LDPC Codes for Discrete Integration

Megasthenis Asteris, Alexandros G. Dimakis
Department of Electrical and Computer Engineering
The University of Texas at Austin
megas@utexas.edu, dimakis@austin.utexas.edu

***Abstract*—Discrete integration over exponentially large combinatorial sets is a key task for approximate probabilistic inference. Ermon *et al.* recently developed an approximation algorithm for discrete integration with provable guarantees using hashing techniques. These works reduce discrete integration to solving a polynomial number of MAP inference queries for models that include random linear parity check constraints. Empirical evidence suggests these inference queries can be solved much more efficiently when the linear parity constraints involve only a few variables. The challenge is how to obtain the required hashing properties with codes that are as sparse as possible.**

In this work, we establish a connection between Average Universal (AU) hashing and low-density parity-check codes; we show how the desired statistical properties of Average Universal (AU) hash families translate to properties of the average distance distribution of the code ensemble. Using this connection, we show that random independent LDPC codes with logarithmic degrees suffice to yield AU hash families with the required properties.

I. Introduction

Model counting and discrete integration over high-dimensional spaces is a frequently encountered task in machine learning and statistics. For example, computing the partition function of a graphical model defined on n boolean variables –a central object in probabilistic inference tasks– requires evaluating and summing a nonnegative function over all possible 2^n boolean configurations. Unfortunately this problem is #P Hard, *i.e.*, believed to be significantly harder than NP-complete problems. For this reason many techniques including MCMC [1], [2] and variational [3], [4] methods have been proposed as approximations.

In a recent line of work, Ermon *et al.* pioneered a novel approach for approximate counting and discrete integration over exponentially large sets with provable guarantees, based on Universal Hashing [5]–[7]. The key idea is to recast the computation into a tractable number of instances of an NP-hard combinatorial optimization problem over random subsets of the original domain determined by the random hash functions. In [5]–[7], these random hash functions take the form of linear parity constraints which randomly partition the domain into buckets succinctly described as sets of solutions to underdetermined systems of linear equations over $\mathrm{GF}(2)$. Their algorithm relies on an *optimization oracle* for the NP-hard subproblem, *i.e.*, it assumes the existence of an efficient black-box that can be queried to solve the combinatorial optimization. Utilizing the partial solutions, the algorithm finally computes a constant factor approximation to the solution of the original problem –a substantially tight guarantee given that the latter may be exponentially large. From a theoretical perspective, this is a significant accomplishment even under the seemingly strong oracle assumption, as the original problem is believed to be even harder than the NP-hard subproblems.

In practice, the algorithm achieves remarkable accuracy leveraging the advances in combinatorial optimization software; depending on the task at hand, solvers for Integer Linear Programming, Satisfiability (SAT) problems, or other dedicated software play the role of the NP-oracle. Linear parities have the additional benefit of being conveniently incorporated into problems of this form.

A simple model for generating the random linear parities achieves the desired theoretical guarantees [5], [6]: every single parity independently includes each one of the variables with probability $p = {}^1\!/_2$. Despite the theoretical guarantees, however, the expected $\Theta(n)$-degree hinders the algorithm performance in practice. Empirical evidence suggests that the optimization solvers achieve substantially higher accuracy when the parities involved in the optimization task are *sparser*, *i.e.*, when they involve fewer variables. Intuitively, this is due to the fact that although all linear parities reduce the feasible region by half regardless of their degree, a higher degree implies more options to be checked by the combinatorial solver at every step of the optimization.

In [7], the authors showed that low-density parities generated as before but with $p < 1/2$ correspond to a weaker class of hash functions referred to as Average Universal whose statistical properties suffice to match the previous theoretical guarantees. The same work established a criterion –a numerical recipe– to determine the optimal (lowest) value for p that achieves the desired properties. In [7], however, it remained unknown how the optimal p scales with the dimension n.

In the same direction, [8] introduced the idea of substituting the independently generated parities with Low-Density Parity-Check (LDPC) codes. Furthermore, they initiated the theoretical analysis of such ensembles and empirically observed that random parities generated according to the Progressive Edge Growth (PEG) construction [9] achieve drastic speedup and improved accuracy over previous methods, at least in certain regimes. Inspired by this previous work we explore what property of an LDPC ensemble creates the required hashing properties and investigate how much sparsity can be achieved.

Our Contributions Our first technical contribution is an analysis of an *i.i.d.* random linear binary code ensemble (also used in [7]). We show that when $\Theta(n)$ parities are generated, with every parity independently including each variable with probability $p = O(\log n/n)$, the random ensemble yields an Average Universal Hashing family with sufficiently good statistical properties. In other words, we show that parities with logarithmic expected degree suffice to yield the hash functions that work well for inference tasks. This is a substantial improvement over previous work[1] which could provably obtain results only for linear parity degrees [5].

Our second contribution is a formal connection between random linear error correcting codes and Average Universal (AU) Hashing. We show that the desired statistical properties of AU families of hash functions translate to properties of the expected weight enumerator of the ensemble. This connection is a first step towards designing LDPC codes that perform well for probabilistic inference tasks.

II. Preliminaries

We begin with a brief overview of the algorithm of [5]–[7] for a counting problem to motivate the subsequent discussion. Consider n boolean variables, and let $S \subseteq \{0,1\}^n$ be a large, but succinctly described subset of all 2^n possible configurations, *e.g.*, the set of solutions to a given Boolean formula. The problem of computing $|S|$ is known as *model counting*.

Let $\mathcal{H}^{(i)} \triangleq \{h : \{0,1\}^n \to \{0,1\}^i\}$ denote a family of hash functions; each function h partitions the space of possible configurations into 2^i buckets. Assume that we have such a family for each $i \in [n]$. The following procedure, hereby referred to as algorithm $\mathcal{A}$, computes an estimate of $|S|$. For $i = 1, \dots, n$, fix an arbitrary bucket $\mathbf{y} \in \{0,1\}^i$, and repeat the following T times: select randomly a hash function h from $\mathcal{H}^{(i)}$ according to a given distribution, and compute $X = |h^{-1}(\mathbf{y}) \cap S|$, *i.e.*, the number of points in S that are hashed to bucket $\mathbf{y}$; we assume that there exists an oracle to do that. If for the current i more than half of the T attempts find $X = 0$, then terminate the algorithm returning 2^{i-1} as the estimate of $|S|$. Otherwise, proceed to the $(i+1)$th round.

The intuition behind the algorithm is as follows. Our goal is to estimate the size of an unknown set S. In each iteration, we throw "blankets" that have random shape, but certain size, and estimate how many of them intersect with S. Each blanket is the set of configurations hashed to bucket $\mathbf{y}$, and will be a coset of a linear code. We progressively throw smaller blankets, reducing their size by half in each iteration, and stop the moment the blankets become so small that fail to hit S most of the time. This procedure allows us to estimate the size of S.

The quality of the estimate depends on the concentration of the random variable X and in turn the statistical properties of the hash families $\mathcal{H}^i$. Intuitively, we want the hash functions to partition the domain in a way which approximates a fully random partition.

