# Sticky Brownian Rounding and its Applications to Constraint Satisfaction Problems

Sepehr Abbasi-Zadeh [*] Nikhil Bansal[†] Guru Guruganesh [‡] Aleksandar Nikolov [§]
Roy Schwartz [¶] Mohit Singh [‖]

**Abstract**

Semidefinite programming is a powerful tool in the design and analysis of approximation algorithms for combinatorial optimization problems. In particular, the random hyperplane rounding method of Goemans and Williamson [23] has been extensively studied for more than two decades, resulting in various extensions to the original technique and beautiful algorithms for a wide range of applications. Despite the fact that this approach yields tight approximation guarantees for some problems, *e.g.*, Max-Cut, for many others, *e.g.*, Max-SAT and Max-DiCut, the tight approximation ratio is still unknown. One of the main reasons for this is the fact that very few techniques for rounding semi-definite relaxations are known.

In this work, we present a new general and simple method for rounding semi-definite programs, based on Brownian motion. Our approach is inspired by recent results in algorithmic discrepancy theory. We develop and present tools for analyzing our new rounding algorithms, utilizing mathematical machinery from the theory of Brownian motion, complex analysis, and partial differential equations. Focusing on constraint satisfaction problems, we apply our method to several classical problems, including Max-Cut, Max-2SAT, and Max-DiCut, and derive new algorithms that are competitive with the best known results. To illustrate the versatility and general applicability of our approach, we give new approximation algorithms for the Max-Cut problem with side constraints that crucially utilizes measure concentration results for the Sticky Brownian Motion, a feature missing from hyperplane rounding and its generalizations.[1]

## 1 Introduction

Semi-definite programming (SDP) is one of the most powerful tools in the design of approximation algorithms for combinatorial optimization problems. Semi-definite programs can be viewed as relaxed quadratic programs whose variables are allowed to be vectors instead of scalars and scalar multiplication is replaced by inner products between the vectors. The prominent approach when designing SDP based approximation algorithms is *rounding*: (1) an SDP relaxation is formulated for the given problem, (2) the SDP relaxation is solved, and lastly (3) the fractional solution for the SDP relaxation is transformed into a feasible integral solution to the original problem, hence the term *rounding*.

In their seminal work, Goemans and Williamson [23] presented an elegant and remarkably simple rounding method for SDPs: a uniformly random hyperplane (through the origin) is chosen, and then each variable, which is a vector, is assigned to the side of the hyperplane it belongs to. This (binary) assignment is used to round the vectors and output an integral solution. For example, when considering Max-Cut, each side of the hyperplane corresponds to a different side of the cut. Using the random hyperplane rounding, [23] gave the first non-trivial approximation guarantees for fundamental problems such as Max-Cut, Max-2SAT, and Max-DiCut. Perhaps the most celebrated result of [23] is the 0.878 approximation for Max-Cut, which is known to be tight [26, 30] assuming Khot's Unique Games Conjecture [25]. Since then, the random hyperplane method has inspired, for more than two decades now, a large body of research, both in approximation algorithms and in hardness of approximation. In particular, many extensions and generalizations of the random hyperplane rounding method have been proposed and applied to a wide range of applications , *e.g.*, Max-DiCut and Max-2SAT [19, 27], Max-SAT [5, 9], Max-Bisection [8, 34], Max-Agreement in correlation clustering [16], the Cut-Norm of a matrix [3].

[*] University of Toronto. E-mail: `sabbasizadeh@gmail.com`.

[†] TU Eindhoven, and Centrum Wiskunde & Informatica. E-mail: `bansal@gmail.com`.

[‡] Google Research. Email: `gurug@google.com`.

[§] University of Toronto. Email:`anikolov@cs.toronto.edu`.

[¶] Technion. Email:`schwartz@cs.technion.ac.il`.

[‖] Georgia Institute of Technology. Email:`mohitsinghr@gmail.com`.

[1] Full version of the paper is available at: https://arxiv.org/abs/1812.07769.

