z(x_1)\,\mathrm{d}\delta\mathbf{p}_z(x_2)\log(1-Qx_1x_2).\end{aligned} \tag{3.48}$$

We begin with an observation that will allow us to obtain bounds for both sums. We prove that if $\underline{\mathbf{p}}$ is a fixed point of $\mathscr{G}\circ\mathscr{F}$ and $\underline{\mathbf{u}}$, $\underline{\mathbf{v}}$ are any position-monotone profile satisfying $\mathscr{H}(\underline{\mathbf{u}})\preceq\underline{\mathbf{v}}\preceq\underline{\mathbf{u}}$, then for any $w\in\{0,\dots,W-1\}$ and any $L+w\le M\le 2L$,

$$\left|\sum_{z=1}^{M}\int_{[0,1/Q]^2}\mathrm{d}\delta\mathbf{p}_{z-w}(x_1)(\mathrm{d}\mathbf{u}_z(x)-\mathrm{d}\mathbf{v}_z(x))\log(1-Qx_1x)\right| \le \frac{4}{QW}. \tag{3.49}$$

We use the expansion $\log(1-Qx_1x) = \sum_{j\le 1} \frac{1}{j} Q^j x_1^j x^j$.

$$\sum_{z=1}^{M} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_{z-w}(x_1)(\mathrm{d}\mathbf{u}_z(x) - \mathrm{d}\mathbf{v}_z(x)) \log(1-Qx_1x) = \tag{3.50}$$

$$= \sum_{z=1}^{M} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_{z-w}(x_1)(\mathrm{d}\mathbf{u}_z(x) - \mathrm{d}\mathbf{v}_z(x)) \sum_{j\ge 1} \frac{1}{j} Q^j x_1^j x^j \tag{3.51}$$

$$= \sum_{j\ge 1} \frac{Q^j}{j} \sum_{z=1}^{L+w} \left(\int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}_{z-w}(x_1) x_1^j \right) \left(\int_{[0,1/Q]} (\mathrm{d}\mathbf{u}_z(x) - \mathrm{d}\mathbf{v}_z(x)) x^j \right). \tag{3.52}$$

where in the last step we limited the scope of the sum to $1 \le z \le L+w$; this is because for $z > L+w$ we have $\delta\mathbf{p}_{z-w} = 0$.

We next use the fact that $\underline{\mathbf{p}}$ is a fixed point of $\mathscr{G}\circ\mathscr{F}$. Thus for $1 \le z \le L$ we have that $\mathbf{p}_z = \left[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})\right]_z$, and since the right hand side is an average (see (3.37)), there are densities $\mathbf{a}_z$ so that $\mathbf{p}_z = \frac{1}{W}\sum_{w'=0}^{W-1} \mathbf{a}_{z-w'}$.

For differences of densities at neighboring positions the terms in the middle cancel:

$$\mathrm{d}\delta\mathbf{p}_z = \mathbf{p}_{z-1} - \mathbf{p}_z = \frac{1}{W}(\mathbf{a}_{z-W} - \mathbf{a}_z).$$

This allows us to obtain the bound

$$\left| \int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}_z(x) x^j \right| \le \frac{2}{WQ^j}. \tag{3.53}$$

Note that degradedness implies ordering of the moments, so that $\mathbf{p} \preceq \mathbf{p}'$ implies $\int \mathrm{d}\mathbf{p}(x)x^j \le \int \mathrm{d}\mathbf{p}'(x)x^j$. This together with the position-monotonicity of the profile $\underline{\mathbf{p}}$ and the bound above allow us to write

$$\left| \sum_{j\ge 1} \frac{Q^j}{j} \sum_{z=1}^{L+w} \left(\int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}_{z+w}(x_1) x_1^j \right) \left(\int_{[0,1/Q]} (\mathrm{d}\mathbf{u}_z(x) - \mathrm{d}\mathbf{v}_z(x)) x^j \right) \right| \le \tag{3.54}$$

$$\le \sum_{j\ge 1} \frac{Q^j}{j} \sum_{z=1}^{L+w} \left(\int_{[0,1/Q]} (\mathrm{d}\mathbf{u}_z(x) - \mathrm{d}\mathbf{v}_z(x)) x_1^j \right) \frac{2}{WQ^j} \tag{3.55}$$

$$\le \sum_{j\ge 1} \frac{2}{Wj} \int_{[0,1/Q]} \mathrm{d}\mathbf{p}_{L+w}(x_1) x_1^j \le \sum_{j\ge 1} \frac{2}{WjQ^j} \le \frac{4}{WQ}, \tag{3.56}$$

which finishes the proof of (3.49).

We now turn back to (3.48) and bound the two sums. The second sum can be easily seen to fit the pattern of (3.49) with $w = 0$, $\underline{\mathbf{u}} = \underline{\mathbf{p}}$, $\underline{\mathbf{v}} = \mathscr{H}(\underline{\mathbf{p}})$ and $M = 2L$.

For the first sum in (3.48) more processing is needed, but the idea remains the same. To simplify notation, we introduce $\underline{\tilde{\mathbf{p}}}$ by $[\tilde{\mathbf{p}}]_z = \frac{1}{W}\sum_{w=0}^{W-1} \mathbf{p}_{z-w}$. One can immediately see that if $\underline{\mathbf{p}}$ is position-monotone, so is $\underline{\tilde{\mathbf{p}}}$.

We use the first part of Lemma 35 to get

$$\begin{aligned}
&\sum_{z'=1}^{2L-W+1} \mathbb{E}_d d(d-1) \int_{[0,1/Q]^d} \prod_{i=1}^{2} \left(\mathrm{d}\delta\tilde{\mathbf{p}}_{z'}(x_i)\right) \prod_{i=3}^{d} \left(\mathrm{d}\tilde{\mathbf{p}}'_{z'}(x_i)\right) \log g(x_1,\dots,x_d) \\
&= \mathbb{E}_d d(d-1) \sum_{z'=1}^{2L-W+1} \int_{[0,1/Q]^2} \mathrm{d}\delta\tilde{\mathbf{p}}_{z'}(x_1)\, \mathrm{d}\Big(\mathscr{F}^{(d-1)}(\tilde{\mathbf{p}}_{z'-1}, \tilde{\mathbf{p}}'_{z'},\dots,\tilde{\mathbf{p}}'_{z'}) - \\
&\qquad - \mathscr{F}^{(d-1)}(\tilde{\mathbf{p}}_{z'}, \tilde{\mathbf{p}}'_{z'},\dots,\tilde{\mathbf{p}}'_{z'})\Big)(x) \log(1-Qx_1x) \\
&= \mathbb{E}_d d(d-1) \frac{1}{W} \sum_{w=0}^{W-1} \sum_{z'=1}^{2L-W+1} \int_{[0,1/Q]^2} \mathrm{d}\delta\mathbf{p}_{z'-w}(x_1)\, \mathrm{d}\Big(\mathscr{F}^{(d-1)}(\tilde{\mathbf{p}}_{z'-1}, \tilde{\mathbf{p}}'_{z'},\dots,\tilde{\mathbf{p}}'_{z'}) - \\
&\qquad - \mathscr{F}^{(d-1)}(\tilde{\mathbf{p}}_{z'}, \tilde{\mathbf{p}}'_{z'},\dots,\tilde{\mathbf{p}}'_{z'})\Big)(x) \log(1-Qx_1x).
\end{aligned}$$

At this point we can apply again (3.49), this time with $\mathbf{u}_z = \mathscr{F}^{(d-1)}(\tilde{\mathbf{p}}_{z'-1}, \tilde{\mathbf{p}}'_{z'},\dots,\tilde{\mathbf{p}}'_{z'})$, $\mathbf{v}_z = \mathscr{F}^{(d-1)}(\tilde{\mathbf{p}}_{z'}, \tilde{\mathbf{p}}'_{z'},\dots,\tilde{\mathbf{p}}'_{z'})$ and $M = 2L-W+1$.

It can be checked coordinate-wise that $\mathscr{H}(\underline{\mathbf{u}}) \preceq \underline{\mathbf{v}} \preceq \underline{\mathbf{u}}$ using Lemma 28.

We conclude that the first sum in (3.48) is bounded in absolute value by $\mathbb{E}_d d(d-1)\frac{4}{QW} = \frac{4\alpha^2}{QW}$, which together with the bound on the second sum prove the claim. □

3.6.6 The main argument: $\alpha > \alpha_s$

Lemma 49. *Under the increasing ϕ hypothesis, if $\Delta\Sigma < 0$, for $L > L_0(\Delta\Sigma)$, the profile $\mathscr{F}_c^{(\infty)}(\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}})$ is a nontrivial fixed point of $\mathscr{F}_c$.*

Proof. Let $\mathbf{p}^*$ be a fixed point of Σ such that $\Sigma(\mathbf{p}^*) = \Delta\Sigma$. Such a fixed point exists because of Lemma 37. Moreover, there is L_0 such that for $L > L_0$, the profile $\underline{\mathbf{p}^*}$ given by $\mathbf{p}^*_z = \mathbf{p}^*$ for all $1 \le z \le 2L$ has the property that $\Sigma_{\mathrm{c}}(\underline{\mathbf{p}^*}) < 0$. This can be seen from the fact that contributions to $\Sigma_{\mathrm{c}}(\underline{\mathbf{p}^*})$ from the middle of the chain are negative (in fact equal to $\Delta\Sigma$), and these dominate the sum.

We have that $\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}^*}) \preceq \underline{\mathbf{p}^*}$. This occurs because at each position $[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}^*})]_z$ is a convex combination of $\boldsymbol{\delta}_{\mathbf{0}}$ and $\mathscr{F}(\mathbf{p}^*) = \mathbf{p}^*$. $\boldsymbol{\delta}_{\mathbf{0}} \preceq \mathbf{p}^*$, we get $[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}^*})]_z \preceq [\underline{\mathbf{p}^*})]_z$.

This means that the sequence $\mathscr{F}_{\mathrm{c}}^{(n)}(\underline{\mathbf{p}^*})$ is monotone with respect to degradation and so the limit $\mathscr{F}_{\mathrm{c}}^{(\infty)}(\underline{\mathbf{p}^*})$ exists. The complexity functional is itself monotone (see Lemma 50 below), so $\Sigma(\mathscr{F}_{\mathrm{c}}^{\infty}(\underline{\mathbf{p}^*})) < \Sigma(\underline{\mathbf{p}^*}) < 0$. Consequently, $\mathscr{F}_{\mathrm{c}}^{(\infty)}(\underline{\mathbf{p}^*})$ is not the trivial fixed point $\underline{\boldsymbol{\delta}_{\mathbf{0}}}$, since the latter has complexity 0.

Observe now that $\mathscr{F}_{\mathrm{c}}^{(n)}(\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}})$ is lower-bounded in degradation by $\mathscr{F}_{\mathrm{c}}^{(n)}(\underline{\mathbf{p}^*})$. Thus $\mathscr{F}_{\mathrm{c}}^{(n)}(\underline{\boldsymbol{\delta}_{\mathbf{1/Q}}})$ must be a nontrivial fixed point. □

Lemma 50. *Let $\underline{\mathbf{p}} \in \mathfrak{M}^{\{1,\dots,2L\}}$ be a profile such that $\mathscr{F}_c(\underline{\mathbf{p}}) \preceq \underline{\mathbf{p}}$ or $\mathscr{F}_c(\underline{\mathbf{p}}) \succeq \underline{\mathbf{p}}$. Then $\Sigma_c(\underline{\mathbf{p}}) \le \Sigma_c(\mathscr{F}_c(\underline{\mathbf{p}}))$ or $\Sigma_c(\underline{\mathbf{p}}) \ge \Sigma_c(\mathscr{F}_c(\underline{\mathbf{p}}))$, respectively.*

Proof. Assume that $\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}) \preceq \underline{\mathbf{p}}$, since the other case will be similar. We interpolate between $\underline{\mathbf{p}}$ and $\Sigma_{\mathrm{c}}(\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}))$ by setting $\underline{\mathbf{p}}' = (1-t)\underline{\mathbf{p}} + t\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})$. It is enough to show that $\frac{\mathrm{d}}{\mathrm{d}t}\Sigma_{\mathrm{c}}((1-t)\underline{\mathbf{p}} + t\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})) \preceq 0$ at every $t \in (0,1)$.

We set $\delta\underline{\mathbf{p}} = \mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}) - \underline{\mathbf{p}}$, and carry out the computations of the first directional derivative in the same way as in the proof of Lemma 47, where we derived (3.47):

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\Sigma_{\mathrm{c}}((1-t)\underline{\mathbf{p}} + t\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})) = \\
&= \frac{\mathrm{d}}{\mathrm{d}s}\Sigma_{\mathrm{c}}(\underline{\mathbf{p}}' + s\delta\underline{\mathbf{p}})\Big|_{s=0} \\
&= \alpha \sum_{z=1}^{2L} \int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}_z(x_1) \int_{[0,1/Q]} \mathrm{d}\left(\left[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}')\right]_z - \mathbf{p}'_z\right)(x) \log(1 - Qx_1x)\,.
\end{aligned}$$

We expand $\log(1-Qx_1x)$ as $-\sum_{j\geq 1}\frac{Q^j x_1^j x^j}{j}$, and rewrite the first derivative as

$$\alpha \sum_{z=1}^{2L} \sum_{j\geq 1} \frac{Q^j}{j} \left(\int_{[0,1/Q]} \left(\mathrm{d}[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}})]_z - \mathrm{d}\mathbf{p}_z\right)(x) x^j\right)\left(\int_{[0,1/Q]} \left(\mathrm{d}[\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}')]_z - \mathrm{d}\mathbf{p}'_z\right)(x) x^j\right).$$

The moments of two distributions that are degraded w.r.t. each other are ordered in the same fashion, so the two integrals above are both positive, which proves the claim. We already assumed that $\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}) \preceq \underline{\mathbf{p}}$. The fact that $\mathscr{F}_{\mathrm{c}}(\underline{\mathbf{p}}') \preceq \underline{\mathbf{p}}'$ is easily seen to follow from the monotonicity of convex combinations with respect to degradation (see the discussion at the end of Section E.1). □

3.7 Numerical simulations and results

So far we are only able to understand distributional fixed points of equationslike (3.16) through numerical analysis. In the 1-RSB framework, this can prove challenging because of two-layer distributions that appear for $0 < m < 1$. It becomes even more computationally expensive when spatial coupling is involved, because when simulating a coupled chain, we need to keep separate distributions for each position. In this section we present the numerical results, together with a number of practical observations about the implementation.

3.7.1 Practical observations

Since we typically have no idea of the typical shape of the distributions $\mathbb{v}(\boldsymbol{\mu})$ and $\mathscr{P}(\mathbb{v})$, they will be kept as *populations* of samples. In particular, the distribution $\mathscr{P}$ will need to be represented by a population of populations. We try to reach the fixed point of (3.16) by first initializing $\mathscr{P}^0$ with Dirac deltas on the Dirac deltas on corners $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_Q\}$ of $\boldsymbol{\Delta}_Q$, and then iterating (3.16) to go from $\mathscr{P}^t$ to $\mathscr{P}^{t+1}$. We need to keep track of the populations as the iterations progress, a technique called *population dynamics*. For the special cases $m = 0$ and $m = 1$, one single

level of population dynamics is enough, which makes it much easier to compute with high accuracy.

Analyzing freezing requires some extra precautions. Let us call *hard fields* the values of $\boldsymbol{\mu}$ that have all mass on one single corner of $\boldsymbol{\Delta}_Q$. To be able to measure the amount of frozen variables correctly, one needs to keep track of the hard fields separately. Note that in order to generate hard fields under $\mathscr{P}^{t+1}$, there need to be hard fields present in $\mathscr{P}^t$. This is because a hard field is produced when all incoming messages are hard fields, except one. As the iterations progress, it can be that marginals are produced which are so close to a hard field that they become numerically indistinguishable. If we were not to explicitly label the "true" hard fields by some other means, they could be confused with the latter. This is important, since the latter do not correspond to frozen variables. This dichotomy does not, however, influence the recursion (3.16) or the computation of the entropic quantities.

Sampling the new populations can be made more difficult by re-weighting factors. An equation like (3.16) is easy to handle, since for each member of the new population on the left-hand side we need to choose d independent random individuals of the former population, and then combine them. However, the rule of calculating the new sample ν from the set of d old samples $\nu_1,\dots,\nu_d$ is not so simple to implement. These samples are in themselves populations of marginals. Differently from the case of the ν's, it is not enough to sample d independent marginals $\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d$ from $\nu_1,\dots,\nu_d$, then combine them using $\boldsymbol{\mu}=\mathbf{f}(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)$, and declare $\boldsymbol{\mu}$ to be a representative sample of ν. This does not work because we assumed we pick $\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d$ independently. However, the probability that $\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d$ occur is amplified by a factor $z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m$ (see (3.15)).

To fix this, one idea can be to (i) choose $\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d$ independently; then (ii) compute $\boldsymbol{\mu}=\mathbf{f}(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)$ and the re-weighting factor $z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m$ and then (iii) with probability $\frac{z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m}{A}$ keep $\boldsymbol{\mu}$ as a sample of the new population ν, otherwise discard it. Here A is a constant sufficiently high so that $\frac{z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m}{A}$ is always at most 1. Steps (i), (ii), (iii) are repeated until we obtain the desired number of samples. This strategy can be improved, particularly in the case where we do not know apriori the typical values of the re-weighting factor. A more adaptive strategy is the following: set $A=0$ in the beginning; then to obtain samples do (i) and (ii) as before and then if $z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m\le A$ follow up with (iii), otherwise let $\gamma=A/z(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_d)^m$ and delete each of the samples we already possess with probability γ independently. This can improve the runtime by a factor A_2/A_1 where A_2 is the value of A at the end of the second method, and A_1 is the value used in the first method.

In the case of coupled systems, we need to maintain a population at each position, in order to model $\{\mathscr{P}_z\}$. For the special values $m=0$ and $m=1$ it is perfectly feasible to simulate a lengthy chain. For arbitrary m, one can resort to various ways to diminish the chain length. For example, the chain can be ended on only one side, while at the other end all edges that normally connect beyond the boundary are connected to the last position. In effect, we set $\mathscr{P}_z=\mathscr{P}_L$ for all $z>L$. This chain will tend to descend (i.e. move towards the trivial fixed point)

faster than the open ended chain with $L \to \infty$ would, but the differences are negligible even when L is a small multiple of W.

There is also the question of how should the total and cluster free entropies be computed for a coupled chain. The "true" free entropies of the simulated chain are those obtained by adding all contributions from all positions of the chain. However, this is not a very meaningful quantity. If the chain is short, the effects of the boundary would be substantial. The right quantities would only be obtained once $L \to \infty$. Using the one-sided chain idea presented in the previous paragraph, the total free entropy and the cluster free entropy can then be computed at position L, which has the role of a proxy for the typical position in the middle of long chain.

3.7.2 Numerical results

The rigorous results are complemented by simulations that show that threshold saturation permeates many aspects relating to the phase transition phenomenology of random CSPs. Among the effects that remind of some form of threshold saturation we count the following. The global picture for $Q = 4$ is provided in Figure 3.3.

- The dynamic threshold α_d moves to the condensation threshold α_c.[8] In other words, for all $\alpha \in (\alpha_d, \alpha_c)$ we observe the equality of the curves $\Phi^{c(m=1)}(\alpha)$ and $\phi^{c(m=1)}(\alpha)$ computed for the coupled problem. This seems to indicate that asymptotically all valid colorings are grouped in a finite amount of clusters, even in this regime where in the uncoupled scenario the solution space shatters into exponentially many clusters. If the geometric interpretation of clustering remains correct for the coupled problem, the solution space of the problem is still highly connected.

  One can imagine the following thought experiment that could potentially explain this effect. Consider a graph that is coupled on a circular chain. This graph will behave in all respects like an uncoupled graph: since there is no boundary where nodes have a smaller degree, every neighborhood of every vertex is distributed in the same manner. Clustering is related to long-range correlations: taking a neighborhood of a vertex v of increasing depth T and fixing the nodes at distance T to arbitrary values can affect the marginal at v. Statistical physics lore suggests that clustering is present if and only if this marginal is not asymptotically almost surely (a.a.s) uniform. We call this *presence of long-range correlations*. In the uncoupled graph and the circularly coupled one the neighborhoods are distributed in the same way, so the same type of clustering occurs. This is no more true in the coupled scenario with an open boundary. The typical neighborhood of the latter can be obtained in the following way: take a typical neighborhood in the circularly coupled graph, together with position labels; fix apriori

[8] To be exact, one would need to have larger and larger W in order to see that the thresholds fully coincide. But already for $W = 4$ it is hard to see a difference. Running a more time consuming simulation could in principle reveal remnant small gaps between the thresholds. This phenomenon has already been observed in [HMU13]

two consecutive positions where to cut the circular chain; then prune the neighborhood by deleting exactly those edges that cross the cut. If T is much bigger than the chain size L, the removal of edges will have a big effect on the marginal at v: in fact they will free v of any influence from the boundary and cause the marginal to be a.a.s. uniform. One can therefore conjecture that as $W \to \infty$, the location where long range correlations appear is α_c.

- The above behaviour can be replicated for all values $m \in (0,1)$. In this setting, we consider m fixed and look at the functions $\Phi^{(m)}(\alpha)$ and $\phi^{(m)}(\alpha)$ as α varies. In a neighborhood of the point $\alpha_*^{(m)}$ where the two are equal the complexity function changes sign from positive to negative. When spatial coupling is used, the two entropic quantities are equal whenever $\alpha < \alpha_*^{(m)}$, and equal to the total free entropy of the uncoupled case. On the other hand, for $\alpha > \alpha_*^{(m)}$ the total and cluster free entropies do not change when spatial coupling is used.