Def. 1 (Average Universal Hashing [7])**.** *A set of functions*

$$\mathcal{H} \triangleq \{h : \{0,1\}^n \to \{0,1\}^m\}$$

along with a distribution $\mathcal{D}$ over $\mathcal{H}$, is an (ϵ, q)-AU hash family for some $\epsilon \geq 1/2^m$, if the following two conditions hold when $h \in \mathcal{H}$ is randomly chosen from $\mathcal{H}$ according to $\mathcal{D}$:

1) *$h(\mathbf{x})$ is uniformly distributed in $\{0,1\}^m$, $\forall \mathbf{x} \in \{0,1\}^n$,*
2) *$\forall \mathbf{y}, \mathbf{y}' \in \{0,1\}^m$ and $\forall S \subseteq \{0,1\}^n : |S| = q$,*

$$\sum_{\substack{\mathbf{x},\mathbf{x}' \in S \\ \mathbf{x} \neq \mathbf{x}'}} \mathbb{P}(h(\mathbf{x}) = \mathbf{y}, h(\mathbf{x}') = \mathbf{y}') \leq |S|(|S|-1)\frac{\epsilon}{2^m}.$$

We refer to the first condition as *Uniformity* and the second as *Universality*. One can verify that if $\mathcal{H}$ is an (ϵ, q)-AU family, then it is also (ϵ, q')-AU, $\forall q' > q$. An $(\epsilon, 2)$-AU family is widely known as ϵ-Strongly Universal (SU). Finally, we define the following weaker family of hash functions:

Def. 2 (Weak AU Hashing)**.** *The definition of Weak Average Universal (WAU) is similar to Def. 1, except we only require that condition 2 holds for $\mathbf{y} = \mathbf{y}'$.*

Interestingly, for algorithm $\mathcal{A}$ to work with the desired provable approximation guarantees, it suffices that the hash families $\mathcal{H}^{(i)}$ are (ϵ, q)-WAU with certain ϵ and q parameters. The following proposition which is compiled combining several arguments in [7] states it more formally:

Proposition II.1. *For $\delta > 2$, $c > 0$, and $\epsilon \geq 2^{-i}$, if the family $\mathcal{H}^{(i)} = \{h : \{0,1\}^n \to \{0,1\}^i\}$, is $(\epsilon, 2^{i+c})$-WAU with $\epsilon \leq 2^{-i}\big(1 + (\delta-1)^{-1} - 2^{-c}\big)$ for all $i = 1, \dots, n$, then $\mathcal{A}$ using $\mathcal{H}^i$ and $T = O(\log n)$ outputs a 2^c-factor approximation of $|S|$ with probability at least $1 - 1/\delta$.*

Therefore, in order to obtain theoretical approximation guarantees for algorithm $\mathcal{A}$, it suffices to design $(\epsilon, 2^{i+c})$-WAU hash families $\forall i \in [n]$, for some given parameters ϵ and c.

III. Distance Distribution and Hashing

We consider families of hash functions defined by *random linear parity* constraints; each function $h \in \mathcal{H}$ is of the form

$$h(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b},$$

where $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ and $\mathbf{b} \in \mathbb{F}_2^m$, and partitions the space of binary sequences $\mathbf{x} \in \{0,1\}^n$ into 2^m buckets. Thinking of $\mathbf{A}$ as a parity check matrix, these hash buckets correspond to cosets of a linear code. Conversely, bucket $\mathbf{y} \in \mathbb{F}_2^m$ consists of all sequences $\mathbf{x}$ that satisfy the linear system $\mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{y}$.

Randomly selecting a function h from $\mathcal{H}$ according to a specified distribution $\mathcal{D}$ is equivalent to randomly generating $\mathbf{A}$ and $\mathbf{b}$ according to that distribution. We only consider the case where $\mathbf{A}$ and $\mathbf{b}$ are generated independently, with $\mathbf{b}$ uniformly distributed over $\mathbb{F}_2^m$. The distribution of the random $m \times n$ matrix $\mathbf{A}$ defines a random ensemble of linear codes with $\mathbf{A}$ as the parity check matrix. The random vector $\mathbf{b}$ can be simply thought of as an affine shift that permutes the bucket labels in the partition induced by $\mathbf{A}$; *e.g.*, without this shift, the sequence $\mathbf{x} = \mathbf{0}_n$ would always be hashed to bucket $\mathbf{0}_m$.

It is implied by the above, that a random ensemble of linear codes –meaning a distribution over the parity check matrix $\mathbf{A}$– defines a family of random hash functions of the

[1] In parallel to our work, a very similar result on logarithmic sparsity appeared very recently [10] through a different proof.

form $h(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ with an accompanying distribution over the functions, and vice-versa. We want to understand the properties that a random linear ensemble must have, so that the corresponding hash family is (ϵ, q)-AU or at least (ϵ, q)-WAU, for given parameters ϵ and q.

Uniformity, *i.e.*, the first condition in Def. 1, is immediately satisfied by the presence of the affine shift $\mathbf{b}$, regardless the distribution of $\mathbf{A}$. On the contrary, the second condition relies solely on the distribution of $\mathbf{A}$: for any set $S \subseteq \mathbb{F}_2^n$ and vectors $\mathbf{y}$, $\mathbf{y}' \in \mathbb{F}_2^m$,

$$\begin{aligned}
&\sum_{\substack{\mathbf{x},\mathbf{x}'\in S\\ \mathbf{x}\neq\mathbf{x}'}} \mathbb{P}(h(\mathbf{x}) = \mathbf{y}, h(\mathbf{x}') = \mathbf{y}')\\
&= 2^{-m}\cdot \sum_{\substack{\mathbf{x},\mathbf{x}'\in S\\ \mathbf{x}\neq\mathbf{x}'}} \sum_{\mathbf{b}\in\mathbb{F}_2^m} \mathbb{P}\big(\mathbf{A}\mathbf{x}+\mathbf{b} = \mathbf{y}, \mathbf{A}\mathbf{x}'+\mathbf{b} = \mathbf{y}' \big| \mathbf{b}\big)\\
&= 2^{-m}\cdot \sum_{\substack{\mathbf{x},\mathbf{x}'\in S\\ \mathbf{x}\neq\mathbf{x}'}} \mathbb{P}(\mathbf{A}(\mathbf{x}-\mathbf{x}') = \mathbf{y}-\mathbf{y}')\\
&= 2^{-m}\cdot \sum_{\mathbf{x}\in S} \sum_{\mathbf{x}'\in S/\{\mathbf{x}\}} \mathbb{P}(\mathbf{A}(\mathbf{x}-\mathbf{x}') = \mathbf{y}-\mathbf{y}'), \quad (1)
\end{aligned}$$

where the randomness is over the choice of $\mathbf{A}$.

For given parameters ϵ and q, the hash family defined by a given distribution on $\mathbf{A}$ is (ϵ, q)-AU if (1) lies below a specific threshold, namely $q(q-1)\cdot\epsilon/2^m$, for all choices of S such that $|S| = q$, and all choices of $\mathbf{y}$ and $\mathbf{y}'$. For (ϵ, q)-WAU, the same bound must hold except only for the case $\mathbf{y}' = \mathbf{y}$.

We focus on Weak Average Universality. For $\mathbf{y}' = \mathbf{y}$, each term in (1) is of the form $\mathbb{P}(\mathbf{A}(\mathbf{x}-\mathbf{x}') = \mathbf{0})$, which is the probability that a fixed sequence $\boldsymbol{\tau} = \mathbf{x}-\mathbf{x}'$ is a codeword; the randomness lies in the choice of the linear code from the ensemble. Then, we have the following worst case (*i.e.*, for all choices of $S : |S| = q$ and $\mathbf{y}$) upper bound on (1):

$$(1) \le 2^{-m}\cdot q\cdot \max_{\substack{T\subseteq\mathbb{F}_2^n/\{\mathbf{0}\}\\ |T|=q-1}} \sum_{\boldsymbol{\tau}\in T} \mathbb{P}(\mathbf{A}\boldsymbol{\tau} = \mathbf{0}). \quad (2)$$

To compute the above upper bound we must determine a worst case set T of $q-1$ distinct nonzero binary sequences $\boldsymbol{\tau}$ that have the highest probability of belonging to a code randomly drawn from the ensemble.