Despite this success and the significant work on variants and extensions of the random hyperplane method, the best possible approximation ratios for many fundamental problems still remain elusive. Several such examples include MAX-SAT, MAX-BISECTION, MAX-2CSP, and MAX-DICUT. Perhaps the most crucial reason for the above failure is the fact that besides the random hyperplane method and its variants, very few methods for rounding SDPs are known.

A sequence of papers by Austrin [6], Raghavendra [32], Raghavendra and Steurer [33] has shown that SDP rounding algorithms that are based on the random hyperplane method and its extensions nearly match the Unique Games hardness of any MAX-CSP, as well as the integrality gap of a natural family of SDP relaxations. However, the universal rounding proposed by Raghavendra and Steurer is impractical, as it involves a brute-force search on a large constant-sized instance of the problem. Moreover, their methods only allow computing an $\varepsilon$ additive approximation to the approximation ratio in time double-exponential in $1/\varepsilon$.

### 1.1 Our Results and Techniques.

Our main contributions are (1) to propose a new SDP rounding technique that is based on diffusion processes, and, in particular, on Brownian motion; (2) to develop the needed tools for analyzing our new SDP rounding technique by deploying a variety of mathematical techniques from probability theory, complex analysis and partial differential equations (PDEs); (3) to show that this rounding technique has useful concentration of measure properties, not present in random hyperplane based techniques, that can be used to obtain new approximation algorithms for a version of the MAX-CUT problem with multiple global side constraints.

Our method is inspired by the recent success of Brownian motion based algorithms for constructive discrepancy minimization, where it was used to give the first constructive proofs of some of the most powerful results in discrepancy theory [10, 11, 12, 28]. The basic idea is to use the solution to the semi-definite program to define the starting point and the covariance matrix of the diffusion process, and let the process evolve until it reaches an integral solution. As the process is forced to stay inside the cube $[-1,1]^n$ (for MAX-CUT) or $[0,1]^n$ (for MAX-2SAT and other problems), and to stick to any face it reaches, we call the most basic version of our algorithm (without any enhancements) the *Sticky Brownian Motion* rounding. The algorithm is defined more formally in Section 1.2.1.

**Sticky Brownian Motion.** Using the tools we introduce, we show that this algorithm is already competitive with the state of the art results for MAX-CUT, MAX-2SAT, and MAX-DICUT.

THEOREM 1.1. *The basic Brownian rounding achieves an approximation ration of* 0.861 *for the* MAX-CUT *problem. Moreover, when the* MAX-CUT *instance has value* $1-\varepsilon$, *Sticky Brownian Motion achieves value* $1-\Omega(\sqrt{\varepsilon})$.

In particular, using complex analysis and evaluating various elliptic integrals, we show that the separation probability for any two unit vectors $u$ and $v$ separated by an angle $\theta$, is given by a certain hypergeometric function of $\theta$ (see Theorem 2.1 for details). This precise characterization of the separation probability also proves that the Sticky Brownian Motion rounding is different from the random hyperplane rounding. The overview of the analysis is in Section 1.2.2 and Section 2 has the details.

We can also analytically show the following upper bound for MAX-2SAT.

THEOREM 1.2. *The Sticky Brownian Motion rounding achieves approximation ratio of at least* 0.8749 *for* MAX-2SAT.

While the complex analysis methods also give exact results for MAX-2SAT, the explicit expressions are much harder to obtain as one has to consider all possible starting points for the diffusion process, while in the MAX-CUT case the process always starts at the origin. Because of this, in order to prove Theorem 1.2 we introduce another method of analysis based on partial differential equations (PDEs), and the maximum principle, which allows us to prove analytic bounds on PDE solutions. Moreover, numerically solving the PDEs suggests the bound 0.921. The overview and details of the MAX-2SAT analysis are, respectively, in Sections 1.2.3 and 3. Section 5.1 has details about numerical calculations for various problems.

For comparison, the best known approximation ratio for MAX-CUT is the Goemans-Williamson constant $\alpha_{GW} \approx 0.878$, and the best known approximation ratio for MAX-2SAT is 0.94016 [27]. The result for MAX-CUT instances of value $1-\varepsilon$ is optimal up to constants [26], assuming the Unique Games Conjecture.