- For $m \in (0,1)$, in the spatially coupled scenario freezing is never observed below $\alpha_*^{(m)}$. This, together with the threshold saturation of the SP threshold to the coloring threshold α_s suggests that in the coupled coloring problem (where now m^* is the right one at every α), freezing is never experienced at all, except possibly at the location of the coloring threshold α_s. An important observation is that the quantities $\alpha_f^{(m)}$ viewed as a function of m have a limit as $m \to 0$ different than $\alpha_f^{(0)}$. This is true in both coupled and uncoupled scenarios.

3.8 Conclusions and open problems

We conclude by mentioning that the picture presented in Figures 3.3 and 3.4 is still very far away from being mathematically rigorous.

The only parts that are proven correct so far are for $m = 1$: the location of α_c and the total free entropy for $\alpha < \alpha_c$ (see [BCOH$^+$14]; the authors show in this paper also that α_c is a point of non-analyticity for $\Phi(\alpha)$) and the location of $\alpha_f^{(m=1)}$ (see [Mol12]). All these results hold only for $Q > Q_0$ for a fixed but rather large Q_0. The freezing threshold on the planted model is determined for all $Q \geq 3$. In the case of the coupled planted model, we have determined in Section 3.4 a lower bound on the freezing threshold.

This does not mean all parts of the picture at $m = 1$ are fully understood. Clustering does not yet have a solid geometric interpretation, and the position of α_d is not determined rigorously. The reconstruction framework of [MM06] provides a model for this where properties can be proven, but the link between this model and actual properties of random CSPs remain conjectured. The first observations of threshold saturation of α_d were offered in [HMU13]. A problem that remains open is whether in the reconstruction framework one can prove threshold saturation. One could envision a proof along the lines presented in the proof of saturation of the SP threshold. The missing pieces are an ordering of marginals $\boldsymbol{\mu}$ in such a way

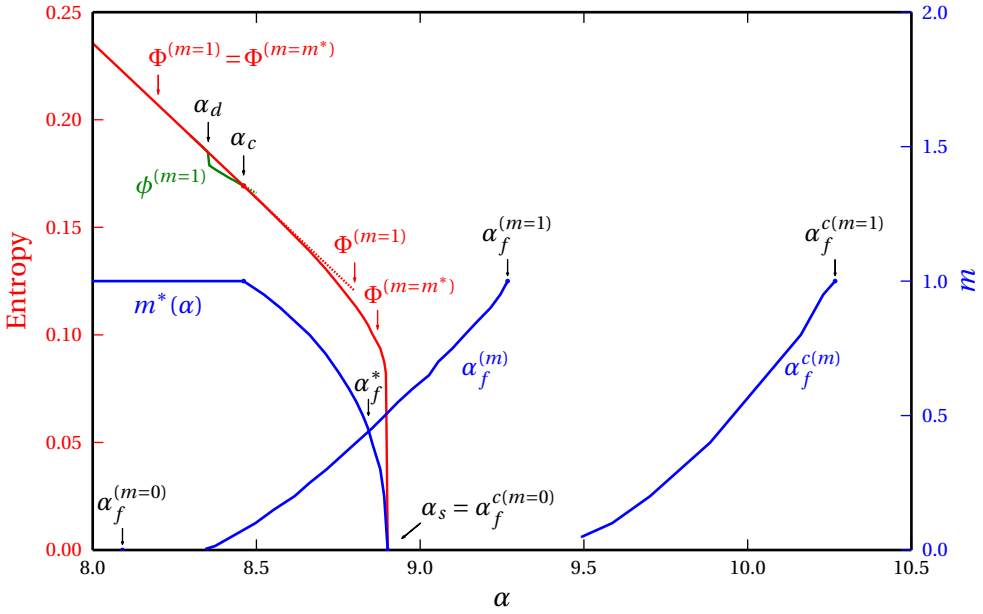


Figure 3.3 – A global picture of all thresholds that appear for $Q = 4$. The solid red curve represents the total free entropy $\Phi(\alpha)$ of the solution set. A phase transition occurs at α_c, where $\Phi(\alpha)$ is non-analytic (the analytic continuation beyond α_c is shown with a dotted line). For $\alpha < \alpha_c$, the $m = 1$ equations were used to determine the total free entropy $\Phi^{(m=1)}$ (the red curve) and the free entropy per cluster $\phi^{(m=1)}$ (the green curve). Beyond α_c, the true value of $\Phi(\alpha)$ is given by the RSB computation with a particular value of m, which is plotted as $m^*(\alpha)$. The values $m^*(\alpha)$ are computed by iterating (3.16) at pairs (m, α) in order to find $\Phi^{(m)}(\alpha)$ and $\phi^{(m)}(\alpha)$; $m^*(\alpha)$ is then determined by searching for the points where the two entropic quantities are equal. Everything done up to this point is pertaining to the uncoupled scenario. For the coupled scenario, the total free entropy (the red curve) is identical. The cluster free entropy (the green curve) undergoes saturation. This is apparent in Figure 3.4.
The freezing curves $\alpha_f^{(m)}$ and $\alpha_f^{c(m)}$ are computed by testing for freezing at many pairs (m, α) and determining the border line between the frozen and non-frozen regions.

that the recursion 3.20 descends monotonously to a fixed point. Another issue is what would be the driving potential for this recursion, and one candidate seems to be the complexity functional. While the geometry of the clusters for the uncoupled model is partly understood, especially beyond the freezing threshold ([Mol12, BCOH$^+$14]), it remains open

The picture where $m < 1$ is much less mathematically developed. Naturally, many of the open questions about clusters apply also to this setting, and much less is known. While the equations at $m = 0$ are understood fairly well, it is not clear at all if there is a direct relationship with some structures on the random graph. The SP equations have been, however, used successfully in efficient algorithms like SP-guided decimation. At $m = 0$ there are various upper and lower bounds for α_s, obtained by sophisticated versions of first and second moment methods, but they do not yet match.

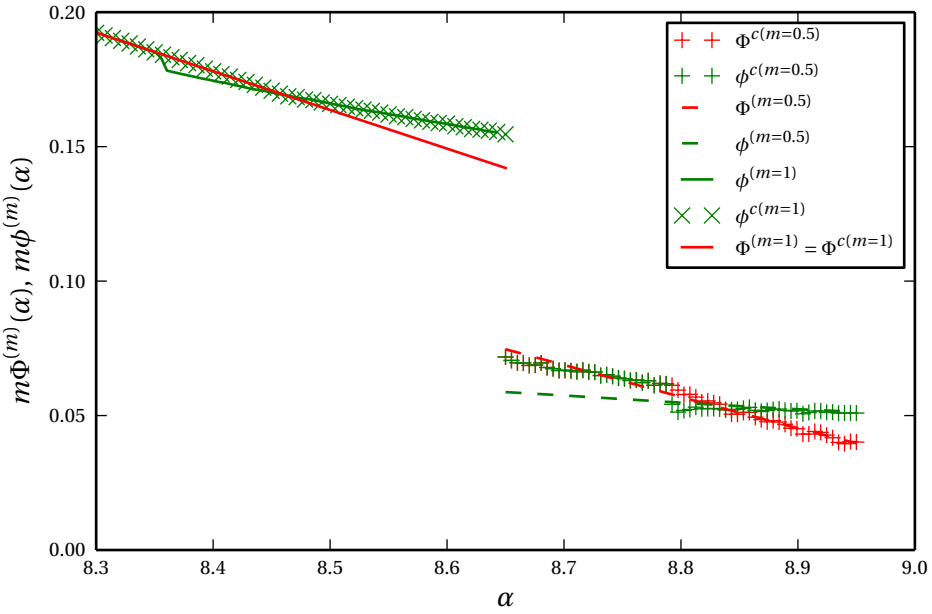


Figure 3.4 – Two pairs of plots showing the saturation of the cluster free entropy at $m = 1$ and $m = 0.5$. For $m = 1$ the numerical results are more accurate because the simplified equations for $m = 1$ are used. For each value of m the results are shown for both the uncoupled and uncoupled scenarios.

Also, the model at $m = 0$ is not obtained by simply taking the limit $m \to 0$, a fact implied by the observed discontinuity of the freezing curve $\alpha_f^{(m)}$. This is observed for both the coupled and the uncoupled models, and may suggest there may be more than one way to handle the limit $m \to 0$.

Finally, the proof we have seen in Section 3.6 for the saturation of the SP threshold holds for $Q = 3$ and is missing a (small) ingredient to turn it into a proof for the case $Q \geq 3$. Numerical tests suggest that the monotonicity of the function ϕ (Lemma 24) holds for arbitrary Q, which would imply that the result holds in general.

4 Finding solutions of random K-SAT using spatial coupling

In this chapter we investigate how spatial coupling can help search for solutions of a base instance[1] of K-SAT. The approach can be summarized as follows: (i) We start with a base instance F, the one for which we need to find a satisfying assignment. We sample a coupled instance $\tilde{F}$ which will maintain the local structure of F locally. (ii) We then run a greedy algorithm on the coupled instance $\tilde{F}$. (iii) We finally attempt to extract a solution for F from the solution (or partial solution) of $\tilde{F}$ found by the algorithm.

We examine the points above in slightly more detail:

(i) Because of the restriction of $\tilde{F}$ resembling F, this coupled construction is different from what we have seen in previous chapters. In particular, the coupled instances we are considering now are not generated from scratch, but they are moulded on the structure of a *base instance*. The exact technical details of how this is achieved will be presented soon, but intuitively what happens is that one exact copy of the base instance is placed at each position. This puts copies of the same variable in the base instance in a special relationship — we call such variables *siblings*. Then we shuffle the edges between positions on the chain, but in such a way that the endpoints change only between siblings. Thus, the local structure of the base logical formula is preserved.

(ii) We will focus on an algorithm called Unit Clause Propagation (UCP), although in principle the framework could be generalized to a wider class of algorithms. UCP has the advantage that it is simple and offers a great amount of flexibility in the step where we choose the next clause to treat. We will need to adapt this so called "free" step in order to encourage the formation of a suitable solution.

(iii) This is arguably the biggest obstacle in this approach. Just having a solution for the coupled instance does not necessarily mean we could "project" it nicely into a base structure, since sibling variables may not agree for a value. Surely, if the coupled truth assignment we found is such that sibling variables take equal values (let us call this

[1] The standard, uncoupled instances of K-SAT will be referred to as *base instances*.

consensus), then the projection of the solution is feasible. The goal is to adapt the algorithm in such a way that it prefers consensus when it has the choice.

We begin with an overview of the method in Section 4.1. This is followed by a description of a special type of spatial coupling in Section 4.2. Sections 4.3 and 4.4 describe the base UCP and coupled UCP algorithms. Numerical results are presented in Section 4.5, and Section 4.6 is dedicated to conclusions.

4.1 Overview

One can imagine multiple ways in which such an algorithm could be driven towards a solution that can be "projected". For example, we could keep changing the values of variables until consensus among siblings is achieved. Here we adopt a different approach. UCP is a greedy algorithm, which does not go back on a decision taken: so if it has assigned a value to a variable, that value remains assigned. In other words, if we make a mistake somewhere, there is no way to turn back and fix it. The solution to this is to move in the chain to some other place where the values have not been decided yet. After all, we do not need consensus at all positions of the chain: it is enough if it emerges for all vertices inside some small region which is $2W-1$ positions wide.

This suggests the following strategy: run the algorithm in such a way so that values that have already been decided are always situated in the left part of the chain, while the right part is pristine. In this manner we are able to "turn space into time". This has the advantage that it makes it unnecessary to keep the whole chain in memory. The left part of the chain (the "past") can be forgotten if no solution for the base instance could be found there. Likewise, we can generate the right part of the chain (the "future") only when the algorithm needs to access it. We would only need to store a finite part of the chain, where the algorithm operates at the current time. This way we are not bound anymore by problems such as chain size: the algorithm can continue running on an arbitrarily long chain.

It was already observed by Hamed Hassani that on a spatially coupled formula, "vanilla" UCP performs much better than on an uncoupled formula, essentially for the same reason why Belief Propagation works much better on coupled LDPC codes: there is a boundary and a decoding wave forms. This is the kind of effect that we are after. We have, however, the additional objective of building consensus among sibling variables. For this we will modify the "free" step in UCP in two ways: whenever it previously had the choice to pick the next variable and assign a random value to it, in the modified case it will (i) seek a not-yet-decided variable as left as possible in the chain and (ii) instead of flipping a fair coin to decide the value, it should take a value that leans towards that of the majority of its siblings. While rule (i) imprints a direction on the chain, rule (ii) pushes the current assignment towards consensus.

The experiments show that, indeed, consensus can be formed, if we wait long enough. Moreover, this happens for values of α larger than the uncoupled UCP threshold (which for $K=3$

lies at 8/3). This consensus takes, however, a significant amount of time (or space) to appear: observations seem to suggest that the required length of the chain is a power law in N.[2] While this is not a tragedy from a time-complexity point of view (the runtime is still at most quadratic in N), it does, however, significantly thwart attempts at rigorous analysis. Tracking the state over such a long period of time is hard. Moreover, since we bias UCP toward consensus-building and we tend to favor free variables situated as to the left as possible takes us even further away from the independence assumptions that make UCP amenable to analysis.

For the above reasons, the only evidence we have at the moment about the performance of this method is experimental in nature. The effects observed are otherwise quite novel: they open the door to the possibility of improving algorithmic thresholds on base problems by way of spatial coupling.

The rest of the chapter is roughly divided into two parts: in the first we explain how to build the coupled instance and how to adapt UCP to build consensus among variables, while in the second we present the numerical results of the simulations.

4.2 Construction of the coupled structure

The coupling procedure we introduce now is markedly different from the one presented in Section 1.9. This is because we already start from a base instance, fixed a priori, and which serves as mould for a distribution of coupled graphs taylored specifically for this base instance.

We now describe what we exactly mean by this. We denote by V the set of variables of the base formula, which without loss of generality we can assume to be $[N] = \{1, \dots, N\}$. The variables take values in a binary alphabet $B = \{0, 1\}$. A *literal* is a variable together with a "sign", i.e. an element of $V \times B$. A *clause* is a vector of literals of size K, i.e. an element of $(V \times B)^K$. The base formula F can then be described by a vector of M *clauses* F_a, for $a \in [M]$, each literal being addressed as $F_{a,k}$, for $k \in [K]$. In the factor graph picture, edges correspond to literals in a clause: these are characterized by tuples (a, k) and linking clause nodes a with variable nodes i and having a built-in sign. An illustration of this is given in Figure 4.1. An assignment is a function $\underline{\sigma} : V \to B$. We say that $\underline{\sigma}$ satisfies F if for all clauses $a \in [M]$ there exists at least one literal (a, k) where $\sigma_i = b$ and $(i, b) = F_{a,k}$.

A coupled formula is defined in a similar manner, with the remark that variables and clauses are additionally indexed by their position on the chain. Let us assume that the chain is infinite, so the set of positions is $\mathbb{Z}$. Then variables are identified by pairs from $V \times \mathbb{Z}$, literals by tuples from $V \times \mathbb{Z} \times B$, clauses by vectors from $(V \times \mathbb{Z} \times B)^k$ and coupled formulas by vectors of clauses indexed also by position, i.e., indexed on the set $[M] \times \mathbb{Z}$. we still let i and a range over $[N]$ and $[M]$, respectively, so variables and clauses will be indexed by pairs (i, z) and (a, z).

In coupled formula edges are only allowed to connect to variables at a close-by position.

[2] For tested values of N, of the order of 10^4.

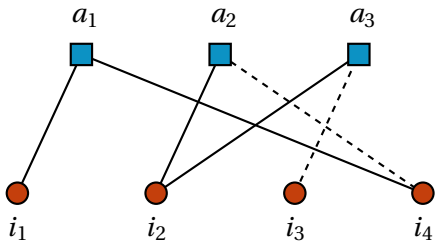


Figure 4.1 – A factor graph representation of the logical formula $(i_1 \vee i_4) \wedge (i_2 \vee \bar{i_4}) \wedge (i_2 \vee \bar{i_3})$. Positive literals (i.e. those that do not appear negated, having a sign of 1) are denoted by a full edge, while those that are negated (having a sign of 1) are dashed. In the notation of this section this formula is encoded by

$$F_{1,1} = (1,1), \quad F_{1,2} = (4,1), \quad F_{2,1} = (2,1), \quad F_{2,2} = (4,0), \quad F_{3,1} = (2,1), \quad F_{3,2} = (3,0),$$

with $K = 2, N = 4$ and $M = 3$.
One satisfying assignment for this formula is $\underline{\sigma} = (1,1,0,0)$, while $\underline{\sigma}' = (0,1,0,0)$ is not satisfying, because the first clause is violated.

We call a coupled formula $\tilde{F}$ *well formed* if it satisfies the *windowing constraints*: for all $(k,a,z) \in [K] \times [M] \times \mathbb{Z}$, the literal $\tilde{F}_{(a,z),k}$ is a tuple$((i,z'),\tau)$ satisfying $z' \in z + [W] - 1$.

We now describe how the coupled formula relates to the base instance. A coupled formula $\tilde{F}$ is said to be a *lift* of a base formula F if all edges of the coupled factor graph "project" to edges of the base factor graph; in other words, for all $(k,a,z) \in [K] \times [M] \times \mathbb{Z}$ we have that if $\tilde{F}_{(a,z),k}$ is the literal $((i,z'),\tau)$ then $F_{a,k} = (i,\tau)$. Moreover, we say that the lift is *regular* if "copies" of the same edge do not meet at variable nodes; in other words $\tilde{F}_{(a,z_1),k} \neq \tilde{F}_{(a,z_2),k}$, for all $(a,z_1,z_2,k) \in [M] \times \mathbb{Z}^2 \times [K]$.

We say that a node (i,z) or (a,z) *projects* to a node i or a in the base instace. The same can be said about edges. The *preimage* of a node i or a through the projection consists of the set of nodes (i,z) and (a,z), respectively, for $z \in \mathbb{Z}$. The nodes that differ only in the position z, i.e. they project to the same thing, will be called *siblings*. For a regular lift all sibling edges are disjoint, i.e. their target variables are all different.

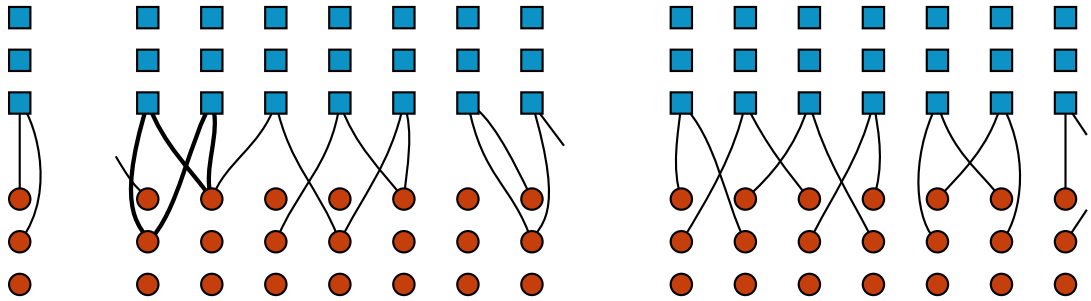
In effect, all the extra information contained in a lift can be encoded by functions $\phi^F_{a,k} : \mathbb{Z} \to \mathbb{Z}$, which we call *connection descriptors*, by setting $\tilde{F}_{(a,z),k} = (i,z',\tau)$, where $F_{a,k} = (i,\tau)$ and $z' = \phi^F_{a,k}(z)$.

A well-formed lift is one for which $\phi^F_{a,k}(z) - z \in [W] - 1$, for all $(a,z,k) \in [M] \times \mathbb{Z} \times [K]$. Such a lift is also regular if for all a and k the functions $\phi^F_{a,k}$ are bijective.

Given a base formula F, to describe a method of sampling a lift, it is sufficient to say how we sample all the connection descriptors. In both models below, all connection descriptors are chosen independently from each other.

- The Poisson random lift. Each entry $\phi^F_{a,k}(z)$ of the connection descriptors is chosen uniformly at random from $\{z,\dots,z+W-1\}$. However, the connection descriptors sampled in this way are most likely not injective, so this does not result in a regular lift. This means that the variable-node degrees are not preserved, and the factor graph $\tilde{F}$ does not look locally as F. It can be easily seen that the resulting coupled factor graph will contain many short loops. This is illustrated in Figure 4.2.
- The permutation-based random lift. Each function $\phi^F_{a,k}$ is a permutation chosen at random, satisfying the window constraint using one of the methods outlined in Appendix F. This approach ensures that the neighborhood of a node (a,z) chosen at random looks exactly as the neighborhood of node a in the base factor graph.

We consider both these models in the simulations. It appears that the Poisson random lift behaves somewhat better. This may be due to the fact that the loops which appear in the Poisson random lift are always reinforcing: if between two nodes there are two different paths in the lifted instance, then these two paths project to the same path in the base formula. This induces some positive correlation between the values of sibling variables on the two paths (note that the signs of literals are also matching on the two paths), which in turn contributes to the creation of consensus.



a) b) c)

Figure 4.2 – In a) we represent two edge of the base instance. In b) and c) we show two typical preimages of the two edges for a Poisson random lift and a permutation-based lift, respectively. The window size was set to $W = 3$. We show how the Poisson random lift can easily create loops even in places where the base graph is tree-like (a cycle is emphasized in b), while for the permutation-based lift this is not possible.

Let $\tilde{F}$ be a lift of a formula F. If a satisfying assignment $\{\sigma_{i,z}\}$ of $\tilde{F}$ is such that all sibling variables take the same value, then for each i there is a value σ_i so that for all z, $\sigma_{i,z} = \sigma_i$. Then $\{\sigma_i\}$ is a satisfying assignment for the formula F, and we also say that the assignment $\{\sigma_{i,z}\}$ projects to $\{\sigma_i\}$. Typically the lifted formula $\tilde{F}$ will have many other satisfying assignments, where preimages of i take different values. It is worth noting, however, that even in this case, if we find a window of $2W-1$ subsequent positions $z-W+1,\dots,z+W-1$ where for each variable i all its preimages at these positions take the same value under an assignment, then from this we can construct a satisfying assignment for the base instance. We call the event

when a collection of variables has the same values under an assignment $\underline{\sigma}$ a $\underline{\sigma}$-*consensus* among the said set of variables.