In the sequel, we consider random ensembles for which the probability that any fixed sequence $\boldsymbol{\tau}$ is a codeword depends only on its Hamming weight $w(\boldsymbol{\tau})$. Then, we can resort to the average distance distribution of the random ensemble to derive an upper bound on (2). The average distance distribution (or weight enumerator) of a random ensemble is a function $f(w) : [n] \to \mathbb{R}_+$ such that $f(w)$ equals the expected number of codewords of weight exactly equal to w in a code drawn randomly from that ensemble. For ensembles satisfying the above assumption, the function $r(w) = f(w)/\binom{n}{w}$ equals the probability that a given sequence $\boldsymbol{\tau}$ of weight w is a codeword.

Utilizing the average distance distribution, we obtain an upper bound on (2) via a simple packing argument. Let $w_1, \ldots, w_n$ be a labeling of all weights in $[n]$ such that $r(w_1) \ge r(w_2) \ge \ldots \ge r(w_n)$. Further, define

$$i_\star \triangleq \max\left\{ j \in [n] : \sum_{i=1}^{j} \binom{n}{w_i} \le q-1 \right\}. \quad (3)$$

Then, the maximization in (2) is upper bounded by

$$\sum_{i=1}^{i_\star} \binom{n}{w_i} r(w_i) + \left(q-1-\sum_{i=1}^{i_\star}\binom{n}{w_i} \right) r(w_{i_\star+1}). \quad (4)$$

Substituting the maximization in (2) with the expression in (4), we obtain an upper bound on (1) which depends only on the average distance distribution of the random ensemble, or equivalently $r(w)$. In turn, for given parameters q and ϵ, a random code ensemble defines an (ϵ, q)-WAU hash family if expression (4) is at most equal to $(q-1)\epsilon$.

In summary, we have obtained a criterion for checking whether a random code ensemble corresponds to a WAU hash family with specific parameters utilizing only properties of its average distance distribution:

Theorem 1. *Let $\mathcal{C}$ be a random ensemble of n-dimensional linear codes with rate $1 - m/n$, or equivalently a distribution over the $m \times n$ parity check matrix $\mathbf{A}$. Let $\mathcal{H}_\mathcal{C}$ be the random linear hash family defined by that distribution on $\mathbf{A}$. For any given parameters $\epsilon > 1/2^m$ and q, if the average distance distribution of $\mathcal{C}$ is such that* (4) *is at most equal to $(q-1)\cdot\epsilon$, then $\mathcal{H}_\mathcal{C}$ is an (ϵ, q)-WAU family.*

This allows us to potentially leverage the extensive literature on bounds on average distance distributions, *e.g.*, [11]–[17] and references therein, to find families of random codes that provably yield statistically useful WAU hash families.

IV. Hashing with Logarithmic Degree Parities

We consider the random linear code ensemble defined by generating the $m \times n$ parity check matrix $\mathbf{A}$ with entries *i.i.d.* $\mathrm{Bern}(p)$. We use the connection between random codes and AU hashing established in the previous section to analyze this independent LDPC ensemble (also used in [7]) and show that very sparse parities suffice to yield hash families with useful statistical properties. In particular, we show that when $m = \Theta(n)$ parities are generated, an expected logarithmic parity degree suffices to obtain an (ϵ, q)-AU hash family for any $\epsilon > 2/2^m$ and $q = 2^m$. Such families, for example, satisfy the requirements of Prop. II.1 and can be used in the counting algorithm $\mathcal{A}$ of Section II to obtain interesting guarantees.

Theorem 2. *Let $\mathcal{H}_{(p)} \triangleq \{h : \{0,1\}^n \to \{0,1\}^m\}$ be the family of functions of the form $h(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ with $\mathbf{A} \in \mathbb{F}_2^{m\times n}$ and $\mathbf{b} \in \mathbb{F}_2^m$. We randomly select a function from $\mathcal{H}_{(p)}$ generating the entries of $\mathbf{A}$ independently according to $\mathrm{Bern}(p)$ and selecting $\mathbf{b}$ uniformly at random from $\mathbb{F}_2^m$. If $m = \alpha n$ for some $\alpha \in (0,1)$, and $p = \frac{2}{H^{-1}(\alpha)} \cdot \frac{\log_2(n)}{n}$, then for any $\epsilon > 2/2^m$, $\mathcal{H}_{(p)}$ along with the specified distribution is an $(\epsilon, 2^{\alpha n})$-AU family for all $n \ge N_0$.*

Here, $H^{-1} : [0,1] \to [0, 1/2]$ is the inverse binary entropy function restricting its domain to $[0, 1/2]$ so that the inverse is well defined. The constant N_0 depends on the other parameters of the theorem.

The remainder of this section is devoted to the proof of Theorem 2. In particular, we show that the second condition of Def. 1 (Universality) is satisfied utilizing the results of Sec. III; as discussed in the previous section, the first condition (Uniformity) is trivially satisfied by the presence of the uniformly and independently distributed affine shift $\mathbf{b}$.

Lemma IV.1. *Let $\mathbf{A}$ be an $m \times n$ binary matrix with i.i.d. entries $\mathrm{Bern}(p)$, $p \in (0, 1/2]$. For any $\boldsymbol{\tau} \in \mathbb{F}_2^n$,*

$$\mathbb{P}(\mathbf{A}\boldsymbol{\tau} = \mathbf{0}) = \left(\tfrac{1}{2} + \tfrac{1}{2}(1-2p)^w\right)^m \quad =: r(w),$$

where $w = w(\boldsymbol{\tau})$ is the Hamming weight of $\boldsymbol{\tau}$. Further, $\mathbb{P}(\mathbf{A}\boldsymbol{\tau} = \mathbf{z}) \le \mathbb{P}(\mathbf{A}\boldsymbol{\tau} = \mathbf{0})$, $\forall \mathbf{z} \in \mathbb{F}_2^m$.

Proof. See Appendix A-A in [18]. □

The second part of Lemma IV.1 has an interesting implication: if the hash family of Thm. 2 is (ϵ, q)-WAU, then it is also (ϵ, q)-AU. Hence, it suffices to show that the hash family is WAU with the desired parameters.

Further observe that $r(w)$ is monotonically decreasing: the smaller the weight of a sequence, the higher the probability the latter will be a codeword. This allows us to simplify the notation of Sec. III: for a given parameter q, we define

$$w_\star \triangleq \max\left\{ w \in [n] : \sum_{i=1}^{w} \binom{n}{i} \le q-1 \right\}, \tag{5}$$

and our objective, adapted from Thm. 1, is to show that

$$\epsilon(q-1) - \sum_{w=1}^{w_\star} \binom{n}{w} r(w) - \left(q - 1 - \sum_{w=1}^{w_\star} \binom{n}{w} \right) r(w_\star + 1) \ge 0. \tag{6}$$

for given ϵ and q. Observe that for (6) to hold, it is necessary that $r(w_\star + 1) < \epsilon$. For p as in Thm. 2, this is in fact the case. Rearranging and removing nonnegative terms from the left-hand side (LHS) of (6), one can verify that

$$\text{LHS (6)} \ge \sum_{w=1}^{w_\star} \binom{n}{w} (\epsilon - r(w)). \tag{7}$$

We define

$$w_{\text{bp}} \triangleq \max\{w : r(w) \ge \epsilon\}. \tag{8}$$

Clearly, $w_{\text{bp}} \le w_\star$. Continuing from (7),

$$\begin{aligned} \text{LHS (6)} &= \sum_{w=1}^{w_{\text{bp}}} \binom{n}{w} (\epsilon - r(w)) + \sum_{w=w_{\text{bp}}+1}^{w_\star} \binom{n}{w} (\epsilon - r(w)) \\ &\ge -\sum_{w=1}^{w_{\text{bp}}} \binom{n}{w} r(w) + \binom{n}{w_\star} (\epsilon - r(w_\star)). \end{aligned} \tag{9}$$

Hence, to show that (6) holds, it suffices to show that

$$\sum_{w=1}^{w_{\text{bp}}} \binom{n}{w} r(w) \le \binom{n}{w_\star} (\epsilon - r(w_\star)). \tag{10}$$

We proceed by establishing suitable upper and lower bounds for the left (LHS) and right-hand side (RHS) of (10), respectively, and show that for sufficiently large n, a gap between the two exists.