We emphasize that our results above are achieved with a single algorithm "out of the box", without any additional engineering. While the analysis uses sophisticated mathematical tools, the algorithm itself is simple, efficient, and straightforward to implement.

**Extensions.** Next, we consider two different modifications of Sticky Brownian Motion that allow us to improve the approximation guarantees above, and show the flexibility of diffusion based rounding algorithms.

| Algorithm | Max-Cut | Max-2SAT | Max-DiCut $^\star$ |
|---|---|---|---|
| Brownian | 0.861 | 0.921 | 0.79 |
| Slowdown Brownian | $0.878^\dagger$ | 0.929 | 0.81 |

Table 1: Approximation ratios for Sticky Brownian Motion rounding and Sticky Brownian Motion with Slowdown. † indicates that for Max-Cut, the approximation for the slowed down walk differs from the GW bound only in the fourth decimal. For Max-DiCut, the $^\star$ indicates that we only consider the $n+1$-dimensional walk.

The first one is to smoothly slow down the process depending on how far it is from the boundaries of the cube. As a proof of concept, we show, numerically, that a simple modification of this kind matches the Goemans-Williamson approximation of Max-Cut up to the first three digits after the decimal point. We also obtain significant improvements for other problems over the vanilla method.

Second, we propose a variant of Sticky Brownian Motion running in $n+1$ dimensions rather than $n$ dimensions, and we analyze it for the Max-DiCut problem. The extra dimension is used to determine whether the nodes labeled $+1$ or those labeled $-1$ are put on the left side of the cut. We show that this modification achieves an approximation ratio of 0.79 for Max-DiCut. Slowing down the process further improves this approximation to 0.81. We give a summary of the obtained results[2] in Table 1. An overview and details of the extensions are given, respectively, in Sections 1.2.4 and 5.1.

**Recent Progress.** Very recently, in a beautiful result, Eldan and Naor [18] describe a slowdown process that exactly achieves the Goemans-Williamson (GW) bound of 0.878 for Max-Cut, answering an open question posed in an earlier version of this paper. This shows that our rounding techniques are at least as powerful as the classical random hyperplane rounding, and are potentially more general and flexible.

In general, given the dearth of techniques for rounding semidefinite programs, we expect that rounding methods based on diffusion processes, together with the analysis techniques introduced in this paper, will find broader use, and, perhaps lead to improved results for Max-CSP problems.

**Applications.** To further illustrate the versatility and general applicability of our approach, we consider the Max-Cut with Side Constraints problem, abbreviated Max-Cut-SC, a generalization of the Max-Bisection problem which allows for multiple global constraints. In an instance of the Max-Cut-SC problem, we are given an $n$-vertex graph $G = (V, E)$, a collection $\mathcal{F} = \{F_1, \ldots, F_k\}$ of subsets of $V$, and cardinality bounds $b_1, \ldots, b_k \in \mathbb{N}$. The goal is to find a subset $S \subset V$ that maximizes the weight $|\delta(S)|$ of edges crossing the cut $(S, V \setminus S)$, subject to having $|S \cap F_i| = b_i$ for all $i \in [k]$.

Since even checking whether there is a feasible solution is NP-hard [17], we aim for bi-criteria approximation algorithms.[3] We give the following result for the problem, using the Sticky Brownian Motion as a building tool.

Theorem 1.3. *There exists an $O(n^{\mathrm{poly}(\log(k)/\varepsilon)})$-time algorithm that on input a satisfiable instance $G = (V, E)$, $\mathcal{F}$, and $b_1, \ldots, b_k$, as defined above, outputs a $(0.843 - \varepsilon, \varepsilon)$-approximation with high probability.*

In the presence of a single side constraint, the problem is closely related to the Max-Bisection problem [8, 34], and, more generally to Max-Cut with a cardinality constraint. While our methods use the stronger semi-definite programs considered in [34] and [8], the main new technical ingredient is showing that the Sticky Brownian Motion possesses concentration of measure properties that allow us to approximately satisfy multiple constraints. By contrast, the hyperplane rounding and its generalizations that have been applied previously to the Max-Cut and Max-Bisection problems do not seem to allow for such strong concentration bounds. For this reason, the rounding and analysis used in [34] only give an $O(n^{\mathrm{poly}(k/\varepsilon)})$ time algorithm for the Max-Cut-SC problem, which is trivial for $k = \Omega(n)$, whereas our algorithm has non-trivial quasi-polynomial running time even in this regime. We expect that this concentration of measure property will find further applications, in particular to constraint satisfaction problems with global constraints.