The goal in the rest of this chapter is to run an algorithm on $\tilde{F}$ to find an assignment for the coupled formula and try to find such a window (a *projection window*) where consensus forms among all sets of siblings. This enables us to project the solution to F.

4.3 Unit Clause Propagation

Unit Clause Propagation (UCP) is a greedy algorithm [DG84] for finding satisfying assignments of K-SAT. It works as follows. A *partial assignment* is maintained during the execution of the algorithm, under which variables have values 1, 0 and $*$ (undetermined). Initially, all variables are marked $*$. The following steps are performed iteratively until either (i) at least one clause is contradicted by the partial assignment or (ii) no stars are left in the partial assignment:

1. *Forced step.* A unit clause is one which satisfies two conditions: (i) Exactly one variable v among the K is starred; all the rest are determined and (ii) whether the clause is satisfied or not depends on the value that this variable takes. If there is at least one unit clause, we are forced to satisfy it, setting v to the matching value. In this case we call v a *critical variable*
2. *Free step.* If there is no unit clause, we choose a starred variable at random and assign it a random binary value.

At each iteration, one more variable is determined, so the algorithm runs in linear time (checking whether clauses fall in the categories above has an amortized cost). Unit clause propagation performs as follows: for $\alpha < \alpha_{\text{UCP-low}}$, then it succeeds w.h.p. For $\alpha \in (\alpha_{\text{UCP-low}}, \alpha_{\text{UCP-high}})$, it succeeds with $\Theta(1)$ probability, while for $\alpha > \alpha_{\text{UCP-high}}$ it fails w.h.p. The values for $\alpha_{\text{UCP-low}}$ and $\alpha_{\text{UCP-high}}$ were determined for $K = 3$ in [CR92] and [Ach01] to be 2/3 and 8/3, respectively. The asymptotic behaviour is $O\left(2^K/K\right)$ for both thresholds [CR92].

4.4 Unit Clause Propagation on the Lifted Factor Graph: turning space into time

Just running vanilla UCP on the coupled graph will typically result in a solution where sibling variables take different values. The occurrence of consensus at all variables in a projection window as discussed above is then very improbable. To change this, we will bias the algorithm in order to enhance the probability that sibling variables take the same value. This will be done by modifying the free step of UCP, both in the way it chooses the next variable to set and in the way it chooses its value.

As discussed at the beginning of the chapter we will arrange things such that the region

where the algorithm operates moves to the right along the chain as time progresses. We take advantage of this by only keeping a small part of the chain in memory, which requires us to generate a new part of the chain as it becomes necessary.

The bias that affects the value of the chosen variable in the free step takes into accout the values of σ at all siblings left of the current position. It is exponentially decaying, with a parameter $\beta < 1$, which governs how fast the bias forgets values from the past. We set

$$\text{Bias}_{(i,z)} = (1-\beta) \sum_{z'<z} \beta^{z-z'-1}(2\sigma_{i,z'} - 1).$$

The factor $(1-\beta)$ appears in order to ensure that $\text{Bias}_{(i,z)}$ is clamped in the interval $[-1,1]$. It takes the extremal values $+1$ and -1 when all siblings to the left have value 1 or value 0, respectively. If the value is $*$, we set $\sigma = 1/2$, so that it does not count towards the bias. Note that the bias can be updated online, using the previous bias $\text{Bias}_{(i,z-1)}$ and the value $\sigma_{(i,z-1)}$.

The bias can be used to compute the parameter of a Bernoulli random variable which decides the new value of the variable chosen in the free step. We pass the bias through a sigmoid function, so the parameter of the Bernoulli is given by

$$f(x;\gamma) = \frac{1}{2}(1 + |x|^{\gamma} \operatorname{sign}(x)),$$

where x is the bias. Modifying the bias in this manner turns out to be important: using a "flat" function (i.e. $\gamma = 1$) does not achieve the desired results. Values of γ closer to 0, which increase the rigidity of the decision perform well, and usually better than using the completely rigid sign function ($\gamma = 0$). For most of the experiments we use γ between 0.1 and 0.2.

The algorithm is described below. It takes as parameters the base formula F, the coupling window size W, the number of positions T on which the algorithm is supposed to work at a given time and the two parameters controling the biases β and γ. Also, the algorithm lacks the terminating condition present in standard UCP. In the coupled case, contradicting clauses are ignored, and the algorithm only stops when it finds a satisfying assignment. To prevent the algorithm from running forever, in practice we set a cutoff point at some distant position on the chain and stop the algorithm without a solution in case this is reached.

function COUPLED-UCP(F, W, T, β, γ)
  N: number of variables of F;
  M: number of clauses of F;
  $(z_L, z_H) \leftarrow (0, L)$;
  Generate $\tilde{F}_{(a,z)}$ for $a \in [M]$ and $z \in [z_L - W, z_H + W]$;
  $\sigma_{i,z} \leftarrow *$ for $i \in [N]$ and $z \in [z_L - 2W + 1, z_H + W]$;
  loop
    if $\exists$ unit clause $(a,z) \in [M] \times [z_L - W, z_H + W]$ with critical variable $(i,z') \in [N] \times [z_L, z_H]$ **then**
      Choose $\sigma_{(i,z')}$ so that the clause (a,z) is satisfied;
    else

while there is no position (i, z_L) with $\sigma_{i,z_L} = *$ **do**
if for each $i \in [N]$ there is σ-consensus among variables in $\{i\} \times [z_L - 2W - 2, z_L]$ **then**
return $\{\sigma_{i,z_L}\}_{i\in[N]}$ as a solution of F
end if
Remove from memory $\tilde{F}_{(a,z_L-W)}$ for all $a \in [M]$;
$z_L \leftarrow z_L + 1$, $z_H \leftarrow z_H + 1$;
Generate $\tilde{F}_{(a,z_H+W)}$ for all $a \in [M]$;
$\sigma_{i,z_H+W} \leftarrow *$ for $i \in [N]$;
end while
Let (i, z_L) be such that $\sigma_{i,z_L} = *$;
$$\sigma_{i,z_L} \leftarrow \begin{cases} 1, & \text{w. p. } f(\text{Bias}_i(\underline{\sigma}, \beta); \gamma) \\ 0, & \text{otherwise;} \end{cases}$$
end if
end loop
end function

Note that generating the random lift $\tilde{F}$ can be done "online". Every time we generate the clauses at a particular position we are sampling the next entry in each connection descriptor. In the case of the Poisson random lift, this is fine, since the entries in the connection descriptor are independent. In the case of the permutation-based random lift we use the Markov chain generation model presented in Appendix F.

4.5 Numerical results

We measure the runtime of the algorithm by the length of the coupling chain used by the process until a solution is found. In a typical experiment we choose values for all parameters, like α, N, etc. and run the algorithm a large number of times, in order to obtain the distribution of the coupled chain length. To visualize the outcome, we plot the cdfs, since they are particularly suited to obtain percentiles and estimates for the proportion of algorithm executions that fail.

4.5.1 Dependence on α, N and the bias decay parameter

The results of experiments to determine the threshold of the algorithm in α are presented in Figure 4.3 for both the Poisson random lift and the permutation random lift. The Poisson case exhibits the higher threshold of the two and also its threshold is somewhat sharper. We observe a range of α where for which many runs of the algorithm would not end. A separate plot showing threshold behavior for the case $K = 5$ is given in Figure 4.5.

4.5.2 Dependence on N

To determine the scaling in N, we consider the same experiment with N now increasing in powers of 2. We observe that for the range of N under consideration, the length of chain required is governed by a power law. Plots and a brief discussion of this is given in Figure 4.4.

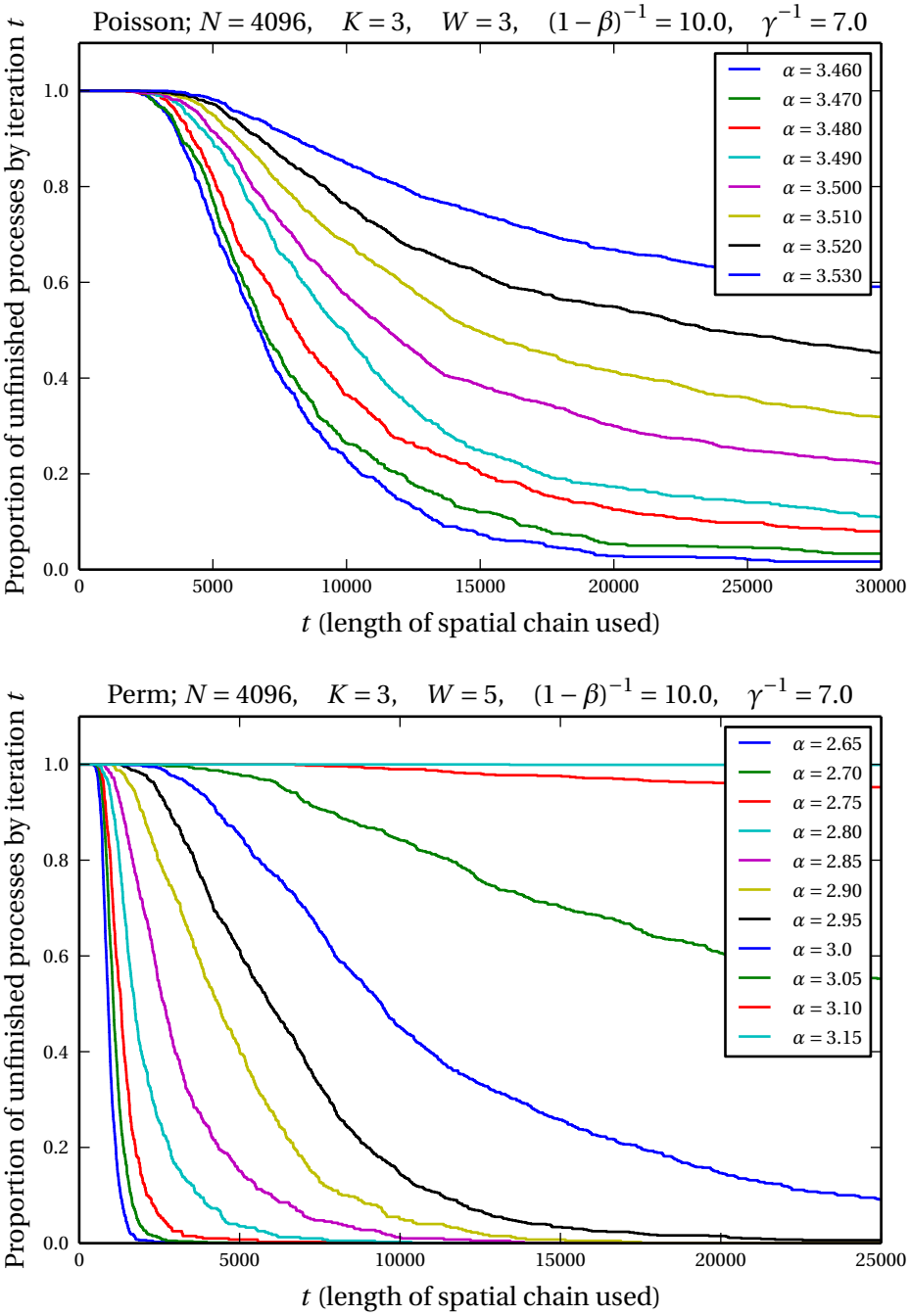


Figure 4.3 – In these two plots we show the influence of α on the runtime of the algorithm, on the top for the Poisson lift, and on the bottom for the permutation-based lift. The Poisson lift exhibits a more definite threshold at in the region $3.45 < \alpha < 3.55$. For the permutation-based lift, the transition is much smoother and also lies at a lower value of α, between around 2.7 and 3.1. We observe two phenomena, which are more clear for the Poisson case: at a first stage, around $\alpha = 3.46$, we start to see a heavy tail for the distribution of t. This shows that on many runs, the algorithm is not able to find a solution at all. At a much later stage (Poisson: 3.56, not shown; Permutations: around 3.1), the algorithm fails on all runs.

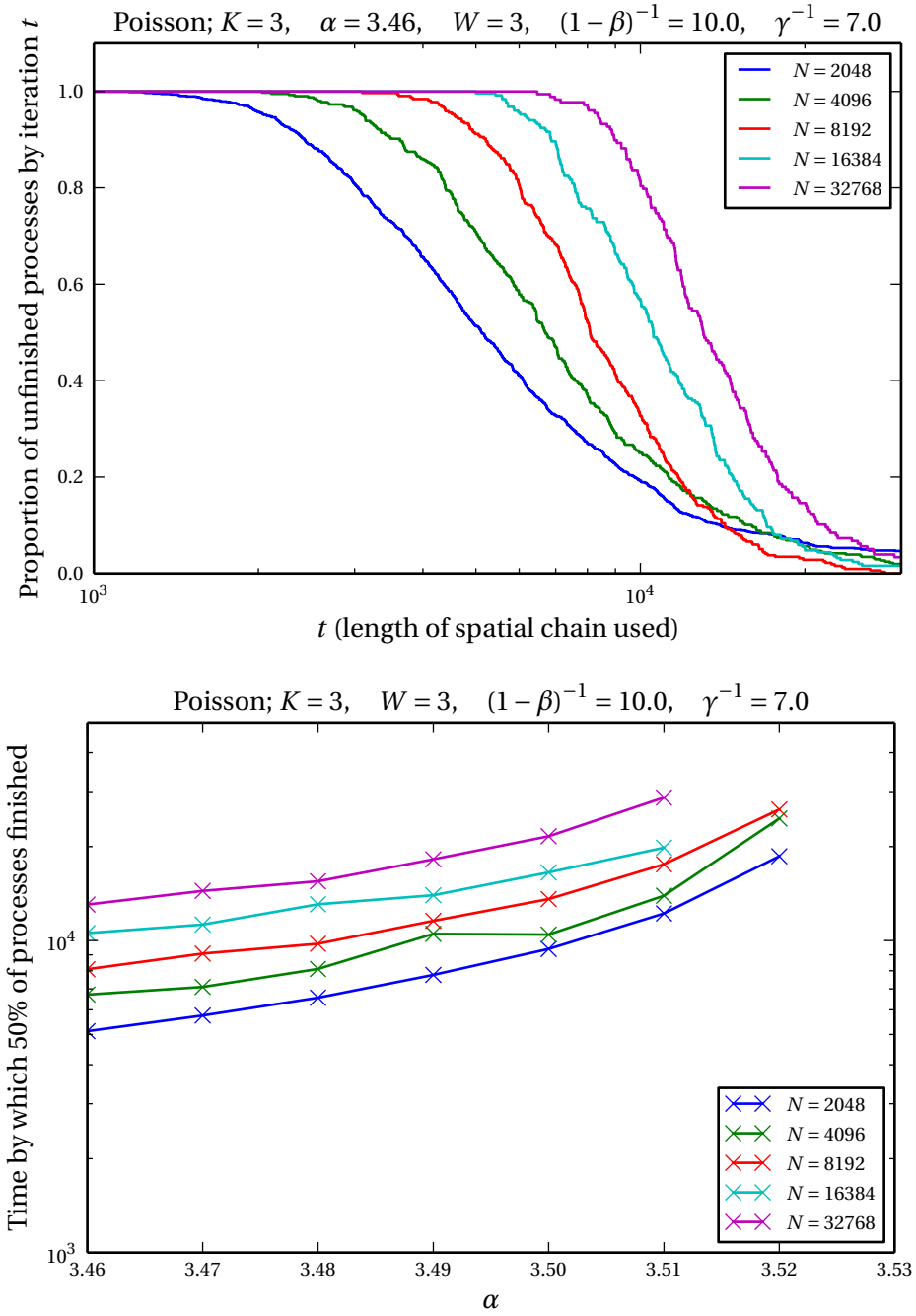


Figure 4.4 – Computation of the scaling factor of the chain length with N that increase in powers of 2 in the Poisson case. On the top we show how the distribution of chain lengths changes, while on the bottom we show the median of the chain lengths for various values of α and N. The median is not shown in the cases it exceeds the cutoff value of $3 \cdot 10^4$.
In both plots, the chain length is shown in log-scale, to emphasize the power law that emerges. This enables us to determine that for the values of N plotted the scaling is of the form $\Theta(N^\eta)$, with η around 0.4. The similarity in the shape of the curves suggests that the threshold in α is around the same place for all values of N.

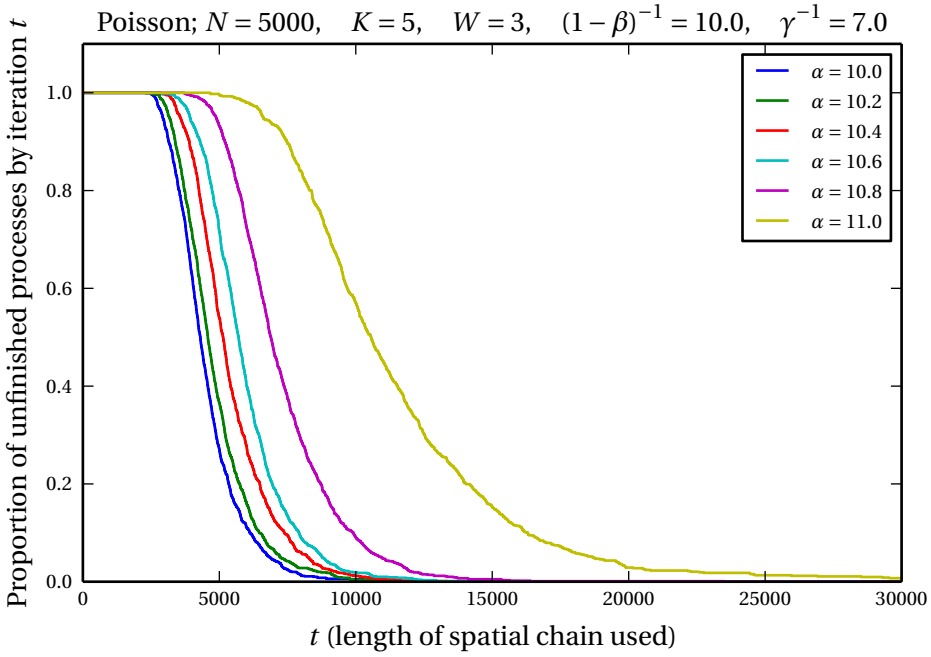


Figure 4.5 – For $K = 5$ we see roughly the same picture. Here the transition observed is for α between 10 and 11, whereas the threshold beyond which standard UCP finds no solutions for the base instance at $N = 5000$ is around 7.3. Just for comparison, the satisfiability threshold for $K = 5$ is conjectured to be at $\alpha_s = 13.67$.

To unclutter the presentation, we omit the equivalent results for the permutation random lift, as they are very similar.

4.5.3 The varying hardness of base instances

The experiments presented so far have been done by sampling base instances at random and then solving them using spatial coupling. But not all base instances share the same level of hardness. One can see this by running experiments that keep the base instance fixed. In Figure 4.6 we see that near the transition ($\alpha = 3.48$) it can even happen that some base instances can be solved all the time (all runs on that base instance are successful), while some cannot be solved at all (all runs on that base instance would run forever). Also, the variance in the chain length becomes considerably smaller once we condition on a fixed base instance. This provides a reasonable explanation for the heavy tails obtained in Figure 4.3, where some runs of the algorithm were able to finish, while others were not.

A valid question is whether running spatially coupled UCP really differentiates hard from easy instances at fixed values of α as $N \to \infty$. There is some evidence that this is not true: in Figure 4.4 we see that the distributions depicted by the cdfs tend to concentrate as N increases, and their tail becomes thinner. This may suggest that the existence of easy and hard base instances

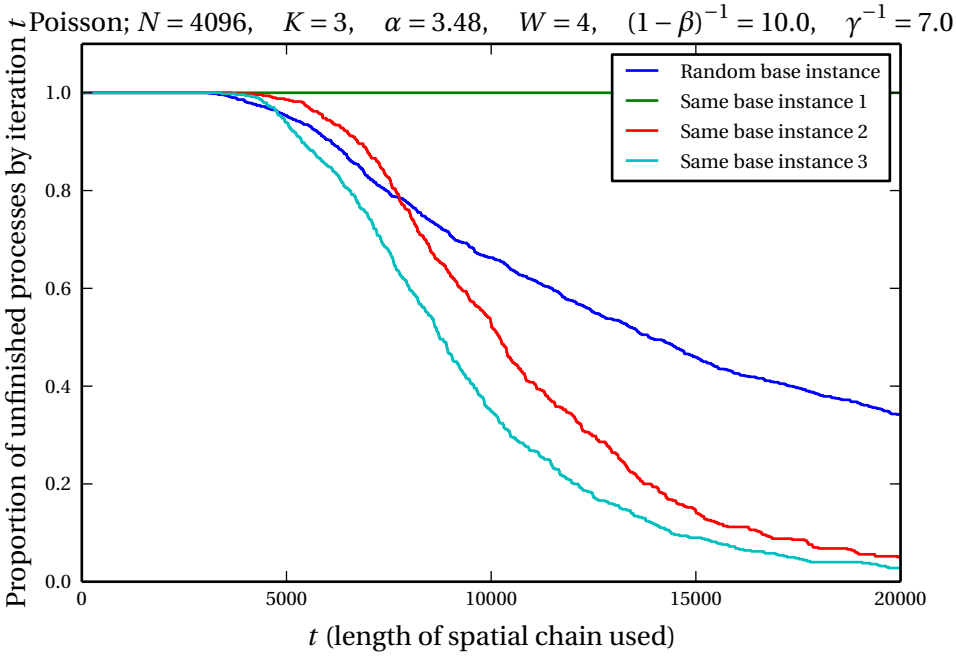


Figure 4.6 – Not all base instances are created equal: we compare the outcome of an experiment where each time a new random base instance is chosen with the case where a base instance is sampled only once at the beginning of the experiment and then reused every time. The latter experiment is repeated 3 times, for 3 different base instances. We observe for one of the base instances chosen, Coupled UCP in never able to find a solution, suggesting this instance is much harder to satisfy than the other two.

is a finite-size effect.