Lower bound on the RHS of (10)*:* By definition (5), $w_\star$ is the radius of the largest Hamming ball containing less than q points. The volume of the latter cannot be much smaller than q:

Lemma IV.2. *For any $1 \le q \le 2^{n-1}$, and $w_\star$ defined in* (5)*, $\binom{n}{w_\star} \ge q/(n+1)$.*

Proof. (See Appendix A-B in [18].) □

We also derive a lower bound on $w_\star$. To do that, first note that if $q \le 2^{n-1}$, then $w_\star \le \lfloor n/2 - 1 \rfloor$ and in turn $w_\star + 1 \le n/2$. Then, by the entropy bound on the sum of binomials, we have

$$q \le \sum_{w=1}^{w_\star+1} \binom{n}{w} \le 2^{nH(\frac{w_\star+1}{n})}.$$

For $q = 2^{\alpha n}$ as specified in Thm 2, we conclude that $w_\star \ge n \cdot H^{-1}(\alpha) - 1$, or more loosely for $n \ge 4/H^{-1}(\alpha)$:

$$w_\star \ge \tfrac{3}{4} n \cdot H^{-1}(\alpha). \tag{11}$$

Finally, using the fact that $1 + x \le e^x \, \forall x$, for all $w \in [n]$,

$$\begin{aligned} \log_2 r(w) &= -m + m \log_2(1 + (1-2p)^w) \\ &\le -m + m \log_2(1 + \exp(-2pw)). \end{aligned} \tag{12}$$

For p as defined in Thm. 2, taking into account (11), we have $-2pw_\star \le -3\log_2(n)$. Continuing from (12),

$$\begin{aligned} \log_2 r(w_\star) &= -m + m \exp(-2pw_\star) \log_2(e) \\ &\le -m + m \cdot n^{-3} \log_2(e) \le -m + n^{-3+1} \log_2(e). \end{aligned}$$

Let $\overline{\epsilon} = 2^m \cdot \epsilon$. By assumption, $\epsilon \ge 2/2^m$ and in turn $\overline{\epsilon} \ge 2$. By the above, we find that for n sufficiently large (here, $n \ge \ln^{-1/2}(1 + {}^{(\overline{\epsilon}\,-\,1)}/_{2})$),

$$r(w_\star) \le 2^{-m} \cdot (\overline{\epsilon} + 1)/2.$$

Combining this last bound with Lemma IV.2 substituting $q = 2^m$, and taking into account that $\overline{\epsilon} \ge 2$, we finally obtain

$$\binom{n}{w_\star} (\epsilon - r(w_\star)) \ge \frac{2^m}{(n+1)} \cdot \frac{\overline{\epsilon} - 1}{2^m \cdot 2} \ge \frac{1}{2(n+1)}.$$

In turn, we conclude that for sufficiently large n,

$$\text{RHS (10)} \ge \tfrac{1}{3} n^{-1}. \tag{13}$$

Upper bound on the LHS of (10)*:* Define the function $f(w) : [n] \to \mathbb{R}_+$ as $f(w) \triangleq \binom{n}{w} \cdot r(w)$. Then,

$$\text{LHS (10)} \le w_{\text{bp}} \cdot \max_{w \in \{1, \ldots, w_{\text{bp}}\}} f(w). \tag{14}$$

Recall that $r(w)$ is a decreasing function of w, and w_{bp} is by definition (8) the largest w such that $r(w) > \epsilon$, for a given ϵ. Let $w_0 \in \mathbb{R}_+$ be the point where the upper bound (12) on $\log_2 r(w)$ crosses the $\log_2 \epsilon$ threshold. Clearly, $w_{\text{bp}} \le w_0 \le 1/(2p) \ln\left(\frac{n}{\ln \overline{\epsilon}}\right)$. In turn, for p as specified in Thm. 2,

$$w_{\text{bp}} \le \tfrac{1}{2} \cdot H^{-1}(\alpha) \cdot n. \tag{15}$$

for sufficiently large n (here, $n \geq 1/\ln \bar{\epsilon}$).

It remains to bound the maximum of $f(w)$ over the range $w = 1, \ldots, w_{\text{bp}}$. It is convenient to consider three intervals that jointly cover that range and develop a separate bound over each one. Those intervals capture different scaling properties of $f(w)$ with respect to n. In all cases, we use the fact

$$\log_2 f(w) \leq nH(w/n) + \log_2 r(w). \tag{16}$$

Case I: $1 \leq w \leq (2p)^{-1}$. From (12),

$$\begin{aligned}\log_2 r(w) &\leq -m + m\log_2(1 + \exp{(-2pw)}) \\ &\leq -m + m(1 - pw),\end{aligned}$$

where the last inequality follows from the fact $\log_2(1+e^{-x}) \leq 1 - \frac{1}{2}x$ for $0 \leq x \leq 1$, and that $0 \leq 2pw \leq 1$ in this interval. Further, one can show ([18], Lemma B.4) that $\forall w \in [n]$, $H(w/n) \leq \frac{w}{n}\log_2\bigl(\frac{4n}{w}\bigr)$. Hence, continuing from (16),

$$\begin{aligned}\log_2 f(w) &\leq n\frac{w}{n}\log_2\left(\frac{4n}{w}\right) + m - mpw - m \\ &\leq w(\log_2(4n) - mp). \end{aligned} \tag{17}$$

For $m = \alpha n$ and p as specified in Thm. 2, and taking into account the fact that $\forall \alpha \in (0,1)$, $H^{-1}(\alpha) \leq \alpha/2$ (see [18] Lemma B.5), we have

$$\begin{aligned}\log_2(4n) - mp &\leq \log_2(4n) - \alpha n\frac{2}{H^{-1}(\alpha)}\frac{\log_2(n)}{n} \\ &\leq \log_2(4n) - 4\log_2 n,\end{aligned}$$

which is negative for all $n \geq 2$. In turn, for large n, the bound in (17) is maximized for $w = 1$. It then easily follows that for sufficiently large n (here, $n > 16$),

$$\log_2 f(w) < -\frac{5}{2}\log_2 n. \tag{18}$$

Case II: $(2p)^{-1} < w \leq c_0 \cdot n$. The constant c_0 can be arbitrarily selected (affecting other constants in the asymptotic result). For convenience, let $c_0 = H^{-1}(\alpha)/16$. From (12),

$$\begin{aligned}\log_2 r(w) &= -m + m\log_2(1 + \exp{(-2pw)}) \\ &\leq -m + m\log_2(1 + \exp{(-1)}) \leq -0.54\alpha n.\end{aligned}$$

Further, $H(w/n)$ is monotonically increasing in this interval. Hence, utilizing properties of the entropy function (see [18] Lemma B.6), we have $H(w/n) \leq H(c_0) = H\bigl(H^{-1}(\alpha)/16\bigr) \leq \alpha \cdot H(1/32) \leq 0.21\alpha$. Combining the above into (16), we find $\log_2 f(w) \leq -33\alpha n$. In turn, the same bound as in (18) trivially holds for sufficiently large n.

Case III $c_0 \cdot n \leq w \leq w_{\text{bp}}$. Finally, it can be similarly verified that for the earlier choice of c_0, and for sufficiently large n (here, $n \geq 10^4$), in this interval we have $\log_2 r(w) \leq -0.86\alpha n$. Moreover, by (15), we know that $w_{\text{bp}} \leq {}^{1}\!/_{2}H^{-1}(\alpha)n \leq n/4, \forall \alpha$. Hence, the entropy function is increasing in the range $1 \leq w \leq w_{\text{bp}}$ and

$$nH(w/n) \leq nH(w_{\text{bp}}/n) \leq \alpha n H({}^{1}\!/_{4}) < 0.82\alpha n.$$

Combining the above into (16), $\log_2 f(w) \leq -0.04\alpha n$, and in turn, once again, trivially the same bound as in (18) holds for sufficiently large n.

Finally, combining the above three cases, we conclude that $\max_{w \in \{1,\ldots,w_{\text{bp}}\}} f(w) \leq 1/n^{5/2}$, while $w_{\text{bp}} \leq n/2$ trivially from (15). Substituting those bounds in (14), we find

$$\text{LHS } (10) \leq \tfrac{1}{2}n^{-3/2}. \tag{19}$$

Comparing (19) with (13) reveals the existence of a gap for sufficiently large n, which completes the proof of Thm. 2.

ACKNOWLEDGMENT The authors would like to acknowledge support from grants: NSF CCF 1422549, 1344364, 1344179 and an ARO YIP award. We would also like to thank S. Ermon for introducing us to this problem.

REFERENCES

[1] M. Jerrum and A. Sinclair, "The markov chain monte carlo method: An approach to approximate counting and integration," in *Approximation Algorithms for NP-hard Problems* (D. S. Hochbaum, ed.), pp. 482–520, Boston, MA, USA: PWS Publishing Co., 1997.

[2] N. Madras, "Lectures on monte carlo methods, vol. 16 of fields institute monographs," *American Mathematical Society, Providence, RI*, 2002.

[3] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[4] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[5] S. Ermon, C. P. Gomes, A. Sabharwal, and B. Selman, "Optimization with parity constraints: From binary codes to discrete integration," in *Uncertainty in Artificial Intelligence*, p. 202, Citeseer, 2013.

[6] S. Ermon, C. P. Gomes, A. Sabharwal, and B. Selman, "Taming the curse of dimensionality: Discrete integration by hashing and optimization," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, pp. 334–342, 2013.

[7] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman, "Low-density parity constraints for hashing-based discrete integration," in *Proceedings of The 31st International Conference on Machine Learning*, pp. 271–279, 2014.

[8] D. Achlioptas and P. Jiang, "Stochastic integration via errorcorrecting codes," in *Proc. Uncertainty in Artificial Intelligence*, 2015.

[9] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *Information Theory, IEEE Transactions on*, vol. 51, pp. 386–398, Jan 2005.

[10] S. Zhao, S. Chaturapruek, A. Sabharwal, and S. Ermon, "Closing the gap between short and long xors for model counting," in *The 30th Association for the Advancement of Artificial Intelligence (AAAI-16) Conference (to appear)*, 2016.

[11] S. Litsyn and V. Shevelev, "On ensembles of low-density parity-check codes: asymptotic distance distributions," *IEEE Transactions on Information Theory*, vol. 48, no. 4, pp. 887–908, 2002.

[12] S. Litsyn and V. Shevelev, "Distance distributions in ensembles of irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3140–3159, 2003.

[13] R. Ikegaya, K. Kasai, T. Shibuya, and K. Sakaniwa, "Asymptotic weight and stopping set distributions for detailedly represented irregular ldpc code ensembles," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 87, no. 10, pp. 2484–2492, 2004.

[14] D. Burshtein and G. Miller, "Asymptotic enumeration methods for analyzing ldpc codes," *Information Theory, IEEE Transactions on*, vol. 50, no. 6, pp. 1115–1131, 2004.

[15] A. Barg and G. D. Forney Jr, "Random codes: minimum distances and error exponents," *Information Theory, IEEE Transactions on*, vol. 48, no. 9, pp. 2568–2573, 2002.

[16] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[17] V. Pless, W. Huffman, and R. Brualdi, *Handbook of coding theory*. No. v. 2 in Handbook of Coding Theory, Elsevier, 1998.

[18] "http://megasthenis.github.io/repository/sparse_hashing.pdf."

Appendix A
Proofs

A. Proof of Proposition IV.1

Proposition A.2. *Let* $\mathbf{a} \in \{0,1\}^{n\times 1}$ *be a random vector whose entries are i.i.d. Bernoulli(p),* $p \in (0, 1/2]$. *For any given* $\boldsymbol{\sigma} \in \{0,1\}^{n\times 1}$,

$$\mathbb{P}\big(\mathbf{a}^\top \boldsymbol{\sigma} = \mathbf{0}\big) = \frac{1}{2} + \frac{1}{2}(1-2p)^w,$$

where w *is the Hamming weight of* $\boldsymbol{\sigma}$.

Proof. Observe that $\mathbf{a}$ is orthogonal to $\boldsymbol{\sigma}$ if an only if the two vectors have an even number of overlapping non-zero entries. Hence, only the w entries of $\mathbf{a}$ corresponding to the w non-zero entries of $\boldsymbol{\sigma}$ are relevant. We focus on those w entries. Since they are independently distributed, we can assign values successively in an arbitrary order. Let X_i denote the number of non-zero entries after the ith assignment. Then, $\mathbf{a}^\top \boldsymbol{\sigma} = \mathbf{0}$ if and only if X_w is even. Further,

$$\begin{aligned}
&\mathbb{P}(X_w \text{ is even}) \\
&= p \cdot \mathbb{P}(X_{w-1} \text{ is odd}) + (1-p) \cdot \mathbb{P}(X_{w-1} \text{ is even}) \\
&= p \cdot (1 - \mathbb{P}(X_{w-1} \text{ is even})) + (1-p) \cdot \mathbb{P}(X_{w-1} \text{ is even}) \\
&= p + (1-2p) \cdot \mathbb{P}(X_{w-1} \text{ is even}).
\end{aligned}$$

Unfolding the recursion and taking into account that $\mathbb{P}(X_0 \text{ is even}) = 1$,

$$\begin{aligned}
\mathbb{P}(X_w \text{ is even}) &= \sum_{i=0}^{w-1} (1-2p)^i \cdot p + (1-2p)^w \\
&= p \cdot \frac{1-(1-2p)^w}{1-(1-2p)} + (1-2p)^w \\
&= \frac{1-(1-2p)^w}{2} + (1-2p)^w \\
&= \frac{1}{2} + \frac{1}{2}(1-2p)^w,
\end{aligned}$$

which is the desired result. □

The first part follows trivially from Proposition A.2 and the fact that each of the m rows of $\mathbf{A}$ is selected independently, with i.i.d. entries Bernoulli(p). For the second part, note that for $\mathbf{a} \in \{0,1\}^{n\times 1}$ with i.i.d. Bernoulli(p) entries, $p \in (0, 1/2]$, and any given $\mathbf{x} \in \{0,1\}^n$, $\mathbb{P}(\mathbf{a}^\top \boldsymbol{\tau} = 0) \geq 1/2$. In turn,

$$\mathbb{P}(\mathbf{a}^\top \boldsymbol{\tau} = 1) = 1 - \mathbb{P}(\mathbf{a}^\top \boldsymbol{\tau} = 0) \leq \mathbb{P}(\mathbf{a}^\top \boldsymbol{\tau} = 0).$$