4.6 Concluding remarks

The main purpose of this investigation was to explore whether using spatial coupling to obtain solutions for the base instance can give useful results. We gathered some evidence that suggests that this is indeed the case, using a modified form of UCP. We overcome the need to store a large spatial chain in memory by having UCP operate on a small part of the chain at a time and moving in a definite direction along the chain. However, for the time being, little seems to suggest that a rigorous proof for this method is obtainable.

We see that much of the improvement gained depends on the local structure of the coupled factor graph. The fact that the Poisson random lift performs much better than the permutation-based random lift suggest that there may be even better ways to do the coupling. However, the improvement comes at the price of losing the local equivalence of neighborhoods between the base instance and the random lift.

We have also inferred a scaling law that relates the size of the base formula to the runtime of the algorithm (or the length of the spatially coupled chain used). It could be, however, that this law does not hold anymore for N much larger than 10^5, a place where simulations become prohibitive. This, together with the question of whether at high N there is still a dichotomy between hard and easy base instances remain open.

A The root-free expression of the Bethe free entropy on trees

Given a factor graph that is a tree, pick one root i among the variable nodes, and for each node $u \neq i$ in the graph (be it variable node or function node), let $p(u)$ be its only neighbor on the path to i. Using recursion on a tree it can be easily be checked that

$$\log Z = \log Z^i + \sum_{u \neq i} \log Z^{u \to p(u)}.$$

Since Z is independent of the choice of i, we need a formula which is not dependent on the way we choose the root. Such a formula would be readily generalizable to situations where the graph is not a tree. We first define some quantities which are not "directional".

$$Z^i = \sum_{\sigma_i} \prod_{b \in \partial i} \mu^{b \to i}(\sigma_i), \tag{A.1}$$

$$Z^a = \sum_{\sigma_{\partial a}} \psi_a(\sigma_{\partial a}) \prod_{j \in a} \mu^{j \to a}(\sigma_j), \tag{A.2}$$

$$Z^{ia} = \sum_{\sigma_i} \mu^{i \to a}(\sigma_i) \mu^{a \to i}(\sigma_i). \tag{A.3}$$

Using the following two relations, we are able to transform directional contributions to Z into direction-less ones:

$$\begin{aligned} Z^a &= \sum_{\sigma_i} \mu^{i \to a}(\sigma_i) \sum_{\sigma_{\partial a \setminus i}} \psi_a(\sigma_{\partial a}) \prod_{j \in a \setminus i} \mu^{j \to a}(\sigma_j) \\ &= \sum_{\sigma_i} \mu^{i \to a}(\sigma_i) \mu^{a \to i}(\sigma_i) Z^{a \to i} \\ &= Z^{ia} Z^{a \to i}; \end{aligned} \tag{A.4}$$

$$\begin{aligned} Z^i &= \sum_{\sigma_i} \mu^{a\to i}(\sigma_i) \prod_{b\in\partial i\setminus a} \mu^{b\to i}(\sigma_i) \\ &= \sum_{\sigma_i} \mu^{a\to i}(\sigma_i)\mu^{i\to a}(\sigma_i) Z^{i\to a} \\ &= Z^{ia} Z^{i\to a}. \end{aligned} \tag{A.5}$$

Finally, putting everything together, we obtain

$$N\Phi = \sum_i \log Z^i + \sum_a \log Z^a - \sum_{i\sim a} \log Z^{ia}.$$

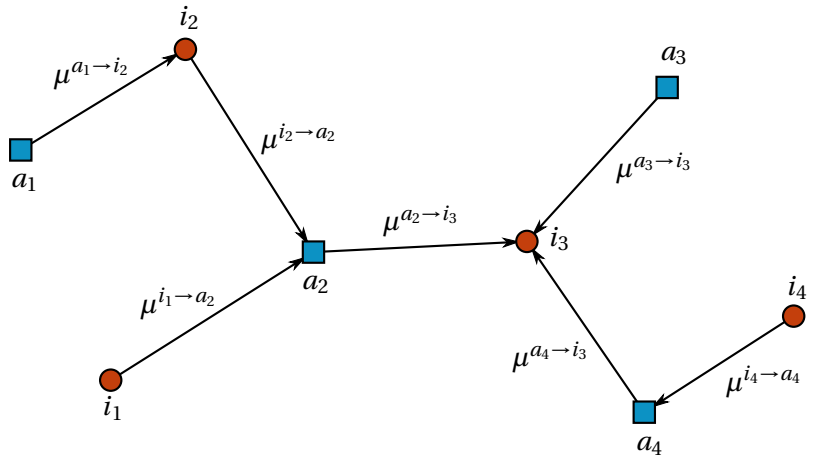


Figure A.1 – An example of factor graph, with messages that are needed to compute the marginals for i_3. The partition function can be computed by gathering all normalization constants resulting from the message processing:

$$Z \;=\; Z^{i_3} Z^{a_3\to i_3} Z^{a_2\to i_3} Z^{a_4\to i_3} Z^{i_2\to a_2} Z^{i_1\to a_2} Z^{a_1\to i_2} Z^{i_4\to a_4}.$$

Using relations (A.5) and (A.4) we obtain

$$Z \;=\; Z^{i_3} \frac{Z^{a_3}}{Z^{a_3 i_3}} \frac{Z^{a_2}}{Z^{a_2 i_3}} \frac{Z^{a_4}}{Z^{a_4 i_3}} \frac{Z^{i_2}}{Z^{i_2 a_2}} \frac{Z^{i_1}}{Z^{i_1 a_2}} \frac{Z^{a_1}}{Z^{a_1 i_2}} \frac{Z^{i_4}}{Z^{i_4 a_4}}.$$

B The belief propagation formalism and density evolution for LDPC codes on BMS channels

B.1 Message passing in terms of beliefs

The message passing equations for coding are given by (1.6) and for LDPC codes they take the form

$$\mu^{i\to a}(\sigma_i) = \frac{1}{Z^{i\to a}} e^{h_i(\sigma_i-1)} \prod_{b\in\partial i\setminus a} \mu^{b\to i}(\sigma_i), \qquad Z^{i\to a} = \sum_{\sigma_i} e^{h_i(\sigma_i-1)} \prod_{b\in\partial i\setminus a} \mu^{b\to i}(\sigma_i),$$
$$\mu^{a\to i}(\sigma_i) = \frac{1}{Z^{a\to i}} \sum_{\sigma_{\partial a\setminus i}} \frac{1+\prod_{j\in a}\sigma_j}{2} \prod_{j\in a\setminus i} \mu^{j\to a}(\sigma_j), \quad Z^{a\to i} = \sum_{\sigma_{\partial a}} \frac{1+\prod_{j\in a}\sigma_j}{2} \prod_{j\in a\setminus i} \mu^{j\to a}(\sigma_j). \tag{B.1}$$

Using the HLLR messages $\nu^{i\to a} = \frac{1}{2}\log\frac{\mu^{i\to a}(+1)}{\mu^{i\to a}(-1)}$, it is easy to see that the processing at a node i becomes

$$\nu^{i\to a} = h_i + \sum_{b\in\partial i\setminus a} \nu^{b\to i}.$$

For processing at a node a, we observe the following:

$$\begin{aligned}\mu^{a\to i}(\sigma_i) &= \sum_{\sigma_{\partial a}} \frac{1+\prod_{j\in\partial a}\sigma_j}{2} \prod_{j\in\partial a\setminus i} e^{(\sigma_j-1)\nu^{j\to a}} \mu^{j\to a}(+1) \\ &= 2^{|\partial a|-2}\left[\prod_{j\in\partial a\setminus i} e^{-\nu^{j\to a}}\mu^{j\to a}(+1)\cosh\nu^{j\to a} + \sigma_i \prod_{j\in\partial a\setminus i} e^{-\nu^{j\to a}}\mu^{j\to a}(+1)\sinh\nu^{j\to a}\right].\end{aligned}$$

Then $\frac{\mu^{a\to i}(+1)}{\mu^{a\to i}(-1)} = \frac{1+\prod_{j\in\partial a\setminus i}\tanh\nu^{j\to a}}{1-\prod_{j\in\partial a\setminus i}\tanh\nu^{j\to a}}$, and we are able to deduce

$$\tanh\nu^{a\to i} = \prod_{j\in\partial a\setminus i} \tanh\nu^{j\to a}.$$

For the terms occurring in the Bethe expression, we treat $\log Z_i$, $\log Z_a$ and $\log Z_{ia}$ separately.

We obtain

$$\begin{aligned}
\log Z_i &= \log \sum_{\sigma_i} e^{h_i(\sigma_i - 1)} \prod_{b \in \partial i \setminus a} \mu^{b \to i}(\sigma_i) \\
&= \log\left(1 + e^{-2(h_i + \sum_{a \in \partial i} \nu^{a \to i})}\right) + \sum_{a \in \partial i} \log \mu^{a \to i}(+1), \\
\log Z_a &= \log \sum_{\sigma_{\partial a}} \frac{1 + \prod_{j \in a} \sigma_j}{2} \prod_{j \in a} \mu^{j \to a}(\sigma_j) \\
&= \log \left[2^{|\partial a| - 1} \left(\prod_{j \in \partial a} e^{-\nu^{j \to a}} \cosh \nu^{j \to a} + \sigma_i \prod_{j \in \partial a} e^{-\nu^{j \to a}} \sinh \nu^{j \to a} \right) \right] + \sum_{i \in \partial a} \log \mu^{j \to a}(+1) \\
&= \log \frac{1 + \prod_{j \in \partial a} \tanh \nu^{j \to a}}{2} + \sum_{i \in \partial a} \log(1 + e^{-2\nu^{i \to a}}) + \sum_{i \in \partial a} \log \mu^{j \to a}(+1) \\
&= -\log\left(1 + e^{-2 \tanh^{-1} \prod_{j \in \partial a} \tanh \nu^{j \to a}}\right) + \sum_{i \in \partial a} \log(1 + e^{-2\nu^{i \to a}}) + \sum_{i \in \partial a} \log \mu^{j \to a}(+1), \\
\log Z_{ia} &= \log \sum_{\sigma_i} \mu^{i \to a}(\sigma_i) \mu^{a \to i}(\sigma_i) \\
&= \log(1 + e^{-2(\nu^{i \to a} + \nu^{a \to i})}) + \log \mu^{i \to a}(+1) + \log \mu^{a \to i}(+1),
\end{aligned}$$

where we used the identity $\frac{1+x}{2} = \frac{1}{1+e^{-2\tanh^{-1} x}}$.

Note that in the Bethe expression $N\Phi = \sum_i \log Z_i + \sum_a \log Z_a - \sum_{i \sim a} \log Z_{ia}$ all terms of the form $\log \mu^{i \to a}(+1)$ and $\log \mu^{a \to i}(+1)$ cancel out. This allows us to write

$$\begin{aligned}
N\Phi = &\sum_i \log\left(1 + e^{-2(h_i + \sum_{a \in \partial i} \nu^{a \to i})}\right) - \sum_a \log\left(1 + e^{-2 \tanh^{-1} \prod_{j \in \partial a} \tanh \nu^{j \to a}}\right) + \\
&+ \sum_{i \sim a} \log(1 + e^{-2\nu^{i \to a}}) - \sum_{i \sim a} \log(1 + e^{-2(\nu^{i \to a} + \nu^{a \to i})}).
\end{aligned} \tag{1.22}$$

B.2 Properties of symmetric densities

The following identities hold in fact for all h, h':

$$\frac{(1 + e^{-2h})(1 + e^{-2h'})}{1 + e^{-2h-2h'}} = \frac{2}{1 + \tanh h \tanh h'} = 1 + e^{-2 \tanh^{-1}(\tanh h \tanh h')}. \tag{B.2}$$

By taking the logarithm on both sides, and assuming h and h' are distributed according to x and y, respectively, we obtain

$$H(\mathsf{x}) + H(\mathsf{y}) = H(\mathsf{x} \circledast \mathsf{y}) + H(\mathsf{x} \boxplus \mathsf{y}). \tag{B.3}$$

C Auxilliary proofs for the interpolation method

C.1 Proof of (2.27)

Proposition 51. *Given a fixed configuration graph G whose underlying type set is m-admissible for $m > K^2$ and a fixed channel realisation h, then with the notation from the proof of Lemma 14 we have that*

$$\frac{1}{|B_\alpha|}\sum_{a\in B_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle = \frac{1}{|B'_\alpha|}\sum_{a\in B'_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle + O\left(\frac{1}{m}\right). \tag{C.1}$$

Proof. Rewrite the left hand side as

$$\frac{1}{|B'_\alpha|}\frac{|B'_\alpha|}{|B_\alpha|}\left(\sum_{a\in B'_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle - \sum_{a\in B'_\alpha\setminus B_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle\right). \tag{C.2}$$

We will first find an estimate of the quantity $|B'_\alpha \setminus B_\alpha|$, i.e. the number of (pseudo-)check constraints that connect to at least one socket multiple times. To do this, let us look at the subset of B'_α where $a_i = a_j$ (i.e. edges i and j connect to the same socket), for some distinct i, j with $1 \le i, j \le K$. The cardinality $q_{i,j}$ of this subset is 0 if $\alpha_i \neq \alpha_j$, and is equal to $|B'_\alpha|/|F_i| \le |B'_\alpha|/m$ if $\alpha_i = \alpha_j$.

A (rough) upper bound for $|B'_\alpha \setminus B_\alpha|$ is given then by sum $\sum_{i\neq j} q_{i,j}$, which in turn never exceeds $K^2|B'_\alpha|/m$.

We are now able to bound the ratio $|B'_\alpha|/|B_\alpha|$ appearing in (C.2) by $m/(m-K^2)$. Indeed, this follows from

$$\frac{|B'_\alpha|}{|B_\alpha|} = \frac{|B'_\alpha|}{|B'_\alpha| - |B'_\alpha \setminus B_\alpha|}.$$

The absolute value of the second sum in (C.2) is clearly upper-bounded by $|B'_\alpha \setminus B_\alpha|$, since the

bracket takes values between 0 and 1. Putting everything together, we obtain

$$\frac{1}{|B_\alpha|}\sum_{a\in B_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle \le \Big(\frac{m}{m-K^2}\Big)\frac{1}{|B'_\alpha|}\sum_{a\in B'_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle + \frac{K^2}{m-K^2},$$
$$\frac{1}{|B_\alpha|}\sum_{a\in B_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle \ge \frac{1}{|B'_\alpha|}\sum_{a\in B'_\alpha}\langle\sigma_a^{(1)}\cdots\sigma_a^{(r)}\rangle - \frac{K^2}{m-K^2}.$$

□

C.2 Proof of Theorem 9

We construct a smooth family of channels by interpolating between the given channel c^* and the worst channel, denoted by Δ_0 (since in the log-likelihood representation it consists of a point mass at 0):

$$\mathsf{c}_\mathsf{h} = \frac{\mathsf{h}-\mathsf{h}^*}{1-\mathsf{h}^*}\Delta_0 + \frac{1-\mathsf{h}}{1-\mathsf{h}^*}\mathsf{c}^*,$$

where $\mathsf{h}^* = H(\mathsf{c}^*)$ and the parameter h has been chosen in such a way that it coincides with $H(\mathsf{c})$, varying from h^* to 1. Also, to ease notation, for the DE fixed point we use x_h as a shorthand for $\mathsf{x}_{\mathsf{c}_\mathsf{h}}$.

The plan is as follows: first we will show that

$$\frac{d}{d\mathsf{h}}\Phi(\mathsf{x}_\mathsf{h},\mathsf{c}_\mathsf{h}) = g^{\mathrm{BP}}(\mathsf{h}). \tag{C.3}$$

Then by Theorem 7, we can replace $g^{\mathrm{BP}}(\mathsf{h})$ with $g^{\mathrm{MAP}}(\mathsf{h})$. We integrate the two sides between h^* and 1 and check that for the worst channel

$$\Phi(\mathsf{x}_1,\Delta_0) = R = \lim_{N\to\infty}\frac{1}{N}\mathbb{E}H(\underline{X}|\underline{Y}(1)),$$

thereby ending the proof of Theorem 9.

It remains to check (C.3). Note that an equivalent form of (2.6) written in the density evolution language is

$$g^{\mathrm{BP}}(\mathsf{h}) = \left[\frac{d}{d\mathsf{h}}H(\mathsf{c}_\mathsf{h}\circledast\Lambda^{\circledast}(\rho^{\boxplus}(\mathsf{x}_{\mathsf{h}'})))\right]_{\mathsf{h}'=\mathsf{h}}. \tag{C.4}$$

In the ensuing calculations, we will replace x_h by x whenever its meaning is clear from context. It can be easily checked that this form is very similar to the left hand side of (C.3), except that the differential operator only affects c and not x (i.e. it is a partial derivative). We will subsequently show that since x is the forward DE fixed point, the partial derivative equals the

total derivative.

We will compute the derivative of each term in (2.11) separately. The treatment is somewhat similar to the calculation of directional derivatives of the potential function in [KYMP12]. Each of the first three terms is of the form

$$\begin{aligned}\frac{d}{d\mathrm{h}} H(f^{\circledast}(\mathrm{x}_\mathrm{h})) &= \lim_{\Delta\mathrm{h}\to 0} \frac{H(f^{\circledast}(\mathrm{x}_{\mathrm{h}+\Delta\mathrm{h}})) - H(f^{\circledast}(\mathrm{x}_\mathrm{h}))}{\Delta\mathrm{h}} \\ &= \lim_{\Delta\mathrm{h}\to 0} \frac{H(f^{\circledast}(\mathrm{x}+\Delta\mathrm{x})) - H(f^{\circledast}(\mathrm{x}))}{\Delta\mathrm{h}},\end{aligned}$$

where $f(u) = \sum_k f_k u^k$ is some polynomial and $\Delta\mathrm{x}$ is a shorthand for $\mathrm{x}_{\mathrm{h}+\Delta\mathrm{h}} - \mathrm{x}_\mathrm{h}$. To keep the formulas uncluttered, in all expressions containing the limit $\Delta\mathrm{x} \to 0$ we suppress the h indices. Expanding, we obtain

$$\begin{aligned}&\frac{d}{d\mathrm{h}} H(f^{\circledast}(\mathrm{x}_\mathrm{h})) = \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\sum_k f_k \sum_{j\geq 1} \binom{k}{j} \Delta\mathrm{x}^{\circledast j} \circledast \mathrm{x}^{k-j}\right)}{\Delta\mathrm{h}} \\ &= \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\sum_k k f_k \Delta\mathrm{x} \circledast \mathrm{x}^{k-1}\right)}{\Delta\mathrm{h}} + \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\Delta\mathrm{x}^{\circledast 2} \circledast g(\mathrm{x},\Delta\mathrm{x})\right)}{\Delta\mathrm{h}} \\ &= \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\Delta\mathrm{x} \circledast f'^{\circledast}(\mathrm{x})\right)}{\Delta\mathrm{h}},\end{aligned}$$

where in the last step all the higher order terms (i.e. those containing a $\circledast$-power of $\Delta\mathrm{x}$ higher than 1 disappear. The polynomial g was introduced just to collect those terms, and the fact that they vanish is shown below in Lemma 57. Explicitly, the derivatives of the first three terms are:

$$\begin{aligned}\frac{d}{d\mathrm{h}}\left[-\frac{\Lambda'(1)}{P'(1)} H(P^{\circledast}(\mathrm{x}_\mathrm{h}))\right] &= -\Lambda'(1) \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\Delta\mathrm{x} \circledast \rho^{\circledast}(\mathrm{x})\right)}{\Delta\mathrm{h}}, \\ \frac{d}{d\mathrm{h}}\left[-\Lambda'(1) H(\rho^{\circledast}(\mathrm{x}_\mathrm{h}))\right] &= -\Lambda'(1) \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\Delta\mathrm{x} \circledast \rho'^{\circledast}(\mathrm{x})\right)}{\Delta\mathrm{h}}, \\ \frac{d}{d\mathrm{h}}\left[\Lambda'(1) H(\mathrm{x}_\mathrm{h} \circledast \rho^{\circledast}(\mathrm{x}_\mathrm{h}))\right] &= \\ &= \Lambda'(1) \lim_{\Delta\mathrm{h}\to 0} \frac{H\left(\Delta\mathrm{x} \circledast \rho^{\circledast}(\mathrm{x})\right) + H\left(\Delta\mathrm{x} \circledast \mathrm{x} \circledast \rho'^{\circledast}(\mathrm{x})\right)}{\Delta\mathrm{h}}.\end{aligned}$$

Using Lemma 55, we replace $H\left(\mathrm{x} \circledast \rho'^{\circledast}(\mathrm{x}) \circledast \Delta\mathrm{x}\right)$ with $H\left(\rho'^{\circledast}(\mathrm{x}) \circledast \Delta\mathrm{x}\right) - H\left(\mathrm{x} \circledast (\rho'^{\circledast}(\mathrm{x}) \circledast \Delta\mathrm{x})\right)$, and we are thus able to cancel the contributions of the first two terms.