Since each row of $\mathbf{A}$ is drawn independently with i.i.d. entries $\mathrm{Bern}(p)$, it follows that for any given $\mathbf{z} \in \{0,1\}^m$,

$$\begin{aligned}
\mathbb{P}(\mathbf{A}\boldsymbol{\tau} = \mathbf{z}) &= \prod_{i=1}^{m} \mathbb{P}(\mathbf{A}_{i,:}\boldsymbol{\tau} = z_i) \\
&\leq \big(\mathbb{P}(\mathbf{a}^\top \boldsymbol{\tau} = 0)\big)^m = \mathbb{P}(\mathbf{A}\boldsymbol{\tau} = \mathbf{0}),
\end{aligned}$$

which is the desired result.

B. Proof of Lemma IV.2

The lemma holds trivially if $|S| \leq n$, in which case $w_\star = 0$. Next, we show it holds for $n+1 \leq |S| \leq 2^{n-1}$. Observe that if $|S| \leq 2^{n-1}$, then $w_\star \leq \lfloor n/2 \rfloor$. Otherwise,

$$\sum_{w=1}^{w_\star} \binom{n}{w} > \sum_{w=1}^{\lfloor n/2 \rfloor} \binom{n}{w} \geq 2^{n-1}.$$

By the definition of $w_\star$,

$$\begin{aligned}
|S| &\leq \sum_{w=1}^{w_\star} \binom{n}{w} + \binom{n}{w_\star + 1} \\
&= \sum_{w=1}^{w_\star - 1} \binom{n}{w} + \binom{n}{w_\star} + \binom{n}{w_\star + 1} \\
&\leq \frac{n+1}{2}\binom{n}{w_\star} + \binom{n}{w_\star} + \frac{n - w_\star}{w_\star + 1}\binom{n}{w_\star} \\
&\leq \frac{n+1}{2}\binom{n}{w_\star} + \binom{n}{w_\star} + \frac{n-1}{1+1}\binom{n}{w_\star} \\
&= (n+1)\binom{n}{w_\star}, \qquad (20)
\end{aligned}$$

where the first inequality follows from Cor. 3 and Lemma B.7, and the second from the fact that $w_\star \geq 1$ for $|S| \geq n+1$ and that $(n-k)/(k+1)$ is decreasing in $k \in [n]$.

Appendix B
Auxiliary Lemmas

Lemma B.3. *For* $0 \leq x \leq 1$,

$$\log_2(1 + e^{-x}) \leq 1 - \tfrac{1}{2}x.$$

Proof. Let $g(x) = 1 - x/2 - \log_2(1 + e^{-x})$. The second derivative of $g(\cdot)$ is

$$g^{(2)}(x) = -\frac{e^{-x}}{\ln(2)(1 + e^{-x})} \leq 0,$$

i.e., $g(x)$ is concave. Concavity along with the fact that $g(0) = 0$ and $g(1) = 1/2 - \log_2(1 + e^{-1}) \geq 0$ implies that $g(x) \geq 0$ in the specified interval. □

Lemma B.4.

$$\forall p \in [0,\, 1], \quad p\log_2(1/p) \leq H(p) \leq p\log_2(4/p).$$

Proof. The lower bound follows trivially from the definition of $H(p)$ and the fact that $-(1-p)\log(1-p) \geq 0$ for $\forall p \in [0,\, 1]$. For the upper bound, let $f(p) = -(1-p) \cdot \log_2(1-p) - 2p$. Note that $f(0) = 0$ and

$$f^{(1)}(p) = \log_2(1-p) - 1 \leq 0, \forall p \in [0,\, 1],$$

which implies that $f(p) \leq 0$, $\forall p \in [0,\, 1]$. Equivalently, $-(1-p) \cdot \log_2(1-p) \leq 2p$ and in turn, $H(p) \leq -p\log_2(p) + 2p = p\log_2(4/p)$. □

Lemma B.5. $\forall \alpha \in [0,\, 1], \quad H^{-1}(\alpha) \leq \tfrac{1}{2}\alpha.$

Proof. We restrict the domain of the entropy function $H(p)$ to $[0,\, 1/2]$ where the function is strictly monotonic and the inverse entropy $H^{-1} : [0,1] \to [0, 1/2]$ is well defined. Observe that

$$\forall p \in [0,\, 1/2], \quad H(p) \geq 2p.$$

To see that note that any $p \in [0,\ {}^{1}\!/_{2}]$ can be written as $p = \lambda/2$ for some $\lambda \in [0,1]$ and by the concavity of $H(\cdot)$,

$$\begin{aligned} H(p) &= H((1-\lambda)\cdot 0 + \lambda/2) \\ &\geq (1-\lambda)\cdot H(0) + \lambda \cdot H({}^{1}\!/_{2}) = \lambda = 2p. \end{aligned}$$

Further, $L(p) = 2p$ is strictly monotone everywhere and its inverse $L^{-1}(\alpha) = \alpha/2$ is also well defined. Since $H(p) \geq L(p)\ \forall p \in [0, {}^{1}\!/_{2}]$, it follows that $H^{-1}(\alpha) \leq L^{-1}(\alpha) = \alpha/2$ for all $\alpha \in [0,1]$, which is the desired result. □

Corollary 1. *For $k \leq n/2$,*

$$\frac{1}{\sqrt{2n}} \cdot 2^{nH(\frac{k}{n})} \leq \sum_{i=0}^{k} \binom{n}{i}.$$

Proof. Let $p = k/n$. Then $p \in [0, 1/2]$. The lower bound is equal to $(8n \cdot p \cdot (1-p))^{-1/2}$. But $p \cdot (1-p) \leq 1/4$, with equality achieved for $p = 1/2$. Hence,

$$(8n \cdot p \cdot (1-p))^{-1/2} \geq (2n)^{-1/2}.$$

□

Lemma B.6. *For any constant $c \in (0,1]$,*

$$H(c \cdot p) \leq H({}^{c}\!/_{2}) \cdot H(p), \quad \forall p \in (0, {}^{1}\!/_{2}].$$

Proof. Let $f_c(p) = H(c \cdot p)/H(p)$. Then,

$$f_c'(p) = [H'(c\cdot p)\cdot H(p) - H(c \cdot p) \cdot H'(p)]/H^2(p).$$

Recall that $H(x)$ is increasing and concave function in $x \in [0, {}^{1}\!/_{2}]$, *i.e.*, $H'(x) \geq 0$ and $H''(x) \leq 0$ for $x \in [0, {}^{1}\!/_{2}]$. It follows that $H(p) \geq H(c\cdot p) \geq 0$, and $H'(c\cdot p) \geq H'(p) \geq 0$. Hence, $H'(c \cdot p)\cdot H(p) \geq H'(p) \cdot H(c \cdot p),$. In turn, $f_c(p)$ is increasing in $p \in (0, {}^{1}\!/_{2}]$ and achieves its maximum value at $p = {}^{1}\!/_{2}$. In other words, $\forall p \in (0, {}^{1}\!/_{2}]$,

$$f_c(p) \leq f_c({}^{1}\!/_{2}) = H({}^{c}\!/_{2})/H({}^{1}\!/_{2}) = H({}^{c}\!/_{2}),$$

which is the desired result. □

Corollary 2. *Let $k = \alpha \cdot n \in [\lfloor n/2 \rfloor]$ for some $\alpha \in (0, 1/2]$, and $\ell \leq \lfloor \beta \cdot k \rfloor$ for some $\beta \in (0,1]$. Then,*

$$\binom{n}{\ell} \leq \binom{n}{k} \cdot 2^{-nH(\alpha)(1-H(\beta/2)) + \log_2(n+1)}.$$

Proof. We have

$$\binom{n}{\ell} \leq 2^{nH({}^{\ell}\!/_{n})} \leq 2^{nH(\beta\alpha)} \leq 2^{nH(\alpha)\cdot H(\beta/2)}.$$

The second inequality follows from the fact that $\ell/n \leq \beta k/n \leq \beta\alpha \leq 1/2$ and that $H(\cdot)$ is increasing in $(0, {}^{1}\!/_{2}]$, while the last from Lemma B.6. Further,

$$\binom{n}{k} \geq \frac{1}{n+1} 2^{nH(\alpha)} = 2^{nH(\alpha) - \log_2(n+1)}.$$

The desired result is obtained combining the two inequalities. □

Lemma B.7. *For any $k \in [n]$,*

$$\binom{n}{k+1} = \frac{n-k}{k+1}\binom{n}{k}.$$

Proof.

$$\begin{aligned} \binom{n}{k+1} &= \frac{n!}{(k+1)!(n-k-1)!} \\ &= \frac{n!(n-k)}{k!(k+1)(n-k)!} = \frac{n-k}{k+1}\binom{n}{k}. \end{aligned}$$

□

Lemma B.8. *For any $n \in \mathbb{N}_+$,*

$$\binom{n}{\lfloor n/2 \rfloor} \geq 2^n/(n+1).$$

Proof. The lemma follows from the fact that there are $n+1$ binomial coefficients, whose sum is equal to 2^n, among which $\binom{n}{\lfloor n/2 \rfloor}$ is the largest. □

Lemma B.9. *For any $k \leq \lfloor n/2 \rfloor$,*

$$\sum_{i=0}^{k-1}\binom{n}{i} < \frac{2^{n-1}}{\binom{n}{\lfloor n/2 \rfloor}} \cdot \binom{n}{k}.$$

Proof. The Lemma is Lemma 3.8.2 in [19]. There it is shown for even n, but the arguments of the proof go through in general. We repeat the proof for convenience.

Let $t = \lfloor n/2 \rfloor - k$, and define

$$A \triangleq \sum_{i=1}^{k-1}\binom{n}{i} = \sum_{i=1}^{\lfloor n/2 \rfloor - t - 1}\binom{n}{i},$$

and

$$B \triangleq \sum_{i=k}^{\lfloor n/2 \rfloor - 1}\binom{n}{i} = \sum_{i=\lfloor n/2 \rfloor - t}^{\lfloor n/2 \rfloor - 1}\binom{n}{i}.$$

We want to compare the quantities A and B. First, note that

$$A + B < 2^{n-1}. \tag{21}$$

Further, let

$$c \triangleq \binom{n}{k} \Big/ \binom{n}{\lfloor n/2 \rfloor}.$$

Then

$$\binom{n}{\lfloor n/2 \rfloor - t} = c \cdot \binom{n}{\lfloor n/2 \rfloor}.$$

Further,

$$\begin{aligned} \binom{n}{\lfloor n/2 \rfloor - t - 1} &= \frac{\lfloor n/2 \rfloor - t}{n - \lfloor n/2 \rfloor + t + 1}\binom{n}{\lfloor n/2 \rfloor - t} \\ &< c \cdot \frac{\lfloor n/2 \rfloor - t}{n - \lfloor n/2 \rfloor + t + 1}\binom{n}{\lfloor n/2 \rfloor} \\ &\leq c \cdot \frac{\lfloor n/2 \rfloor}{n - \lfloor n/2 \rfloor + 1}\binom{n}{\lfloor n/2 \rfloor} \end{aligned}$$

$$= c \cdot \binom{n}{\lfloor n/2 \rfloor - 1}.$$

Repeating the same argument, for any $j \geq 0$

$$\binom{n}{\lfloor n/2 \rfloor - t - j} < c \cdot \binom{n}{\lfloor n/2 \rfloor - j},$$

which in turn implies that the sum of t consequtive binomials is at most c times the sum of the next t consecutive binomials, as long as they all lie on the left half of Pascal's triangle. Starting from $\binom{n}{\lfloor n/2-1 \rfloor}$ and moving backwards, the first t terms sum to B, while the next t sum to less than cB, the next t to less than $c^2 B$, etc. Allowing the sum to go to infinity and taking into account that $c < 1$,

$$A < \sum_{\ell=1}^{\infty} c^{\ell} B = B\left(\frac{1}{1-c} - 1\right) = B\frac{c}{1-c}$$

and in turn $(1-c)/c \cdot A < B$. Combining with (21),

$$c \cdot 2^{n-1} > c(A+B) > c(A + \frac{1-c}{c}A) = A,$$

which is the desired result. □

Corollary 3. *For $k \leq \lfloor n/2 \rfloor$,*

$$\sum_{i=0}^{k-1} \binom{n}{i} \leq \frac{n+1}{2} \binom{n}{k}.$$

Proof. It follows combining Lemmata B.9 and B.8. □

Appendix C
(Almost) right-regular ensemble

Consider a random $m \times n$ matrix $\mathbf{A}$ generated as follows: the ith row $\mathbf{a}_i \in \mathbb{R}^{1 \times n}$, $i = 1, \ldots, m$ (*i.e.*, the ith parity check) is zero everywhere except (at most) d entries which are selected uniformly at random and independently from $[n]$ with replacement. Replacement implies that the parity check may include the same variable twice (*i.e.* multi-edges are allowed in the Tanner graph of the parity check matrix). If the same variable is selected twice by the above process (or more generally an even number of times for that matter), then the particular variable no longer affects the parity check. In other words, the effective degree of each parity check is at most d, and is expect it to be asymptotically equal to d assuming that $d = o(n)$.

One way to formalize the above construction of $\mathbf{A}$ is the following. Each row $\mathbf{a} \in \{0,1\}^n$ is generated independently as the sum of d vectors $\mathbf{z}_j \in \mathbb{F}_2^n$, $j = 1, \ldots, d$,

$$\mathbf{a} = \sum_{j=1}^{d} \mathbf{z}_j,$$

where each $\mathbf{z}_j$ is chosen independently and uniformly at random from $\{\mathbf{e}_i\}_{i=1}^n$. Here, $\mathbf{e}_i$ denotes the ith vector of the standard basis of $\mathbb{F}_2^n$.

Proposition C.3. *Let $\mathbf{a} \in \mathbb{F}_2^{n \times 1}$ be a random vector constructed as*

$$\mathbf{a} = \sum_{j=1}^{d} \mathbf{z}_j,$$

where each $\mathbf{z}_i$ is selected uniformly at random and independently from the set $\{\mathbf{e}_i\}_{i=1}^n$ of the standard basis vectors (the summation is over $\mathbb{F}_2$). For any given $\boldsymbol{\sigma} \in \mathbb{F}_2^{n \times 1}$,

$$\mathbb{P}\big(\mathbf{a}^\top \boldsymbol{\sigma} = \mathbf{0}\big) = \frac{1}{2} + \frac{1}{2} \cdot \Big(1 - 2 \cdot \frac{w}{n}\Big)^d,$$

where w is the Hamming weight of $\boldsymbol{\sigma}$.