The derivative of the last of the four terms in (2.11) needs to be handled more carefully, since it contains both kinds of operations on densities. However, the idea remains the same: we examine the quantity

$$H((\mathsf{c}+\Delta\mathsf{c}) \circledast \Lambda^{\circledast}(\rho^{\boxtimes}(\mathsf{x}+\Delta\mathsf{x})) - \mathsf{c} \circledast \Lambda^{\circledast}(\rho^{\boxtimes}(\mathsf{x})))$$

andi we classify the terms that appear according to the position of $\Delta\mathsf{c}$ and $\Delta\mathsf{x}$. There are two terms that contain once either $\Delta\mathsf{c}$ and $\Delta\mathsf{x}$:

- $\Delta\mathsf{c} \circledast \Lambda^{\circledast}(\rho^{\boxtimes}(\mathsf{x}))$,
- $\mathsf{c} \circledast \Lambda'^{\circledast}(\rho^{\boxtimes}(\mathsf{x})) \circledast (\rho'^{\boxtimes}(\mathsf{x}) \boxtimes \Delta\mathsf{x})$.

The higher order terms (the ones that contain at least two of $\Delta\mathsf{x}$ and $\Delta\mathsf{c}$) are of the types

- $(\Delta\mathsf{x} \boxtimes \Delta\mathsf{x} \boxtimes g_1(\mathsf{x},\Delta\mathsf{x})) \circledast g_2(\mathsf{x},\Delta\mathsf{x},\mathsf{c})$,
- $(\Delta\mathsf{x} \boxtimes g_1(\mathsf{x})) \circledast (\Delta\mathsf{x} \boxtimes g_2(\mathsf{x})) \circledast g_2(\mathsf{x},\Delta\mathsf{x},\mathsf{c})$,
- $(\Delta\mathsf{x} \boxtimes g_1(\mathsf{x},\Delta\mathsf{x})) \circledast g_2(\mathsf{x},\Delta\mathsf{x}) \circledast \Delta\mathsf{c}$,

where the functions g_1, g_2, g_3 are products involving $\circledast$ and $\boxtimes$ of their parameters. All the terms above have vanishing contributions in the limit, by Lemma 57.

We are now able to collect all the terms that remain and assemble them in the form

$$\begin{aligned}\frac{d}{d\mathrm{h}}U(\mathsf{x}_\mathrm{h},\mathsf{c}_\mathrm{h}) &= \lim_{\Delta\mathrm{h}\to 0}\frac{H((\mathsf{x}-\mathsf{c}\circledast\Lambda'^{\circledast}(\rho^{\boxtimes}(\mathsf{x})))\circledast(\rho'^{\boxtimes}(\mathsf{x})\boxtimes\Delta\mathsf{x}))}{\Delta\mathrm{h}} + \lim_{\Delta\mathrm{h}\to 0}\frac{H(\Delta\mathsf{c}\circledast\Lambda^{\circledast}(\rho^{\boxtimes}(\mathsf{x})))}{\Delta\mathrm{h}}\\ &= 0 + g^{\mathrm{BP}}(\mathrm{h}),\end{aligned}$$

where in the last step we used the fact that x is the fixed point of the DE equation, and also the alternative definition of the BP GEXIT curve provided by (C.4).

The proof is now complete, and we are left to show that the higher order terms do not contribute in the limit. We begin with some definitions and some new notations. Degradation induces a partial ordering on $\mathscr{X}$, which we denote by $z \prec z'$, where z' is degraded with respect to z. Note that density evolution preserves degradation, and the following proposition follows from standard arguments in [RU08].

Proposition 52. *If* $\mathsf{c},\mathsf{c}' \in \mathscr{X}$ *and* $\mathsf{c} \prec \mathsf{c}'$ *then* $\mathsf{x}_\mathsf{c} \prec \mathsf{x}_{\mathsf{c}'}$.

For any $\mathsf{z} \in \mathscr{X}$, the Bhattacharyya functional [RU08] is given by

$$\mathscr{B}(\mathsf{z}) = \int \mathsf{z}(h)e^{-h}d\mathsf{z}(h). \tag{C.5}$$

There is a metric defined on $\mathscr{X}$, the *Wasserstein distance (on the $|D|$ domain)* [KRU12], that has the following useful properties which we state here without proof. For any $z, z'y \in \mathscr{X}$,

$$d(z \circledast y, z' \circledast y) \leq 2d(z, z'),$$
$$d(z \boxtimes y, z' \boxtimes y) \leq d(z, z').$$

Let $\mathscr{F}$ be the set of functions $f : \mathscr{X} \to \mathscr{X}$ of the form

$$f(z) = y_1 *_1 (y_2 *_2 (\dots (y_k *_k z)))$$

for some $y_1, \dots, y_k \in \mathscr{X}$ and $*_1, \dots, *_k \in \{\circledast, \boxtimes\}$. We can easily extend f by linearity, in order to define quantities like $f(z - z')$. Then for each $f \in \mathscr{F}$ there is a constant M such that for all $z \prec z'$ we have that

$$d(f(z), f(z')) \leq Md(z, z'). \tag{C.6}$$

If $z \prec z'$, the Wasserstein distance is bounded above and below by powers of the Bhattacharyya functional, in the sense that

$$\frac{1}{4}(\mathscr{B}(z') - \mathscr{B}(z))^2 \leq d(z, z') \leq 2\sqrt{\mathscr{B}(z') - \mathscr{B}(z)}.$$

The following lemma (part of Lemma 21 in [KRU12]) will enable us to factorize the entropy of a $\circledast$-product. The reason why we consider the Bhattacharyya functional is contained in the following lemmas.

Lemma 53. *Let $z, z', y, y' \in \mathscr{X}$ such that $z \succ z'$. Then*

$$|H((z - z') \circledast (y - y')| \leq \frac{8}{\log 2} \mathscr{B}(z - z') \sqrt{2d(y, y')}.$$

We are now ready to tackle the higher order contributions. Let $M_1, M_2, \dots$ denote constants independent of the channel.

Proposition 54. *With the notation from the beginning of this section, for any $f \in \mathscr{F}$ (extended by linearity), we have*

$$\lim_{\Delta h \to 0} \frac{H(\Delta x \circledast f(\Delta x))}{\Delta h} = 0,$$
$$\lim_{\Delta h \to 0} \frac{H(\Delta c \circledast f(\Delta x))}{\Delta h} = 0.$$

Proof. We concentrate on the first limit, as the second is similar but easier. Applying Lemma

53 we obtain the upper bound

$$\lim_{\Delta h\to 0} M_1 \frac{\mathscr{B}(\Delta x)\sqrt{2d(f(x),f(x+\Delta x))}}{H(\Delta c)}.$$

Since the parametrization is just a linear interpolation between c^* and Δ_0 and $H(\cdot)$ and $\mathscr{B}(\cdot)$ are linear functionals, we have that $H(\Delta c) = M_2\mathscr{B}(\Delta c)$. Then we can replace the denominator by the Bhattacharyya quantity and use the regularity condition (2.13). The only thing left to be shown is that $\sqrt{2d(f(x),f(x+\Delta x))} \to 0$. This follows from inequality (C.6) and the fact that d is a metric. □

The main tool to turn $\boxtimes$ into $\circledast$ and vice-versa is the following.

Lemma 55 (Duality lemma, [RU08])**.** *Let* $z, z', y, y' \in \mathscr{X}$*. Then*

$$H(z \circledast y) + H(z \boxtimes y) = H(z) + H(y).$$

For differences of densities, because of linearity of H*, this takes the forms*

$$H((z-z') \circledast y) + H((z-z') \boxtimes y) = H(z-z'), \tag{C.7}$$

$$H((z-z') \circledast (y-y')) + H((z-z') \boxtimes (y-y')) = 0. \tag{C.8}$$

Proposition 54 with the identity map as f and (C.8) implies

$$\lim_{\Delta h\to 0} \frac{|H(\Delta x \boxtimes \Delta x)|}{\Delta h} = 0. \tag{C.9}$$

Proposition 56 (Proposition 6 in [KYMP12])**.** *If* z *is any symmetric measure (not necessarily signed), then*

$$H(z) = z(\overline{\mathbb{R}}) - \sum_{k=1}^{\infty} \frac{(\log 2)^{-1}}{2k(2k-1)} M_k(z),$$

where $M_k(z) = \int (\tanh h)^{2k}\, dz(h)$ *and* $z(\overline{\mathbb{R}})$ *is the total mass of te* z*.*

Moreover, for any symmetric measures z_1 *and* z_2*,*

$$M_k(z_1 \boxtimes z_2) = M_k(z_1) M_k(z_2).$$

Since the quantities $M_k(\Delta x \boxtimes \Delta x) = M_k(\Delta x)^2$ are all positive, the previous proposition implies that

$$|H(\Delta x \boxtimes \Delta x \boxtimes y)| \le |H(\Delta x \boxtimes \Delta x)|, \tag{C.10}$$

for all $y \in \mathscr{X}$. By an application of (C.7), one also obtains

$$|H((\Delta x \boxtimes \Delta x \boxtimes y_1) \circledast y_2)| \leq 2|H(\Delta x \circledast \Delta x)|. \tag{C.11}$$

We are finally ready to state the result proving the vanishing contribution of higher order terms:

Lemma 57. *We have*

$$\lim_{\Delta h \to 0} \frac{H((\Delta x \boxtimes \Delta x \boxtimes g_1(x, \Delta x)) \circledast g_2(x, \Delta x, c, \Delta c))}{\Delta h} = 0, \tag{C.12}$$

$$\lim_{\Delta h \to 0} \frac{H((\Delta x \boxtimes g_1(x)) \circledast (\Delta x \boxtimes g_2(x)) \circledast g_3(x, \Delta x, c))}{\Delta h} = 0, \tag{C.13}$$

$$\lim_{\Delta h \to 0} \frac{H(\Delta c \circledast (\Delta x \boxtimes g_2(x)) \circledast g_3(x, \Delta x, c))}{\Delta h} = 0. \tag{C.14}$$

Proof. The limit (C.12) is a direct consequence of (C.11). The third one, (C.14), is a consequence of Proposition 54. The second one can also be reduced to the form appearing in Proposition 54 by using the Duality Lemma twice:

$$\begin{aligned} & H((\Delta x \boxtimes g_1(x)) \circledast (\Delta x \boxtimes g_2(x)) \circledast g_3(x, \Delta x, c)) \\ = \; & H(\Delta x \boxtimes g_1(x) \boxtimes ((\Delta x \boxtimes g_2(x)) \circledast g_3(x, \Delta x, c))) \\ = \; & H(\Delta x \circledast (g_1(x) \boxtimes ((\Delta x \boxtimes g_2(x)) \circledast g_3(x, \Delta x, c)))). \end{aligned}$$

□

D Auxilliary lemmas and calculations for freezing threshold in graph coloring

D.1 Relating the planted model to the Galton-Watson process

We call an assignment *balanced* if the number of vertices of each color under the assignment is $\frac{N}{Q} - o(N^{-2/3})$. This definition was chosen in such a way that an assignment chosen uniformly at random will w.h.p. be balanced.

Lemma 58. *The total variation distance between the distribution of the graph neighborhood* $\mathcal{N}_v(G(N,\alpha;\underline{\sigma});t)$ *and the distribution of* $\mathcal{T}(\alpha;t)$ *is* $1-o(1)$ *as* $N\to\infty$.

Proof. Let E_1 be the event that $\underline{\sigma}$ is not balanced and let E_2 be the event that $\underline{\sigma}$ is balanced but $\mathcal{N}_v(G(N,\alpha;\sigma);t)$ is not a tree. Their probabilities are both $o(1)$: for the former it is a consequence of the Central Limit Theorem; for the latter it will become clear towards the end of the proof. In what follows we condition on E_1 not happening.

We now play a card game: place one card on each potential edge of the graph, face down. The card indicates whether that is an actual edge or not, and is drawn according to the planted model. This means all cards are independent. We first pick a node at random, call it u: this will be the root, and we mark it red, and we turn all cards neighboring u. We mark the neighbors of u green.[1] As long as there are green nodes at distance $\le t$, pick one of them, mark it red and do the same thing as we did with the root, with one exception: never reveal cards that are between this node and another green node. These cards are to remain face down until the end of the game: we color the back of these cards red. Not revealing the red cards means that we always reveal a tree, even if the full neighborhood is not a tree. We forget now the labels of the vertices, but we keep the ordering of children (this helps with computing probabilities); we also annotate the nodes with their colors under $\underline{\sigma}$.

Suppose the revealed tree has size m. Then there will be less than m^2 red cards. Each card in general indicates "edge" with probability $p=\frac{\alpha Q}{(Q-1)N}$, so given the revealed tree, the probability that there is in fact a cycle in the neighborhood is $O(\frac{m^2}{N})$.

[1] These colors are just for explaining the proof; they have nothing to do with the planted coloring $\underline{\sigma}$!

We now compare the probability of the revealed tree in the two distributions, in two steps.

(A) We first compute the probability that a revealed node colored q has $d \leq N^{1/10}$ children colored $q' \neq q$. In the card game, this is given by the binomial distribution:

$$\binom{\frac{N}{Q} - o(N^{2/3})}{d} \left(\frac{\alpha Q}{(Q-1)N}\right)^d \left(1 - \frac{\alpha Q}{(Q-1)N}\right)^{\frac{N}{Q} - o(N^{2/3}) - d)}.$$

In the Galton-Watson tree, the same quantity is given by the Poisson distribution $e^{-\frac{\alpha}{Q-1}} \frac{1}{d!} \left(\frac{\alpha}{Q-1}\right)^d$. After simplifications it is apparent that the ratio between the two is $1 + o(N^{-1/5})$.

(B) The probability of the whole tree is a product of $m(Q-1)$ quantities of the type computed in step (A). Assuming $m = o(N^{1/10})$ (we will see this is a reasonable choice), the probability of the tree showing up only differs by a ratio of $1 - o(N^{1/20})$.

The expected size and of the Galton-Watson tree of depth t is $O(1)$, since it does not depend on N at all. The variance is also $O(1)$, so clearly the event E_3 of trees with $m > N^{1/10}$ occuring has probability $o(1)$ probability by Chebyshev's inequality.

Then by the union bound $\Pr[E_2] \leq \Pr[E_3] + O(\frac{m^2}{N}) = o(1)$. The total variation bound is obtained by summing the differences between the two distributions, outside E_1, E_2 and E_3. □

This lemma can be extended to the coupled scenario quite easily.

Lemma 59. *For fixed W and L, the total variation distance between the distribution of the graph neighborhood $\mathcal{N}_v^{coup}(G(N,\alpha,L,W;\underline{\sigma});t)$ and the distribution of $\mathcal{T}^{coup}(\alpha,L,W;t)$ is $1-o(1)$ as $N \to \infty$.*

Proof. The only difference to the uncoupled scenario is the existence of the additional label representing position. When comparing the probability of each tree appearing under the two distributions, the quantities computed in step (A) will be the probabilities that a node situated at position z and colored q has d neighbors at position $z+w$ with $w \in \{-W+1,\dots,W-1\}$, and colored $q' \neq q$. At boundaries there will be fewer terms but none of this impacts the final result. □

D.2 Asymptotic behaviour of the freezing threshold for the coupled model

We derive here the first terms in the asymptotic expansion of the freezing threshold as $Q \to \infty$. It is more convenient to work with $a = \frac{\alpha}{(Q-1)\log(Q-1)}$, as the following lemmas will make clear

that this is the right scaling. With this convention, we rewrite the potential and its derivative as

$$U(x) = \frac{x^2}{2} - x + \frac{1}{a\log(Q-1)} \sum_{q=1}^{Q-1} \frac{1}{q} \left(1 - \frac{1}{(Q-1)^{ax}}\right)^q. \tag{D.1}$$

$$U'(x) = x - \left(1 - \frac{1}{(Q-1)^{ax}}\right)^{Q-1}. \tag{D.2}$$

Lemma 60. *For fixed $Q, a > 0, x > 0$, if $ax < 1$ then $U'(x) > 0$. By integration we also obtain that $U(x) > 0$. The claim is also valid for $ax = 1$ when $a < e$.*

Proof. We distinguish two cases.

When $x \leq \frac{1}{a\log^2(Q-1)}$, we use $1 - e^{-y} \geq y$ to obtain $U'(x) \geq x - (\log(Q-1)ax)^{Q-1} > 0$.

When $x \geq \frac{1}{a\log^2(Q-1)}$, we use $1 - y \geq e^{-y}$ to obtain $U'(x) \geq x - e^{-(Q-1)^{1-ax}} \geq 0$. □

Lemma 61. *If $a > 0, x \in (0,1]$ are fixed such that $1 < ax < (1 - x/2)^{-1}$, then for all sufficiently large Q, $U(x) > 0$. Moreover, for $a > 2$ and $x = 1$, we have $U(x) < 0$ for all sufficiently large Q.*

Proof. We first observe that $\sum_{q=1}^{Q-1} \frac{1}{q} \left(1 - \frac{1}{(Q-1)^{ax}}\right)^q = \sum_{q=1}^{Q-1} \frac{1}{q} - O\left((Q-1)^{1-ax}\right)$.

Using $\sum_{q=1}^{Q-1} \frac{1}{q} = \log(Q-1) + O(1)$, it we easily check that for $a < 2$ and for large enough Q

$$U(x) = \frac{x^2}{2} - x + \frac{1}{a} - o(1) > 0.$$

For $a > 3$, we see immediately that the above expression attains negative values. □

From this it becomes apparent that one should search for the coupled freezing threshold in the vicinity of $a = 2$; we also infer that we should find the minimum of $U(x)$ at values of x close to 1. There are two possibilities: (i) either the minimum is achieved in the interior at some x_*, where $U'(x_*) = 0$ is fulfilled or (ii) the minimum is achieved at $x = 1$. In any case, for the first few terms in the expansion we will see that $U(1)$ will be identical to $U(x_*)$.

We write $x_* = 1 - \kappa$ and $a = 2 + \delta$, with $\delta, \kappa = o(1)$. The condition $U'(x_*) = 0$ gives $\kappa = (Q-1)^{-1+2\kappa-\delta+\delta\kappa} + O((Q-1)^{-2+o(1)})$.

The following expansion will be useful. For two functions f and ϵ, if $\epsilon = (C + o(1))\log^{-1}(f)$, then

$$\frac{1}{f^{1+\epsilon}} = e^{-\log f - C + o(1)} = \frac{e^{-C}}{f} + o(\frac{1}{f}). \tag{D.3}$$

We will self-consistently check that $\delta = O(\frac{1}{\log(Q-1)})$. Using the remark above, we can deduce

$\kappa = O(\frac{1}{Q-1})$.

We now turn to the condition $U(x_*) = 0$. For this expansions

$$\sum_{q=1}^{Q-1} \frac{1}{q}(1-\frac{1}{(Q-1)^{ax}})^q = \sum_{q=1}^{Q-1} \frac{1}{q} - \frac{1}{(Q-1)^{1-2\kappa+\delta-\kappa\delta}} + O\left(\frac{1}{(Q-1)^{2-o(1)}}\right),$$
$$\sum_{q=1}^{Q-1} \frac{1}{q} = \log(Q-1) + \gamma + \frac{1}{2(Q-1)} + O(\frac{1}{(Q-1)^{2-o(1)}}),$$

where γ is the Euler-Mascheroni constant. After some rearrangement we obtain

$$\left(\frac{1}{2}+\frac{\kappa^2}{2}\right)(2+\delta) = 1 + \frac{\gamma}{\log(Q-1)} + \frac{1}{2(Q-1)\log(Q-1)} - \frac{1}{(Q-1)^{1+\delta}\log(Q-1)} + O(\frac{1}{(Q-1)^{2-o(1)}}).$$

Note that if we were to look at the condition $U(1) = 0$ instead, the differece would be only the term $\frac{\kappa^2}{2}$, which is so small that we will anyway ignore. After another application of (D.3) and the change $\alpha = (2+\delta)(Q-1)\log(Q-1)$, we finally obtain that for the coupled system

$$\alpha_f^{c(m=1)} = 2(Q-1)\log(Q-1) + 2\gamma(Q-1) + 1 - 2e^{-2\gamma} + o(1), \tag{D.4}$$

or in terms of Q,

$$\alpha_f^{c(m=1)} = 2Q\log Q + 2\gamma Q - 2\log Q - 1 - 2\gamma - 2e^{-2\gamma} + o(1). \tag{D.5}$$

E Proofs and observations for the SP threshold saturation

E.1 Properties of the space of densities

We make now a number of observations in order to understand better the densities $\mathbf{p}$. We first introduce a suitable metric and then an ordering between densities. We will view the equation (3.35) as the fixed point equation of an operator on densities, and the ordering on densities will be chosen in such a way that this operator be monotone. Since these properties may turn out to be useful later, we phrase them in a more general language.

E.1.1 The metric space

Let $\mathfrak{M}$ be the space of probability measures on a closed interval I of the real line, hereafter referred as *densities*. We begin with the observation that elements of M can be put in bijective correspondence with the (i) continuous-to-the-right-limit-to-the-left (càdlàg) functions that are nondecreasing and 0 on $(-\infty, \inf I)$ and 1 on $[\sup I, +\infty)$ or (ii) elements of $L^1(I')$, nondecreasing a.e., 0 to the left of I and 1 to the right, where I' is some neighborhood of I. We refer to the object given by (i) and (ii) as the cdf of a probability measure, even though the cdf is given strictly speaking by (i), while (ii) is the equivalence class of (i) in the sense of equality almost everywhere. The cdf will typically be written as $\mathbf{p}((-\infty, \cdot))$.

The metric $d_C : \mathfrak{M} \times \mathfrak{M} \to \mathbb{R}$ will be given by the distance in L^1 norm between the cdfs of the measures, in other words by

$$d_C(\mathbf{p}, \mathbf{p}') = \int \left| \mathbf{p}((-\infty, x]) - \mathbf{p}'((-\infty, x]) \right| \mathrm{d}x. \tag{E.1}$$

The metric space thus defined is complete, because $L^1(I')$ is complete and the subset of valid cdfs of $\mathfrak{M}$ in $L^1(I')$ is closed. The latter is true because the limit of a converging sequence of cdfs (given in the case of cdfs a.e. by pointwise limit) is clearly nondecreasing and takes values 0 and 1 to the left and to the right of I, respectively. Note that it is important in this definition

that I be compact.

The metric space is totally bounded. To see this, assume for simplicity that $I = [0,1]$. Fix $\epsilon > 0$ and set $n = \lceil \frac{1}{\epsilon} \rceil$. Let $\mathfrak{M}_\epsilon$ be defined as the set of densities with support on the set whose cdfs are nondecreasing step functions that jump only at positions in $B_n = \{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$ and take the discrete values also in B_n. Then any density $\mathbf{p} \in \mathfrak{M}$ is at most 2ϵ away from an element from the finite set $\mathfrak{M}_\epsilon$. Since $\mathfrak{M}$ as a metric space is complete and totally bounded, it is compact.

E.1.2 Partial ordering

The partial ordering we set on $\mathfrak{M}$ is an ordering of cdfs in the sense that $\mathbf{p} \preceq \mathbf{p}'$ if the cdf of $\mathbf{p}$ lies above that of $\mathbf{p}'$, i.e. $\mathbf{p}((-\infty, x]) \geq \mathbf{p}'((-\infty, x])$. The intuition regarding the direction of the inequalities is recovered when one thinks that for the "lesser" density the probability mass lies to the left of the mass of the "greater" density. The ordering can be expressed equally by $\mathbf{p}((x, +\infty)) \leq \mathbf{p}'((x, +\infty))$. Mirroring the terminology used for channel coding, we will say that $\mathbf{p}'$ is *degraded* with respect to $\mathbf{p}$.

Note that in the case where two densities are ordered, the absolute value appearing in (E.1) comes out of the integral. As such, in a monotone sequence of densities, the distances are additive. This implies that a monotone sequence of densities always converges.

We introduce the operations $\vee$ and $\wedge$ by defining

$$(\mathbf{p} \vee \mathbf{p}')((x, +\infty)) = \max\{\mathbf{p}((x, +\infty)), \mathbf{p}'((x, +\infty))\}, (\mathbf{p} \wedge \mathbf{p}')((x, +\infty)) = \min\{\mathbf{p}((x, +\infty)), \mathbf{p}'((x, +\infty))\},$$

for arbitrary $\mathbf{p}, \mathbf{p}' \in \mathfrak{M}$. It can be easily checked that

$$d_C(\mathbf{p} \wedge \mathbf{p}', \mathbf{p}') \leq d_C(\mathbf{p}, \mathbf{p}'). \qquad d_C(\mathbf{p} \vee \mathbf{p}', \mathbf{p}') \leq d_C(\mathbf{p}, \mathbf{p}'), \tag{E.2}$$

(E.3)

E.1.3 Linear combinations of densities

We will make use of addition and multiplication by scalar of elements of $\mathfrak{M}$. These are to be understood as operations on signed measures, and they are equivalent to the same operations performed on cdfs in $L^1(I')$ (regardless of whether the results are cdf's or not). It is also clear that convex combinations of elements of $\mathfrak{M}$ are, however, in $\mathfrak{M}$. Infinite convex combinations will also be used, of the kind $\mathbb{E}_d \mathbf{p}^{(d)}$, where $\{\mathbf{p}^{(d)}\}$ is an infinite sequence of densities. These should be interpreted as the densities corresponding to the limit (in $L^1(I')$) as $n \to \infty$ of the cdfs of the partial sums $\sum_{d=0}^{n} \frac{e^{-\alpha}\alpha^d}{d!} \mathbf{p}^{(d)}$. The limit always exists in the case of Poisson distributions, as it can be verified immediately that the sequence of partial sums is Cauchy.

Convex combinations of densities that are pairwise ordered are themselves ordered. In other words, if $\{t_i\}$ is a countable sequence of nonnegative reals such that $\sum_i t_i = 1$, and if $\mathbf{p}_1^{(i)} \preceq \mathbf{p}_2^{(i)}$

for all i, then $\sum_i t_i \mathbf{p}_1^{(i)} \preceq \sum_i t_i \mathbf{p}_2^{(i)}$ when the sums converge.

E.2 Proofs of properties of functions f, g and ϕ

E.2.1 Proof of Lemma 23

Proof. We write the partial derivative with respect x_1 (the function is symmetric under reordering of parameters). It is

$$\begin{aligned}\frac{\partial g_d}{\partial x_1}(x_1,\dots,x_d) &= \frac{\partial}{\partial x_1}\sum_{l=0}^{Q-1}(-1)^l\binom{Q}{l+1}\prod_{i=1}^{d}(1-(l+1)x_i)\\ &= -\sum_{l=0}^{Q-1}(-1)^l\binom{Q}{l+1}(l+1)\prod_{i=2}^{d}(1-(l+1)x_i)\\ &= -Q\sum_{l=0}^{Q-1}(-1)^l\binom{Q-1}{l}\prod_{i=2}^{d}(1-(l+1)x_i)\\ &= -Qf_{d-1}(x_2,\dots,x_d). \end{aligned} \tag{E.4}$$

The first claim now follows, since $f_{d-1}(x_2,\dots,x_d)$ has a probabilistic interpretation and is thus nonnegative.

For the last claim, we have $\lim_{d\to\infty}(1-\frac{1}{Q})^{-d}g_d(\frac{1}{Q},\dots,\frac{1}{Q}) = Q$. Thus there is $c_0 > 0$ and d_0 such that for all $d \geq d_0$ it holds that $g_d(\frac{1}{Q},\dots,\frac{1}{Q}) \geq c_0(1-\frac{1}{Q})^d$. We will set $K = \min\{c_0, \min_{d=1,\dots,d_0}(1-\frac{1}{Q})^{-d}g_d(\frac{1}{Q},\dots,\frac{1}{Q})\}$ and the claim follows. □

E.2.2 Proof of Lemma 24

Proof. Because of symmetry, it will be sufficient to prove that $\phi(\underline{x})$ is increasing in x_1. To simplify notation, let us set $T(l+1) = \prod_{i=2}^{d}(1-(l+1)x_i)$. The fact that $\phi(\underline{x})$ is increasing in x_1 is equivalent to the positivity of the quantity

$$\begin{aligned}&\frac{\partial f(\underline{x})}{\partial x_1}g(\underline{x}) - f(\underline{x})\frac{\partial g(\underline{x})}{\partial x_1} =\\ &\quad= \sum_{l=0}^{Q-1}(-1)^{l+1}\binom{Q-1}{l}(l+1)T(l+1)\sum_{l'=0}^{Q-1}(-1)^{l'}\binom{Q}{l'+1}(1-(l'+1)x_1)T(l'+1)\\ &\quad\quad-\sum_{l=0}^{Q-1}(-1)^{l}\binom{Q-1}{l}(1-(l+1)x_1)T(l+1)\sum_{l'=0}^{Q-1}(-1)^{l'+1}\binom{Q}{l'+1}(l'+1)T(l'+1)\\ &\quad= \sum_{l,l'=0}^{Q-1}(-1)^{l+l'+1}\binom{Q-1}{l}\binom{Q}{l'+1}[(l+1)-(l'+1)]T(l+1)T(l'+1)\end{aligned}$$

$$= \sum_{l,l'=0}^{Q-1} \frac{(-1)^{l+l'+1}}{Q} \binom{Q}{l+1}\binom{Q}{l'+1} [(l+1)^2 - (l'+1)(l+1)]\, T(l+1)\, T(l'+1)$$
$$= \sum_{0 \le l < l' < Q} \frac{(-1)^{l+l'+1}}{Q} \binom{Q}{l+1}\binom{Q}{l'+1} (l-l')^2\, T(l+1)\, T(l'+1).$$

For $Q = 3$, the above condition reduces to

$$3T(1)T(2) - 4T(1)T(3) + T(2)T(3) > 0.$$

Since if $T(3) = 0$ the condition holds trivially, we will assume that $T(1), T(2), T(3)$ are all positive and it will be enough to show that $\frac{3}{4}\frac{1}{T(3)} + \frac{1}{4}\frac{1}{T(1)} > \frac{1}{T(2)}$. This we prove using the arithmetic-geometric mean inequality twice, as follows:

$$\frac{3}{4}\frac{1}{T(3)} + \frac{1}{4}\frac{1}{T(1)} \ge \frac{1}{T(3)^{3/4} T(1)^{1/4}} = \prod_{i=2}^{d} \frac{1}{(1-3x_i)^{3/4}(1-x_i)^{1/4}} \ge$$
$$\ge \prod_{i=2}^{d} \frac{1}{\frac{3}{4}(1-3x_i) + \frac{1}{4}(1-x_i)} \ge \prod_{i=2}^{d} \frac{1}{1-2x_i} = \frac{1}{T(2)}.$$

□

E.2.3 Proof of Lemma 25

Proof. We begin with the bounds on ϕ. The positivity is a direct consequence of the monotonicity of ϕ. We derive now the upper bound. To make notation more compact, let $\underline{x}$ be the vector $(x_1, \dots, x_d)$ and $\overline{x}$ the same without x_1, i.e. $(x_2, \dots, x_d)$.

We have

$$\frac{\partial}{\partial x_1}\phi(\underline{x}) = \frac{\partial}{\partial x_1}\frac{f(\underline{x})}{g(\underline{x})} = \frac{1}{g(\underline{x})^2}\left(\frac{\partial f(\underline{x})}{\partial x_1} g(\underline{x}) - f(\underline{x})\frac{\partial g(\underline{x})}{\partial x_1}\right). \tag{E.5}$$

The quantities $f(\overline{x})$, $g(\overline{x})$, $\frac{\partial f(\underline{x})}{\partial x_1}$ and $\frac{\partial g(\underline{x})}{\partial x_1}$ only depend on $\overline{x}$, and all have a probabilistic interpretation. It is in fact the same probabilistic interpretation that allowed us to write f and g in the first place. We have $d-1$ balls indexed from 2 to d that are placed into $Q+1$ bins, labeled $\{1, \dots, Q, *\}$. Ball i will be placed in each of the numbered bins with probability x_i and with probability $1 - Qx_i$ it will go into the $*$ bin. Let B be the random variable that counts the number of empty bins among $\{1, \dots, Q\}$.

For $f(\overline{x})$ and $g(\overline{x})$ we already know the probabilistic interpretation. For the other two it arises using the inclusion-exclusion principle:

$$f(\overline{x}) = \frac{1}{Q}\mathbb{P}[B = 1], \qquad g(\overline{x}) = \mathbb{P}[B \ge 1].$$

$$\begin{aligned}
\frac{\partial f(\underline{x})}{\partial x_1} &= \sum_{l=0}^{Q-1}(-1)^{l+1}\binom{Q-1}{l}(l+1)\prod_{i=2}^{d}(1-(l+1)x_i) \\
&= \sum_{l=0}^{Q-1}(-1)^{l+1}\binom{Q-1}{l}l\prod_{i=2}^{d}(1-(l+1)x_i) + \sum_{l=0}^{Q-1}(-1)^{l+1}\binom{Q-1}{l}\prod_{i=2}^{d}(1-(l+1)x_i) \\
&= \sum_{l'=0}^{Q-2}(-1)^{l'}(Q-1)\binom{Q-2}{l'}\prod_{i=2}^{d}(1-(l'+2)x_i) - \sum_{l=0}^{Q-1}(-1)^{l}\binom{Q-1}{l}\prod_{i=2}^{d}(1-(l+1)x_i) \\
&= \frac{2}{Q}\mathbb{P}[B=2] - \frac{1}{Q}\mathbb{P}[B=1].
\end{aligned}$$

$$\begin{aligned}
\frac{\partial g(\underline{x})}{\partial x_1} &= \sum_{l=0}^{Q-1}(-1)^{l+1}\binom{Q}{l+1}(l+1)\prod_{i=2}^{d}(1-(l+1)x_i) \\
&= -\sum_{l=0}^{Q-1}(-1)^{l}Q\binom{Q-1}{l}\prod_{i=2}^{d}(1-(l+1)x_i) \\
&= -\mathbb{P}[B=1].
\end{aligned}$$

We make the dependence on x_1 explicit in $f(\underline{x})$ and $g(\underline{x})$:

$$\begin{aligned}
f(\underline{x}) &= f(\overline{x}) + x_1\frac{\partial f(\underline{x})}{\partial x_1} = \frac{1}{Q}\left(\mathbb{P}[B=1] + x_1(2\mathbb{P}[B=2] - \mathbb{P}[B=1])\right), \\
g(\underline{x}) &= g(\overline{x}) + x_1\frac{\partial g(\underline{x})}{\partial x_1} = \mathbb{P}[B\geq 1] - x_1\mathbb{P}[B=1].
\end{aligned}$$

Replacing all these into (E.5) and noticing that the numerator does not in fact depend on x_1, we obtain

$$\begin{aligned}
\frac{\partial}{\partial x_1}\phi(\underline{x}) &= \frac{(2\mathbb{P}[B=2]-\mathbb{P}[B=1])\mathbb{P}[B\geq 1] + \mathbb{P}[B=1]\mathbb{P}[B=1]}{Q(\mathbb{P}[B\geq 1] - x_1\mathbb{P}[B=1])^2} \\
&\leq \frac{2\mathbb{P}[B=2]\mathbb{P}[B\geq 1] - \mathbb{P}[B=1](\mathbb{P}[B\geq 1] - \mathbb{P}[B=1])}{Q(\mathbb{P}[B\geq 1] - \frac{1}{Q}\mathbb{P}[B=1])^2} \\
&= \frac{2\mathbb{P}[B=2]\mathbb{P}[B\geq 1] - \mathbb{P}[B=1]\mathbb{P}[B\geq 2]}{Q(\frac{Q-1}{Q}\mathbb{P}[B\geq 1])^2} \\
&\leq \frac{2}{Q}\frac{\mathbb{P}[B\geq 1]}{\mathbb{P}[B\geq 2]} = \frac{2Q}{(Q-1)^2}
\end{aligned} \tag{E.6}$$

In the case of $\log g(\underline{x})$ we have

$$\frac{\partial}{\partial x_1}\log g(\underline{x}) = \frac{\frac{\partial}{\partial x_1}g(\underline{x})}{g(\underline{x})} = \frac{-\mathbb{P}[B=1]}{\mathbb{P}[B\geq 1] - x_1 P[B=1]} \geq \frac{-\mathbb{P}[B=1]}{\frac{Q-1}{Q}\mathbb{P}[B\geq 1]} \geq -\frac{Q}{Q-1}.$$

□

E.3 Properties of the operator $\mathscr{F}$

E.3.1 Proof of Lemma 28 (Monotonicity with respect to the densities)

We will first prove that the operator $\mathscr{F}$ is monotone with respect to $\preceq$, by first splitting up $\mathscr{F}$ in the terms corresponding to each degree d. Let $\mathscr{F}^{(d)} : \mathfrak{M}^d \to \mathfrak{M}$ be defined by

$$\mathscr{F}^{(d)}(\mathbf{p}_1,\dots,\mathbf{p}_d) = \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}_1(x_1)\cdots\mathrm{d}\mathbf{p}_d(x_d)\delta_{\phi(x_1,\dots,x_d)}. \tag{E.7}$$

The fixed-degree version of $\mathscr{F}$ will be used for all the proofs in this group.

Proof. We show that for fixed $\mathbf{p}_2,\dots,\mathbf{p}_d$, we have that $\mathscr{F}^{(d)}(\mathbf{p},\mathbf{p}_2,\dots,\mathbf{p}_d) \preceq \mathscr{F}^{(d)}(\mathbf{p}',\mathbf{p}_2,\dots,\mathbf{p}_d)$.

Let us first make an observation. Since $\phi(x_1,\dots,x_d)$ is increasing in x_1, there is a function $\psi : [0,1/Q]^d \to \mathbb{R}$, such that $\phi(x_1,\dots,x_d) > a$ is equivalent to $x_1 > \psi(a,x_2,\dots,x_d)$, for all $x_1,\dots,x_d,a \in [0,1/Q]$.

We can now prove the first part of the lemma by making use of the function ψ:

$$\begin{aligned}
&\mathscr{F}^{(d)}(\mathbf{p},\mathbf{p}_2,\dots,\mathbf{p}_d)((a,1/Q]) = \\
&= \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}_1(x_1)\cdots\mathrm{d}\mathbf{p}_d(x_d)\mathbb{1}(\phi(x_1,\dots,x_d) > a) \\
&= \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1)\cdots\mathrm{d}\mathbf{p}(x_d)\mathbb{1}(x_1 > \psi(a,x_2,\dots,x_d)) \\
&= \int_{[0,1/Q]^{d-1}} \mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_d)\mathbf{p}((\psi(a,x_2,\dots,x_d),1/Q]) \\
&\le \int_{[0,1/Q]^{d-1}} \mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_d)\mathbf{p}'((\psi(a,x_2,\dots,x_d),1/Q]) \\
&= \mathscr{F}^{(d)}(\mathbf{p}',\mathbf{p}_2,\dots,\mathbf{p}_d)((a,1/Q]).
\end{aligned}$$

It is then easy to see that by applying this result for all d parameters one obtains $\mathscr{F}^{(d)}(\mathbf{p},\dots,\mathbf{p}) \preceq \mathscr{F}^{(d)}(\mathbf{p}',\dots,\mathbf{p}')$. Then the second claim of the lemma follows since $\mathscr{F}(\mathbf{p}) = \mathbb{E}_d\mathscr{F}^{(d)}(\mathbf{p},\dots,\mathbf{p})$. □

E.3.2 Proof of Lemma 29 (Monotonicity with respect to α)

Proof. Let d_1 be drawn randomly from a Poisson(α) distribution, and independently let d_2 be drawn from Poisson($\alpha' - \alpha$). Then $d_1 + d_2$ is Poisson(α')-distributed.

Since $\boldsymbol{\delta_0} \preceq \mathbf{p}$, we observe that

$$\mathscr{F}^{(d+d')}(\mathbf{p},\dots,\mathbf{p}) \preceq \mathscr{F}^{(d+d')}(\underbrace{\mathbf{p},\dots,\mathbf{p}}_{d \text{ times}},\underbrace{\boldsymbol{\delta_0},\dots,\boldsymbol{\delta_0}}_{d' \text{ times}}) = \mathscr{F}^{(d)}(\mathbf{p},\dots,\mathbf{p}).$$

We then have

$$\mathscr{F}_{\alpha'}(\mathbf{p}) = \mathbb{E}_{d_1:\alpha}\mathbb{E}_{d_2:\alpha'-\alpha}\mathscr{F}^{(d_1+d_2)}(\mathbf{p},\dots,\mathbf{p}) \preceq \mathbb{E}_{d_1:\alpha}\mathscr{F}^{(d_1)}(\mathbf{p},\dots,\mathbf{p}) = \mathscr{F}_{\alpha}(\mathbf{p}).$$

□

E.3.3 Proof of Lemma 30 (Continuity)

Proof. We will prove that

$$d_C(\mathscr{F}(\mathbf{p}),\mathscr{F}(\mathbf{p}')) \le \gamma d_C(\mathbf{p},\mathbf{p}'), \tag{E.8}$$

for some constant $\gamma > 0$ depending only on Q. We begin with a weaker claim that assumes that $\mathbf{p} \succeq \mathbf{p}'$.

We first prove a version of the claim for $\mathscr{F}^{(d)}$:

$$\begin{aligned}
&d_C(\mathscr{F}^{(d)}(\mathbf{p},\mathbf{p}_2,\dots,\mathbf{p}_d),\mathscr{F}^{(d)}(\mathbf{p}',\mathbf{p}_2,\dots,\mathbf{p}_d) = \\
&= \int_0^{1/Q} \mathrm{d}x \int_{[0,1/Q]^d} \mathrm{d}(\mathbf{p}-\mathbf{p}')(x_1)\,\mathrm{d}\mathbf{p}_2(x_2)\cdots\mathrm{d}\mathbf{p}_d(x_d)\,\mathbb{1}\{\phi_d(x_1,\dots,x_d) > x\} \\
&= \int_{[0,1/Q]^d} \mathrm{d}(\mathbf{p}-\mathbf{p}')(x_1)\,\mathrm{d}\mathbf{p}_2(x_2)\cdots\mathrm{d}\mathbf{p}_d(x_d) \int_0^{1/Q} \mathrm{d}x\,\mathbb{1}\{\phi_d(x_1,\dots,x_d) > x\} \\
&= \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}_2(x_2)\cdots\mathrm{d}\mathbf{p}_d(x_d)\,\mathrm{d}(\mathbf{p}-\mathbf{p}')(x_1)\phi_d(x_1,\dots,x_d) \\
&= \int_{[0,1/Q]^{d-1}} \mathrm{d}\mathbf{p}_2(x_2)\cdots\mathrm{d}\mathbf{p}_d(x_d) \int_0^{1/Q} \mathrm{d}x_1(\mathbf{p}((x,1/Q]) - \mathbf{p}'((x,1/Q])))\frac{\partial}{\partial x_1}\phi_d(x_1,\dots,x_d).
\end{aligned}$$

Using the upper bound in Lemma 25, we obtain

$$d_C(\mathscr{F}^{(d)}(\mathbf{p},\mathbf{p}_2,\dots,\mathbf{p}_d),\mathscr{F}^{(d)}(\mathbf{p}',\mathbf{p}_2,\dots,\mathbf{p}_d) \le \frac{2Q}{(Q-1)^2} d_C(\mathbf{p},\mathbf{p}'), \tag{E.9}$$

for some fixed K, not dependent on d. Applying the above inequality for each parameter of $\mathscr{F}^{(d)}$ incurs an extra factor of d. We can then write

$$\begin{aligned}
d_C(\mathscr{F}(\mathbf{p}),\mathscr{F}(\mathbf{p}') &= \mathbb{E}_d\, d_C(\mathscr{F}^{(d)}(\mathbf{p},\dots,\mathbf{p}),\mathscr{F}^{(d)}(\mathbf{p}',\dots,\mathbf{p}')) \\
&= \mathbb{E}_d\, d\frac{2Q}{(Q-1)^2} d_C(\mathbf{p},\mathbf{p}') = \frac{2\alpha Q}{(Q-1)^2} d_C(\mathbf{p},\mathbf{p}').
\end{aligned} \tag{E.10}$$

To generalize this result for arbitrary $\mathbf{p}$ and $\mathbf{p}'$, we used what we proved so far for the pairs $\mathbf{p}$, $\mathbf{p}\wedge\mathbf{p}'$ and $\mathbf{p}'$, $\mathbf{p}\wedge\mathbf{p}'$. Using (E.2) and the triangle inequality we obtain the desired result. □

E.3.4 Proof of Lemma 31 ($\mathscr{F}^{(\infty)}(\mathbf{p})$ is a fixed point)

Proof. We need to show that $\mathscr{F}^{(\infty)}(\mathbf{p}) = \mathscr{F}(\mathscr{F}^{(\infty)}(\mathbf{p}))$. Using Lemma eff-continuity, we get

$$d_C(\mathscr{F}^{(n+1)}(\mathbf{p}), \mathscr{F}\circ\mathscr{F}^{(\infty)}(\mathbf{p}')) = d_C(\mathscr{F}\circ\mathscr{F}^{(n)}(\mathbf{p}), \mathscr{F}\circ\mathscr{F}^{(\infty)}(\mathbf{p}')) = O(d_C(\mathscr{F}^{(n)}(\mathbf{p}), \mathscr{F}^{(\infty)}(\mathbf{p}'))).$$

As $n\to\infty$, the right hand side tends to 0 by definition, and from the left hand side we get that $\mathscr{F}\circ F^{(\infty)}(\mathbf{p}')$ is the limit of the sequence $\{\mathscr{F}^{(n+1)}(\mathbf{p})\}$, which concludes the argument. □

E.4 The basin of attraction of $\boldsymbol{\delta_0}$ is an open set

We will showing that $\mathfrak{T}$ contains a neighborhood of $\boldsymbol{\delta_0}$. Let $\mathfrak{B}(\epsilon)$ be the open ball centered at $\boldsymbol{\delta_0}$ of radius ϵ. We first need two very technical lemmas that show that $\mathscr{F}$ is a contracting map around $\boldsymbol{\delta_0}$.