Proof. The inner product is equal to zero if an only if the two vectors have an even number of overlapping non-zero entries. Let $\mathcal{S}$ be the support set of $\boldsymbol{\sigma}$, *i.e.*, the set of indices of its nonzero entries. Then, $|\mathcal{S}| = w$.

Note that $\mathbb{P}(\mathbf{z}_j^\top \boldsymbol{\sigma} = 1) = \mathbb{P}(\text{supp}(\mathbf{z}_j) \cap \mathcal{S} = 1) = w/n$, as $\text{supp}(\mathbf{z}_j)$ contains a single index uniformly distributed over $[n]$. Let $\mathbf{X}_k = \sum_{j=1}^k \mathbf{z}_j^\top \boldsymbol{\sigma}$.Then, taking into account the indepence among the selection of $\mathbf{z}_j$s,

$$\begin{aligned}
&\mathbb{P}\big(\mathbf{a}^\top \boldsymbol{\sigma} = 0\big) = \mathbb{P}(X_d = 0) \\
&= \mathbb{P}(X_{d-1} = 0)\mathbb{P}\big(\mathbf{z}_d^\top \boldsymbol{\sigma} = 0\big) + \mathbb{P}(X_{d-1} = 1)\mathbb{P}\big(\mathbf{z}_d^\top \boldsymbol{\sigma} = 1\big) \\
&= \Big(1 - \frac{w}{n}\Big) \cdot \mathbb{P}(X_{d-1} = 0) + \frac{w}{n} \cdot \mathbb{P}(X_{d-1} = 1) \\
&= \Big(1 - \frac{w}{n}\Big) \cdot \mathbb{P}(X_{d-1} = 0) + \frac{w}{n} \cdot (1 - \mathbb{P}(X_{d-1} = 0)) \\
&= \frac{w}{n} + \Big(1 - 2 \cdot \frac{w}{n}\Big) \cdot \mathbb{P}(X_{d-1} = 0).
\end{aligned}$$

Unfolding the recursion,

$$\begin{aligned}
&\mathbb{P}\big(\mathbf{a}^\top \boldsymbol{\sigma} = 0\big) \\
&= \frac{w}{n} + \sum_{j=1}^{d-2} \Big(1 - 2 \cdot \frac{w}{n}\Big)^j \cdot \frac{w}{n} + \Big(1 - 2 \cdot \frac{w}{n}\Big)^{d-1} \cdot \mathbb{P}(X_1 = 0) \\
&= \frac{w}{n} + \sum_{j=1}^{d-2} \Big(1 - 2 \cdot \frac{w}{n}\Big)^j \cdot \frac{w}{n} + \Big(1 - 2 \cdot \frac{w}{n}\Big)^{d-1} \cdot \Big(1 - \frac{w}{n}\Big) \\
&= \sum_{j=0}^{d-1} \Big(1 - 2 \cdot \frac{w}{n}\Big)^j \cdot \frac{w}{n} + \Big(1 - 2 \cdot \frac{w}{n}\Big)^d \\
&= \frac{w}{n} \cdot \frac{1 - (1 - 2 \cdot w/n)^d}{1 - (1 - 2 \cdot w/n)} + \Big(1 - 2 \cdot \frac{w}{n}\Big)^d \\
&= \frac{1}{2} + \frac{1}{2} \cdot \Big(1 - 2 \cdot \frac{w}{n}\Big)^d,
\end{aligned}$$

which is the desired result. □

Remark C.1. *For* $\mathbf{A}$ *drawn from the (almost) right regular ensemble,* $\mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{0})$ *is a decreasing function of w for w in* $[0,\ n/2]$.

Proposition C.4. *Let $\mathbf{A}$ be an $m \times n$ binary matrix whose rows are generated independently as described by the (almost) right regular ensemble. For any given $\boldsymbol{\sigma} \in \{0,1\}^n$,*

$$\mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{0}) = \left(\frac{1}{2} + \frac{1}{2}\Big(1 - 2 \cdot \frac{w}{n}\Big)^d\right)^m,$$

where w is the Hamming weight of $\boldsymbol{\sigma}$. Further, if $w \leq n/2$, then $\forall \mathbf{y} \in \{0,1\}^m$,

$$\mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{y}) \leq \mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{0}),$$

where the randomness is over the selection of the $m \times n$ matrix $\mathbf{A}$.

Proof. The first part follows trivially from Proposition C.3 and the fact that each of the m rows of $\mathbf{A}$ is selected independently. For the second part, note that for $\mathbf{a}$ generated as described above and any given $\boldsymbol{\sigma} \in \{0,1\}^n$, $\mathbb{P}(\mathbf{a}^\top \boldsymbol{\sigma} = 0) \geq 1/2$. In turn,

$$\mathbb{P}(\mathbf{a}^\top \boldsymbol{\sigma} = 1) = 1 - \mathbb{P}(\mathbf{a}^\top \boldsymbol{\sigma} = 0) \leq \mathbb{P}(\mathbf{a}^\top \boldsymbol{\sigma} = 0).$$

Since each row of $\mathbf{A}$ is drawn independently with i.i.d. entries Bernoulli(p), it follows that for any given $\mathbf{y} \in \{0,1\}^m$,

$$\begin{aligned} \mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{y}) &= \prod_{i=1}^{m} \mathbb{P}(\mathbf{A}_{i,:}\boldsymbol{\sigma} = y_i) \\ &\leq \left(\mathbb{P}(\mathbf{a}^\top \boldsymbol{\sigma} = 0)\right)^m = \mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{0}), \end{aligned} \tag{22}$$

which is the desired result. □

a) Observation::

$$\begin{aligned} \log_2 \mathbb{P}(\mathbf{A}\boldsymbol{\sigma} = \mathbf{0}) &= m \cdot \log_2 \left(1 + \left(1 - 2\frac{w}{n}\right)^d\right) - m \\ &\leq m \cdot \log_2 \left(1 + \exp\left(-2\frac{dw}{n}\right)\right) - m, \end{aligned}$$

where in the last inequality we have used the fact that $1 - x \leq e^{-x}$, $\forall x$.

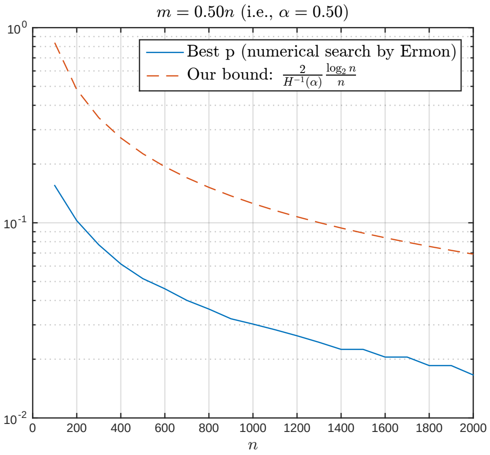

Fig. 1. The figure depicts the scaling of the parameter p, *i.e.*, the probability that a parity constraint includes each one of the n variables, with respect to n (for $m = \alpha n$, $\alpha = 1/2$). The solid curve depicts the optimal value of p as computed via a numerical search approach as in [7] and our theoretical upper bound. The dashed one depicts our theoretical upper bound on p.

Appendix References

[19] L. Lovász, J. Pelikán, and K. Vesztergombi. *Discrete Mathematics: Elementary and Beyond*. Discrete Mathematics: Elementary and Beyond. Springer, 2003.