It is necessary that we treat separately the cases $Q\geq 4$ and $Q=3$. We treat the latter first, as it is simpler. Define $\mathbf{s}_\epsilon = (1-\epsilon)\boldsymbol{\delta}_\epsilon + \epsilon\boldsymbol{\delta}_{\mathbf{1/Q}}$.

Lemma 62. *For $Q\leq 4$ and $\epsilon>0$ sufficiently small, $\mathscr{F}(\mathbf{s}_\epsilon)\preceq\mathbf{s}_{\epsilon^{1.5}}$*

Proof. Consider the following setting. Draw a number d from Poisson(α). Then draw independently d numbers $x_1,\dots,x_d$ from $\mathbf{s}_\epsilon$. We need to check that the probability of the event E that $\phi(x_1,\dots,x_d)\geq\epsilon^{1.5}$ happens is less than $\epsilon^{1.5}$.

Let $d_0=\lfloor\epsilon^{-0.01}\rfloor$. Then for ϵ small enough, we have that $\Pr[d\geq d_0]\leq\epsilon^{0.1}$.

Let Y count the number of $1/Q$ among $x_1,\dots,x_d$. In the event $Y\in\{0,1\}$, we have $\phi(x_1,\dots,x_d)=O((d\epsilon)^2)$ (according to Lemma 65), and for $d<d_0$ this is outside the event E. The case $Y\leq 2$ occurs with probability $O((d\epsilon)^2)$ by the union bound.

Also by the union bound we get

$$\Pr[E]\leq\Pr[d\geq d_0]+\Pr[Y\leq 2|d<d_0]\leq\epsilon^{1.5}.$$

□

For $Q=3$, the approach needs to be more complex. Define

$$\mathbf{r}_\epsilon = (1-\epsilon-\epsilon^2)\boldsymbol{\delta}_{\epsilon^2}+\epsilon\boldsymbol{\delta}_\epsilon+\epsilon^2\boldsymbol{\delta}_{\mathbf{1/Q}}.$$

So instead of working with just two masses, at $1/Q$ and ϵ, we work with three masses, at $1/Q$, ϵ and ϵ^2.

Lemma 63. *For $Q=3$ and $\epsilon>0$ sufficiently small, $\mathscr{F}(\mathbf{r}_\epsilon)\preceq\mathbf{s}_{\epsilon^{1.2}}$.*

Proof. The setting is the same as in the previous proof, except now $x_1, \dots, x_d$ are drawn from $\mathbf{r}_\epsilon$; d_0 is chosen in the same manner. Let Y and Y' count the number of $1/Q$ and ϵ among $x_1, \dots, x_d$, respectively.

Let E be the event that $\phi(x_1, \dots, x_d) \geq \epsilon^{1.2}$. We need to check whether $\mathbb{P}[[]\, E] < \epsilon^{2.4}$. Also, let E' be the event that $\phi(x_1, \dots, x_d) \geq \epsilon^{2.4}$. We will check that $\mathbb{P}[[]\, E'] < \epsilon^{1.2}$.

We distinguish the following events, compute their probabilities and see what the values of ϕ are (using Lemma 65 below) in each of them:

$$\begin{aligned}
&\Pr[Y = 0, Y' = 0] = O(1), && \phi_d(\underline{x}) = O(d^2\epsilon^4),\\
&\Pr[Y = 0, Y' = 1] = O(d\epsilon), && \phi_d(\underline{x}) = O(d\epsilon^3),\\
&\Pr[Y = 1, Y' = 0] = O(d\epsilon^2), && \phi_d(\underline{x}) = O(d\epsilon^2),\\
&\Pr[Y = 0, Y' = 2] = O(d^2\epsilon^2), && \phi_d(\underline{x}) = O(\epsilon^2).
\end{aligned}$$

All the other combinations of Y and Y' result in events that are $O((d\epsilon)^3)$. We can then fit the cases $(Y, Y') \in (0,0), (0,1)$ outside E' and the cases $(Y, Y') \in (1,0), (0,2)$ outside E. We then perform a union bound like in the previous proof to account for the cases $d > d_0$. □

One can check that all $\mathbf{p} \in \mathfrak{B}(\epsilon^2)$ satisfy $\mathbf{p} \preceq \mathbf{s}_\epsilon$ or $\mathbf{p} \preceq \mathbf{r}_\epsilon$, as the case may be. Because of the monotonicity of $\mathscr{F}$ and the previous two lemmas, there will be some small $\epsilon_0 > 0$ for which for all $\mathbf{p} \in \mathfrak{B}(\epsilon_0^2)$ we have that $\mathscr{F}^\infty(\mathbf{p}) = \boldsymbol{\delta_0}$. Thus the set $\mathfrak{T}$ contains a neighborhood of $\boldsymbol{\delta_0}$. We can in fact show more.

Lemma 64. *The set $\mathfrak{T}$ is open. As a consequence $\mathfrak{M} \setminus \mathfrak{T}$ is compact.*

Proof. The last assertion is clear, since the complement of $\mathfrak{T}$ is closed and it is a subset of the compact space $\mathfrak{M}$.

Let $\mathbf{p} \in \mathfrak{T}$. We show that all points $\mathbf{p}' \in \mathfrak{M}$ at distance at most d, where d is still to be determined are inside $\mathfrak{T}$. Let n be such that $\mathscr{F}^{(n)}(\mathbf{p}) \in \mathfrak{B}(\epsilon_0^2)$, set $\chi = \epsilon_0^2 - d_C(\mathscr{F}^{(n)}(\mathbf{p}), \boldsymbol{\delta_0})$. Also, let γ be the constant in (E.8). Then setting $d = \chi\gamma^{-n}$ ensures that $F^{(n)}(\mathbf{p}') \in \mathfrak{B}(\epsilon_0^2)$, which places it in the basin of attraction of $\boldsymbol{\delta_0}$. □

We present here the last part of the calculations used in the two technical lemmas in the beginning.

Lemma 65. *The next identities hold as $\epsilon \to 0$ and $d = o(\epsilon^{-1.1})$, while Q is constant:*

$$\begin{aligned}
&\phi_d^{(Q\geq 4)}(\epsilon, \dots, \epsilon) = O((d\epsilon)^3),\\
&\phi_d^{(Q\geq 4)}(\frac{1}{Q}, \epsilon, \dots, \epsilon) = O((d\epsilon)^2),\\
&\phi_d^{(Q=3)}(\epsilon^2, \dots, \epsilon^2) = O(\epsilon^4),
\end{aligned}$$

$$\phi_d^{(Q=3)}(\epsilon,\epsilon^2,\dots,\epsilon^2) = O(d\epsilon^3),$$
$$\phi_d^{(Q=3)}(\frac{1}{Q},\epsilon^2,\dots,\epsilon^2) = O(d\epsilon^2),$$
$$\phi_d^{(Q=3)}(\epsilon,\epsilon,\epsilon^2,\dots,\epsilon^2) = O(d^2\epsilon^2),$$
$$\phi_d^{(Q=3)}(\frac{1}{Q},\epsilon,\epsilon^2,\dots,\epsilon^2) = O(d^2\epsilon).$$

Proof. In all cases above, we can disregard the denominator (see 3.33) of ϕ_d, since it is $1-o(1)$.

The first identity is obtained as follows: let $\zeta_1,\dots,\zeta_d$ be random colors with probability ϵ each and $*$ with probability $1-Q\epsilon$. Let N be the number of non-stars. Then the numerator of ϕ_d is the probability that $N = Q-1$. For this to happen, a necessary condition is that at least 3 of the ζ's need to be non-$*$. By the union bound, this probability is $O((d\epsilon)^3)$.

The second identity is obtained in a similar way: now ζ_1 is a random color with probability $1/Q$ each, and the rest of the ζ's are chosen as before. But now it is necessary that at least 2 of $\zeta_2,\dots,\zeta_d$ be non-star, which happens with probability $O((d\epsilon)^2)$.

For the case $Q=3$, we use the formula for the numerator of ϕ_d:

$$f_d(x_1,\dots,x_d) = T(1) - 2T(2) + T(3),$$

where $T(j) = \prod_j (1-jx_j)$, and pick the first order terms in $d\epsilon$. □

E.5 Properties of the complexity functional

E.5.1 Proof of Lemma 32 (Continuity)

Proof. Let $\mathbf{p},\mathbf{p}' \in \mathfrak{M}$. We have

$$\begin{aligned}
&\left|\int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1)\cdots\mathrm{d}\mathbf{p}(x_d)\log g(x_1,\dots,x_d) - \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}'(x_1)\cdots\mathrm{d}\mathbf{p}(x_d)\log g(x_1,\dots,x_d)\right| \\
&\le \int_{[0,1/Q]^{d-1}} \mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_d)\int_0^{1/Q}\mathrm{d}x\left|\mathbf{p}((-\infty,x]) - \mathbf{p}'((-\infty,x])\right|\frac{\partial}{\partial x}\log g(x,\dots,x_d) \\
&= O(d_C(\mathbf{p},\mathbf{p}')),
\end{aligned}$$

where in the last step we used the fact that the derivative of g is bounded (and it does not depend on d, see Lemma 25).

Using this we immediately derive the continuity of the first sum of Σ, while the second sum is treated in the same manner. □

E.5.2 Proof of Lemma 33 (Analyticity on line segment)

Proof. Note that the second term that appears in Σ is in fact a second-degree polynomial in t, so we only need to concentrate on the first term, which has the form

$$\begin{aligned}&\mathbb{E}_d \int_{[0,1/Q]^d} \prod_{i=1}^{d} (\mathrm{d}\mathbf{p} + t(\mathrm{d}\mathbf{p}' - \mathrm{d}\mathbf{p}))(x_i) \log g(x_1,\dots,x_d) \\ &= \mathbb{E}_d \sum_{k=0}^{d} t^k \binom{d}{k} \int_{[0,1/Q]^d} \prod_{i=1}^{k} \mathrm{d}\mathbf{p}(x_i) \prod_{i=k+1^d} (\mathrm{d}\mathbf{p}' - \mathrm{d}\mathbf{p})(x_i) \log g(x_1,\dots,x_d). \end{aligned} \tag{E.11}$$

We use the fact that $g(x_1,\dots,x_d)$ is bounded (Lemma 23), which gives $\log K + d\log(1-\frac{1}{Q}) \le \log g(x_1,\dots,x_d) \le 0$ for some constant K. By integrating g we obtain

$$\left| \int_{[0,1/Q]^d} \prod_{i=1}^{k} \mathrm{d}\mathbf{p}(x_i) \prod_{i=k+1^d} (\mathrm{d}\mathbf{p}' - \mathrm{d}\mathbf{p})(x_i) \log g(x_1,\dots,x_d) \right| \le 2^k |\log K + d\log(1-\frac{1}{Q})|.$$

Note that we have made no assumption on what exactly $\mathbf{p}$ is and in fact the previous inequality remains valid even if $x_1, x_2, \dots$ are distributed according to different distributions $\mathbf{p}_1, \mathbf{p}_2, \dots$. This observation is useful in proving the equivalent of this lemma for the coupled complexity functional.

The coefficient a_k of t^k in (E.11) is bounded in absolute value by

$$\sum_{d\ge k} \frac{\alpha^d e^{-\alpha}}{d!} \binom{d}{k} 2^k |\log K + d\log(1-\frac{1}{Q})| = e^{-O(k\log k)},$$

which ensures that the power series $\sum_{k\ge 0} a_k t^k$ converges everywhere. □

E.5.3 Proof of Lemma 36 ($\Sigma(\mathbf{p})$ is decreasing in α)

Proof. Differentiating with respect to α and using the identity (3.39), we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\Sigma^{(\alpha)}(\mathbf{p}) = \frac{\mathrm{d}}{\mathrm{d}\alpha}\Bigg[&\sum_{d\ge 0} \frac{\alpha^d e^{-\alpha}}{d!} \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1)\cdots\mathrm{d}\mathbf{p}(x_d) \log g(x_1,\dots,x_d) \\ &- \frac{\alpha}{2} \int_{[0,1/Q]^2} \mathrm{d}\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x) \log(1 - Q x_1 x) \Bigg]\end{aligned}$$

$$
\begin{aligned}
&= \int_{[0,1/Q]} \mathrm{d}\mathbf{p}(x_1) \Bigg[\mathbb{E}_d \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_2)\cdots\mathrm{d}\mathbf{p}(x_{d+1}) \big(\log g(x_2,\dots,x_{d+1}) \\
&\qquad + \log(1 - Q x_1 \phi(x_2,\dots,x_{d+1})) \big) - \frac{1}{2} \int_{[0,1/Q]} \mathrm{d}\mathbf{p}(x) \log(1 - Q x_1 x) \Bigg] \\
&= \mathbb{E}_d \int_{[0,1/Q]^d} \mathrm{d}\mathbf{p}(x_1)\cdots\mathrm{d}\mathbf{p}(x_d) \log g(x_1,\dots,x_d) \\
&\qquad + \frac{1}{2} \int_{[0,1/Q]^2} \mathrm{d}\mathbf{p}(x_1)\,\mathrm{d}\mathbf{p}(x_2) \log(1 - Q x_1 x_2).
\end{aligned}
$$

□

E.5.4 Proof of Lemma 36 (Negative complexity gap)

Proof. Suppose that $\mathbf{p}_* \neq \mathscr{F}(\mathbf{p}_*)$. Let us interpolate between $\mathbf{p}_*$ and $\mathscr{F}(\mathbf{p}_*)$, i.e. take $\mathbf{p}_* + t\delta\mathbf{p}$, as a function of t, where $\delta\mathbf{p} = \mathscr{F}(\mathbf{p}_*) - \mathbf{p}_*$. If $\frac{\mathrm{d}}{\mathrm{d}t}\Sigma(\mathbf{p}_* + t\delta\mathbf{p})|_{t=0}$ is negative, which we prove below, then there is $t_1 \in (0,1)$ such that $\mathbf{p}_1 = \mathbf{p} + t_1\delta\mathbf{p}$ and $\Sigma(\mathbf{p}_1) < \Sigma(\mathbf{p}_*)$, which yields a contradiction.

We are left to check the negativity of the first derivative, which we check by using Lemma 35:

$$
\frac{\mathrm{d}}{\mathrm{d}t}\Sigma(\mathbf{p} + t\delta\mathbf{p}) = \delta\Sigma(\mathbf{p})[\delta\mathbf{p}] = \alpha \int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}(x_1) \int_{[0,1/Q]} \mathrm{d}(\mathscr{F}(\mathbf{p}) - \mathbf{p})(x) \log(1 - Q x_1 x)\,.
$$

Expanding the logarithm, we obtain

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\Sigma(\mathbf{p} + t\delta\mathbf{p}) &= \alpha \int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}(x_1) \int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}(x) \left[-\sum_{j\geq 1} \frac{Q^j}{j} x_1^j x^j \right] \\
&= -\sum_{j\geq 1} \frac{Q^j}{j} \left(\int_{[0,1/Q]} \mathrm{d}\delta\mathbf{p}(x) x^j \right)^2 < 0.
\end{aligned}
$$

The fact that $\Delta\Sigma$ coincides with the infimum follows from the fact that $\underline{\mathbf{p}_*}$, being a fixed point, is not in $\mathfrak{T}$. □

F Sampling infinite permutations

This appendix deals with the problem arising in sampling the connection descriptors for a regular lift of a base formula in Chapter 4. This means we are looking for a method to sample a permutation $\phi : \mathsf{z} \to \mathsf{z}$ which satisfies the window constraint $\phi(z) - z \in \{0, \dots, W-1\}$ at every z.

Let the entries of the random permutation be denoted by random variables Y_z. Define the random variables T_z with values in $\{0,1\}^{W-1}$ as $T_{z;j} = \sum_{z'<z} \delta_{Y_{z'}, z+j}$, for $j = 0, \dots, W-2$. The bitstring T_z simply encodes the which of the values $\{z, \dots, z+W-2\}$ are occupied by "past" values Y (interpreting the spatial dimension as temporal). Every 1 in the string means "occupied" and every 0 "free".

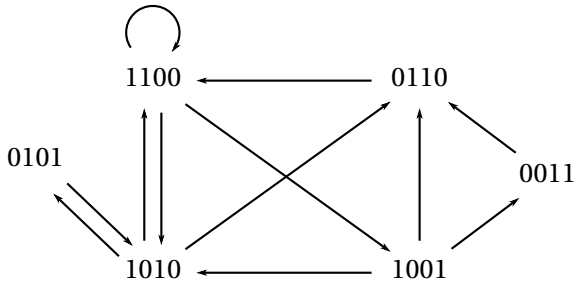
We make the following ansatz: the random variable Y_z is conditionally independent of $Y_{z-1}, Y_{z-2}, \dots$ given T_z. This way $\{T_z\}_z$ becomes a Markov chain and the values of Y_z can be deduced from the transitions of the chain. Note that the quantity $\Gamma = \sum_{j=0}^{W-2} T_{z;j}$ is invariant with respect to z. Thus the states of the Markov chain will be bit strings of length $W-1$ with a fixed weight Γ. Let w be a state of the chain. Let S be the shift-left operator, which moves all bits of w except the first one position to the left and adds a 0 at the end. Let R_j be the operator that changes the bit in position j to 1. Then the only transitions allowed in the Markov chain are (i) if w starts with a 0, then $S(w)$ and (ii) if w starts with a 1 then $S \circ R_j(w)$ for all j such that $w_j = 0$.

In the case $W = 2$, there are only two infinite permutations that satisfy the windowing constraint: the identity $z \mapsto z$ and $z \mapsto z+1$. The former has $\Gamma = 0$ and the latter $\Gamma = 1$. However, for $W \geq 3$, the behaviour is richer.

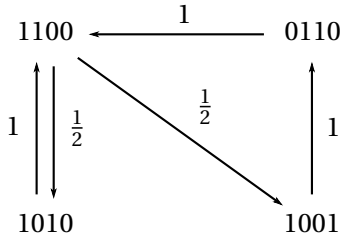
The Markov Chain approach to sampling the permutations enables us to generate the entries sequentially, whenever we need them. This is used in the Coupled-UCP algorithm to generate parts of the coupled structure as more positions are needed.

We used two methods in which to run the Markov chain, but in the simulations of Chapter 4 no difference is observed:

- After fixing W, Γ, set the Markov chain in such a way that all transitions that are allowed from a state have equal probability. Compute the stationary state of the Markov chain, sample T_0 from it, and then run the Markov chain starting from 0 (we are not interested in the negative part of the chain). Note that in this setting Y_z will not be distributed uniformly over $\{z, \ldots, z+W-1\}$. For $W = 5$ and $\Gamma = 2$ the full chain is given below.



- Engineer the degrees of freedom available in the transition probabilities of the Markov chain in such a way that the Y_z are uniformly distributed across the window. Then proceed as above. Finding such a Markov chain is not possible for all pairs (W, Γ), and in particular for $W = 4$ it is not possible at all. Also for $\Gamma = 0$ and $\Gamma = W - 1$ the chain is forced to do the same transition at every step, so this case is not so interesting. An example for $W = 5$ and $\Gamma = 2$ which could be used is given below.



This is the Markov Chain used to generate coupled lifts for the plots in Chapter 4

Bibliography

[Ach01] Dimitris Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 265(1):159–185, 2001.

[ACO08] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 793–802. IEEE, 2008.

[AF99] Dimitris Achlioptas and Ehud Friedgut. A sharp threshold for k-colorability. *Random Structures and Algorithms*, 14(1):63–70, 1999.

[AH89] Kenneth I. Appel and Wolfgang Haken. *Every planar map is four colorable*, volume 98. American mathematical society Providence, RI, 1989.

[AM97] Dimitris Achlioptas and Michael Molloy. The analysis of a list-coloring algorithm on a random graph. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 204–212. IEEE, 1997.

[AM13] Dimitris Achlioptas and Michael Molloy. The solution space geometry of random linear equations. *Random Structures and Algorithms*, 2013.

[ANP05] Dimitris Achlioptas, Assaf Naor, and Yuval Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435(7043):759–764, 2005.

[AP04] Dimitris Achlioptas and Yuval Peres. The threshold for random k-sat is $2^k \log_2 k - O(k)$. *Journal of the American Mathematical Society*, 17(4):947–973, 2004.

[Ari09] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009.

[BCOH+14] Victor Bapst, Amin Coja-Oghlan, Samuel Hetterich, Felicia Rassmann, and Dan Vilenchik. The condensation phase transition in random graph coloring. *arXiv preprint arXiv:1404.5513*, 2014.

[Bet35] Hans A. Bethe. Statistical theory of superlattices. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 150(871):552–575, 1935.

[BGT10] Mohsen Bayati, David Gamarnik, and Prasad Tetali. Combinatorial approach to the interpolation method and scaling limits in sparse random graphs. In *Proceedings of the 42nd ACM Symp. on Theory of Comp.*, STOC '10, pages 105–114, New York, NY, USA, June 2010. ACM.

[BMP+03] Alfredo Braunstein, Roberto Mulet, Andrea Pagnani, Martin Weigt, and Riccardo Zecchina. Polynomial iterative algorithms for coloring and analyzing random graphs. *Physical Review E*, 68(3):036702, 2003.

[BMZ05] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27(2):201–226, 2005.

[Bol01] Béla Bollobás. *Random Graphs*. Cambridge University Press, second edition, 2001. Cambridge Books Online.

[BZ04] Alfredo Braunstein and Riccardo Zecchina. Survey propagation as local equilibrium equations. *Journal of Statistical Mechanics: Theory and Experiment*, 2004:P06007, 2004.

[COE14] Amin Coja-Oghlan and Charilaos Efthymiou. On independent sets in random graphs. *Random Structures & Algorithms*, 2014.

[Coo71a] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[Coo71b] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[COV13] Amin Coja-Oghlan and Dan Vilenchik. Chasing the k-colorability threshold. In *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 380–389. IEEE, 2013.

[COZ12] Amin Coja-Oghlan and Lenka Zdeborová. The condensation transition in random hypergraph 2-coloring. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 241–250. SIAM, 2012.

[CR92] Vašek Chvátal and Bruce Reed. Mick gets some (the odds are on his side). In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 620–627. IEEE, 1992.

[DG84] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *The Journal of Logic Programming*, 1(3):267–284, 1984.

[DJM13] David L. Donoho, Adel Javanmard, and Andrea Montanari. Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing. *IEEE Transactions on Information Theory*, 59(11):7434–7464, 2013.

[DSS14] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. *arXiv preprint arXiv:1411.0650*, 2014.

[EA75] Samuel Frederick Edwards and Phil W. Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5(5):965, 1975.

[ER59] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[FB99] Ehud Friedgut and Jean Bourgain. Sharp thresholds of graph properties, and the k-sat problem. *Journal of the American Mathematical Society*, 12(4):1017–1054, 1999.

[FL03] Silvio Franz and Michele Leone. Replica bounds for optimization problems and diluted spin systems. *J. Stat. Phys.*, 111:535–564, 2003.

[FLMRT02] Silvio Franz, Michele Leone, Andrea Montanari, and Federico Ricci-Tersenghi. Dynamic phase transition for decoding algorithmsthe glassy phase of Gallager codes. *Phys. Rev. E*, 66:046120, 2002.

[FLT03] Silvio Franz, Michele Leone, and Fabio Lucio Toninelli. Replica bounds for diluted non-Poissonian spin systems. *J. Phys. A: Math. Gen.*, 36:10967—10985, 2003.

[FZ99] A. Jiménez Felström and Kamil Sh. Zigangirov. Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Transactions on Information Theory*, 45(5):2181–2190, September 1999.

[Gal63] Robert G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, USA, 1963.

[Gil59] Edgar Nelson Gilbert. Random graphs. *Ann. Math. Statist.*, 30(4):1141–1144, 12 1959.

[GM75] Geoffrey R. Grimmett and Colin J. H. McDiarmid. On colouring random graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 77, pages 313–324. Cambridge Univ Press, 1975.

[GMU12] Andrei Giurgiu, Nicolas Macris, and Rüdiger Urbanke. How to prove the Maxwell conjecture via spatial coupling – a proof of concept. In *Proceedings of the IEEE Int. Symp. Inf. Theory (ISIT) 2012*, pages 458–462, 2012.

[GMU13] Andrei Giurgiu, Nicolas Macris, and Rüdiger Urbanke. And now to something completely different: spatial coupling as a proof technique. In *Proceedings of the IEEE Int. Symp. Inf. Theory (ISIT) 2013*, 2013.

[GMU15] Andrei Giurgiu, Nicolas Macris, and Ruediger Urbanke. Spatial coupling as a proof technique and three applications. *submitted to IEEE Trans. Inform. Theory, arXiv preprint arXiv:1301.5676*, 2015.

[GT04] Francesco Guerra and Fabio Lucio Toninelli. The high temperature region of the Viana-Bray diluted spin glass model. *J. Stat. Phys.*, 115(1/2):501–555, April 2004.

[HMU13] S Hamed Hassani, Nicolas Macris, and Ruediger Urbanke. Threshold saturation in spatially coupled constraint satisfaction problems. *Journal of Statistical Physics*, 150(5):807–850, 2013.

[Isi25] Ernst Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[JLR11] Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[Kau48] Walter Kauzmann. The nature of the glassy state and the behavior of liquids at low temperatures. *Chemical Reviews*, 43(2):219–256, 1948.

[Kem79] Alfred B Kempe. On the geographical problem of the four colours. *American journal of mathematics*, 2(3):193–200, 1879.

[KFL01] Frank R. Kschischang, Brendan J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory,*, 47(2):498–519, 2001.

[KKM07] Satish Babu Korada, Shrinivas Kudekar, and Nicolas Macris. Exact solution for the conditional entropy of poissonian ldpc codes over the binary erasure channel. In *Proceedings of the IEEE Int. Symp. Inf. Theory (ISIT) 2007*, pages 1016–1020, 2007.

[KM01] Shrinivas Kudekar and Nicolas Macris. Decay of correlations for sparse error correcting codes. *SIAM J. Discrete Math.*, 25:956–988, 2001.

[KM09] Shrinivas Kudekar and Nicolas Macris. Sharp bounds for optimal decoding of low-density parity-check codes. *IEEE Trans. Inform. Theory*, 55(10):4635–4650, October 2009.

[KMS00] Yoshiyuki Kabashima, Tatsuto Murayama, and David Saad. Typical performance of Gallager-type error-correcting codes. *Phys. Rev. Lett.*, 84:1355, 2000.

[KPW04] Florent Krząkała, Andrea Pagnani, and Martin Weigt. Threshold values, stability analysis, and high-q asymptotics for the coloring problem on random graphs. *Physical Review E*, 70(4):046705, 2004.

[KRU11] Shrinivas Kudekar, Thomas J Richardson, and Rüdiger L Urbanke. Threshold saturation via spatial coupling: Why convolutional ldpc ensembles perform so well over the bec. *IEEE Transactions on Information Theory*, 57(2):803–834, 2011.

[KRU12] Shrinivas Kudekar, Tom Richardson, and Ruediger Urbanke. Spatially coupled ensembles universally achieve capacity under belief propagation. E-print arXiv:1201.2999, January 2012.

[KYMP12] Santhosh Kumar, Andrew J. Young, Nicolas Macris, and Henry D. Pfister. A proof of threshold saturation for spatially-coupled LDPC codes on BMS channels. *Proc. Annual Allerton Conference on Communication, Control and Computation*, 2012.

[KYMP14] Santhosh Kumar, Andrew J. Young, Nicolas Macris, and Henry D. Pfister. Threshold saturation for spatially coupled ldpc and ldgm codes on bms channels. *IEEE Transactions on Information Theory*, 60(12):7389–7415, Dec 2014.

[KZ09] Florent Krząkała and Lenka Zdeborová. Hiding quiet solutions in random constraint satisfaction problems. *Physical review letters*, 102(23):238701, 2009.

[LF10] Michael Lentmaier and Gerhard P Fettweis. On the thresholds of generalized ldpc convolutional codes based on protographs. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 709–713. IEEE, 2010.

[LSCJZ10] Michael Lentmaier, Arvind Sridharan, Daniel J Costello Jr, and Kamil Zigangirov. Iterative decoding threshold analysis for ldpc convolutional codes. *IEEE Transactions on Information Theory*, 56(10):5274–5289, 2010.

[Łuc91] Tomasz Łuczak. The chromatic number of random graphs. *Combinatorica*, 11(1):45–54, 1991.

[MÓ6] Cyril Méasson. *Conservation laws for coding, PhD Thesis*. EPFL, 2006.

[Mac07] Nicolas Macris. Griffith-Kelly-Sherman correlation inequalities: a useful tool in the theory of error correcting codes. *IEEE Transactions on Information Theory*, 53(2):664–683, February 2007.

[MKSV00] Tatsuto Murayama, Yoshiyuki Kabashima, David Saad, and Renato Vicente. Low-density parity-check error-correcting codes. *Phys. Rev. E*, 62:1577, 2000.

[MM06] Marc Mézard and Andrea Montanari. Reconstruction on trees and spin glass transition. *Journal of statistical physics*, 124(6):1317–1350, 2006.

[MM07] Marc Mézard and Andrea Montanari. *Information, Physics, and Computation*. Clarendon Press, Oxford, 2007.

[MMRU09] Cyril Méasson, Andrea Montanari, Thomas J. Richardson, and Rüdiger Urbanke. The generalized area theorem and some of its consequences. *IEEE Transactions on Information Theory*, 55(11):4793–4821, November 2009.

[MMU08] Cyril Méasson, Andrea Montanari, and Rüdiger Urbanke. Maxwell construction: The hidden bridge between iterative and maximum a posteriori decodingdecay of correlations for sparse error correcting codes. *IEEE Transactions on Information Theory*, 54(12):5277–5307, 2008.

[MMW07] Elitza Maneva, Elchanan Mossel, and Martin J Wainwright. A new look at survey propagation and its generalizations. *Journal of the ACM (JACM)*, 54(4):17, 2007.

[MMZ05] Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94(19):197205, 2005.

[Mol12] Michael Molloy. The freezing threshold for k-colourings of a random graph. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 921–930. ACM, 2012.

[Mon01] Andrea Montanari. The glassy phase of Gallager codes. *Phys. J. B*, 23:121, 2001.

[Mon05] Andrea Montanari. Tight bounds for LDPC and LDGM codes under MAP decoding. *IEEE Transactions on Information Theory*, 51:3221–3246, 2005.

[MP01] Marc Mézard and Giorgio Parisi. The bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20(2):217–233, 2001.

[MP03] Marc Mézard and Giorgio Parisi. The cavity method at zero temperature. *Journal of Statistical Physics*, 111(1-2):1–34, 2003.

[MPR05] Marc Mézard, Matteo Palassini, and Olivier Rivoire. Landscape of solutions in constraint satisfaction problems. *Physical review letters*, 95(20):200202, 2005.

[MPRT04] Andrea Montanari, Giorgio Parisi, and Federico Ricci-Tersenghi. Instability of one-step replica-symmetry-broken phase in satisfiability problems. *Journal of Physics A: Mathematical and General*, 37(6):2073, 2004.

[MPV87] Marc Mezard, Giorgio Parisi, and Miguel Ángel Virasoro. World Scientific lecture notes in physics: Vol. 9., Spin glass theory and beyond, 1987.

[MPWZ02] Roberto Mulet, Andrea Pagnani, Martin Weigt, and Riccardo Zecchina. Coloring random graphs. *Physical review letters*, 89(26):268701, 2002.

[MPZ02] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[MR13] Michael Molloy and Ricardo Restrepo. Frozen variables in random boolean constraint satisfaction problems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1306–1318. SIAM, 2013.

[MRTS07] Andrea Montanari, Federico Ricci-Tersenghi, and Guilhem Semerjian. Solving constraint satisfaction problems through belief propagation-guided decimation. *arXiv preprint arXiv:0709.1667*, 2007.

[MRTS08] Andrea Montanari, Federico Ricci-Tersenghi, and Guilhem Semerjian. Clusters of solutions and replica symmetry breaking in random k-satisfiability. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04):P04004, 2008.

[MZ02] Marc Mézard and Riccardo Zecchina. The random k-satisfiability problem: from an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66:056126, 2002.

[Ons36] Lars Onsager. Electric moments of molecules in liquids. *Journal of the American Chemical Society*, 58(8):1486–1493, 1936.

[Pap03] Christos H. Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[Par80] Giorgio Parisi. A sequence of approximated solutions to the sk model for spin glasses. *Journal of Physics A: Mathematical and General*, 13(4):L115, 1980.

[Pei36] Rudolf Peierls. Statistical theory of adsorption with interaction between the adsorbed atoms. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 32, pages 471–476. Cambridge Univ Press, 1936.

[RU08] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. Cambridge University Press, March 2008.

[Sem08] Guilhem Semerjian. On the freezing of variables in random constraint satisfaction problems. *Journal of Statistical Physics*, 130(2):251–293, 2008.

[Sha48] Claude Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[SK75] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792, 1975.

[SLCJZ04] Arvind Sridharan, Michael Lentmaier, Daniel J Costello Jr, and Kamil Zigangirov. Convergence analysis for a class of ldpc convolutional codes on the erasure channel. In *Annual Allerton Conference on Communication, Control and Computing (Allerton)*, pages 953–962, 2004.

[Sly09] Allan Sly. Reconstruction for the Potts model. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 581–590, New York, NY, USA, 2009. ACM.

[Tal03] Michel Talagrand. *Spin glasses: a challenge for mathematicians: cavity and mean field models*. Springer, 2003.

[VMS02] Jort Van Mourik and David Saad. Random graph coloring: Statistical physics approach. *Physical Review E*, 66(5):056120, 2002.

[Wu82] Fa-Yueh Wu. The Potts model. *Reviews of modern physics*, 54(1):235, 1982.

[YJNP12] Arvind Yedla, Yung-Yih Jian, Phong S. Nguyen, and Henry D. Pfister. A simple proof of threshold saturation for coupled scalar recursions. In *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*, pages 51–55, Aug 2012.

[ZK07] Lenka Zdeborová and Florent Krząkała. Phase transitions in the coloring of random graphs. *Physical Review E*, 76(3):031131, 2007.

Andrei Giurgiu

Personal Information

- Address: Rue de l'Industrie 18, 1023 Crissier, Switzerland
- Tel: (+41) 21 545 72 09
- Date of Birth: November 2, 1984, Cluj Napoca, Romania
- Citizenship: Romanian

Education

Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland 2010 -

- *PhD* in the *Ecole Doctorale Informatique et Communications (EDIC).*

Swiss Federal Institute of Technology (ETHZ), Zurich, Switzerland 2008 - 2010

- *Master of Science* in Computer Science.

Jacobs University, Bremen, Germany 2005 - 2008

- *Bachelor of Science*, double major in Electrical Engineering and Computer Science (EECS) and Mathematics.

Research Experience

EPFL, Theory of Communications Laboratory (LTHC), Lausanne, Switzerland
Research Assistantship February 2011 - present

- Advisors: Prof. Dr. Rüdiger Urbanke, Dr. Nicolas Macris
- Thesis: Sparse Probabilistic Models: Phase Transitions and Solutions via Spatial Coupling

EPFL, Data Analysis Theory and Applications Laboratory, Lausanne, Switzerland
Research Assistantship September 2010 - January 2011

- Supervisor: Prof. Dr. Christoph Koch
- Topic: dynamical complexity classes for queries on databases and investigating the expressiveness of various query languages.

EPFL, Laboratory of Distributed Programming, Lausanne, Switzerland
Research Assistantship October 2009 - August 2010

- Supervisor: Prof. Dr. Rachid Guerraoui
- Topics: theoretical models for distributed computing to provide for anonymous computation, which are less secure than cryptographic approaches, but more scalable and lightweight; tight bounds for the renaming problem in distributed systems.

ETHZ, Institute of Theoretical Computer Science, Zurich, Switzerland
Master Thesis Project March 2009 - September 2009

- Supervisors: Prof. Dr. Emo Welzl, Dr. Robin Moser
- Master thesis: Random Walk Algorithms for SAT
- Algorithms for boolean satisfiability that are effective in the worst case are randomized. In this work I have focused on the development of theoretical tools that aid the analysis of such algorithms, and used them to obtain tight bounds on different variants of Schoening's algorithm.

Jacobs University, EECS, Bremen, Germany
Guided Research in Computer Science Spring 2008

- Supervisor: Prof. Dr. Herbert Jaeger
- Bachelor Thesis title: Observable Operator Models for Nonstationary Symbol Sequences

Jacobs University, Department of Mathematics, Bremen, Germany
Guided Research in Mathematics Fall 2007

- Supervisor: Prof. Dr. Götz Pfander
- Project title: A Low Dimensional Investigation on Gabor Systems

Robotics Lab of Jacobs University, Bremen, Germany
Summer Project Summer 2006

- Supervisor: Prof. Dr. Andreas Birk
- Jacobs University team for the Student Underwater Challenge, in Pinewood Studios, near London. The task was to build an autonomous mini-submarine, capable of performing various tasks (like moving around, hitting targets, etc.) in a specified order. My work there was on vision processing and on software and hardware for some controllers.

German Institute for Artificial Intelligence (DFKI), Bremen, Germany
Internship June 2006 - January 2007

- Supervisor: Dr. Christoph Lueth
- The project in the Secure Cognitive Systems Lab was concerned with the safety of programs. C code needed to be annotated with facts asserting correct execution, then processed and piped to a theorem prover (Isabelle), in order to determine whether the annotated facts indeed hold under all possible executions of the program. My task consisted of designing and implementing the translation of the annotated code to a form understandable by the theorem prover. This was done in Haskell, a functional programming language.

Teaching Experience

Jacobs University, Bremen, Germany
Teaching Assistant Fall 2005 - Spring 2008

- Algorithms and Data Structures Lab, Formal Languages and Logic, Graphics and Visualisation, Electronics, NatSciLab Unit CS I, Computability and Complexity, Algorithms and Data Structures Lab

EPFL, Lausanne, Switzerland
Doctoral Assistant Fall 2011 - Spring 2014

- Discrete Structures, Traitement Quantique de l'Information, Circuits et Sistèmes II, Applied Probability and Stochastic Processes

Peer-reviewed Publications

- *Spatial Coupling as a Proof Technique and Three Applications*
  Andrei Giurgiu, Nicolas Macris, Rüdiger Urbanke
  IEEE Transactions on Information Theory, under review, preprint available: arXiv:1301.5676, submitted 2014
- *Computing in social networks*, Andrei Giurgiu, Rachid Guerraoui, Kévin Huguenin and Anne-Marie Kermarrec, in Information And Computation, vol. 234, p. 3-16, 2014
- *Statistical estimation: from denoising to sparse regression and hidden cliques*
  Eric W. Tramel, Santhosh Kumar, Andrei Giurgiu, Andrea Montanari
  in *Statistical Physics, Optimization, Inference and Message-Passing: Lecture Notes of the Les Houches Summer School: Special Issue*
  edited by Florent Krzakala, Federico Ricci-Tersenghi, Lenka Zdeborová, Riccardo Zecchina and Eric Tramel, *Oxford University Press*, 2015
- *And now to something completely different: spatial coupling as a proof technique*
  Andrei Giurgiu, Nicolas Macris, Rüdiger Urbanke
  ISIT 2013
- *How to prove the Maxwell Conjecture via spatial coupling*
  Andrei Giurgiu, Nicolas Macris, Rüdiger Urbanke
  ISIT 2012

- *Fast Randomized Test-and-Set and Renaming*
  Dan Alistarh, Hagit Attiya, Seth Gilbert, Andrei Giurgiu, Rachid Guerraoui
  DISC 2010

- *Computing in Social Networks*
  Andrei Giurgiu, Rachid Guerraoui, Kvin Huguenin, Anne-Marie Kermarrec
  SSS 2010

Contributed Talks

- International Symposium on Information Theory, Istanbul, Turkey 2013
- 16th Joint Conference on Communications and Coding, Holzgau, Austria 2013
- International Symposium on Information Theory, Boston, USA 2012

Other Conferences Attended

- Phase transitions in discrete structures and computational problems, Warwick, UK 2014
- Statistical physics, Optimization, Inference and Message-Passing algorithms, Les Houches, France 2013
- Information Theory Workshop, Lausanne, Switzerland 2012
- Statistical Physics of Complexity, Optimization and Systems Biology, Les Houches, France 2012
- Symposium on Principles of Distributed Computing (PODC), Zurich, Switzerland 2010
- 58th Meeting of Nobel Laureates in Lindau, Germany 2008

Computer Skills

- *Programming:* C, C++, Python, Java, Pascal, Prolog, Haskell, Common Lisp, Matlab, Mathematica;
- *Operating Systems:* UNIX-based (Linux, Solaris, Mac OS X), Microsoft Windows
- *Formatting Languages:* LaTeX
- *Applications:* Editors, open source programming tools (various compilers, debuggers, interpreters), database servers, multimedia and office applications

Languages

- English, Romanian — fluent in written and spoken
- French, German — good in written and spoken (B2)
- Italian, Spanish — basic knowledge

Honors and Prizes

- Outstanding Teaching Award, EPFL 2013
- Member of the President's List during the three academic years 2005 - 2008
- Study scholarship from Jacobs University, 2005 - 2008
- Merit scholarship award from the Polytechnic of Bucharest, 2003 - 2005
- National Olympiad of Computer Science 2000-2003, top places (2003, third place)
- Fifth place in the Southeastern European Regional Contest of the ACM 2004
- Twelfth place in the Northwestern European Regional Contest of the ACM 2005
- Third Prize in the International Mathematics Competition (IMC), in Blagoevgrad, 2007