**Remark.** We can achieve better results using Sticky Brownian Motion with slowdown. In particular, in time $O(n^{\mathrm{poly}(\log(k)/\varepsilon)})$ we can get a $(0.858 - \varepsilon, \varepsilon)$-approximation with high probability for any satisfiable instance. However, we focus on the basic Sticky Brownian Motion algorithm to simplify exposition. Note that due to the recent work by Austrin and Stanković [7], we know that adding even a single global cardinality constraint to the Max-Cut problem makes it harder to approximate. In particular, they show that subject to a single side constraint, Max-Cut is Unique Games-hard

[2] Our numerical results are not obtained via simulating the random algorithm but solving a discretized version of a PDE that analyzes the performance of the algorithm. Error analysis of such a discretization can allow us to prove the correctness of these bounds within a reasonable accuracy.

[3] We say that a set $S \subset V$ is an $(\alpha, \varepsilon)$-approximation if $\big||S \cap F_i| - b_i\big| \le \varepsilon n$ for all $i \in [k]$, and $|\delta(S)| \ge \alpha \cdot |\delta(T)|$ for all $T \subset V$ such that $|T \cap F_i| = b_i$ for all $i \in [k]$.

to approximate within a factor of approximately 0.858. Thus, assuming the Unique Games conjecture, our approximation factor for the MAX-CUT-SC problem is optimal up to small numerical errors. (We emphasize the possibility of numerical errors as both our result, and the hardness result in [7] are based on numerical calculations.)

## 1.2 Overview

### 1.2.1 The Sticky Brownian Motion Algorithm.

Let us describe our basic algorithm in some detail. Recall that the Goemans-Williamson SDP for MAX-CUT is equivalent to the following vector program: given a graph $G = (V, E)$, we write

$$\begin{aligned} \max \quad & \sum_{(i,j)\in E} \frac{1 - \mathbf{w}_i \cdot \mathbf{w}_j}{2} \\ s.t. \quad & \mathbf{w}_i \cdot \mathbf{w}_i = 1 \qquad\qquad \forall i \in V \end{aligned}$$

where the variables $\mathbf{w}_i$ range over $n$ dimensional real vectors ($n = |V|$). The Sticky Brownian Motion rounding algorithm we propose maintains a sequence of random fractional solutions $\mathbf{X}_0, \ldots, \mathbf{X}_T$ such that $\mathbf{X}_0 = \mathbf{0}$ and $\mathbf{X}_T \in \{-1, +1\}^n$ is integral. Here, a vertex of the hypercube $\{-1, +1\}^n$ is naturally identified with a cut, with vertices assigned $+1$ forming one side of the cut, and the ones assigned $-1$ forming the other side.

Let $A_t$ be the random set of coordinates of $\mathbf{X}_{t-1}$ which are *not* equal to $-1$ or $+1$; we call these coordinates active. At each time step $t = 1, \ldots, T$, the algorithm picks $\mathbf{\Delta X}_t$ sampled from the Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{W}_t$, where $(\mathbf{W}_t)_{ij} = \mathbf{w}_i \cdot \mathbf{w}_j$ if $i, j \in A_t$, and $(\mathbf{W}_t)_{ij} = 0$ otherwise. The algorithm then takes a small step in the direction of $\mathbf{\Delta X}_t$, i.e. sets $\mathbf{X}_t = \mathbf{X}_{t-1} + \gamma \mathbf{\Delta X}_t$ for some small real number $\gamma$. If the $i$-th coordinate of $\mathbf{X}_t$ is very close to $-1$ or $+1$ for some $i$, then it is rounded to either $-1$ or $+1$, whichever is closer. The parameters $\gamma$ and $T$ are chosen so that the fractional solutions $\mathbf{X}_t$ never leave the cube $[-1, 1]^n$, and so that the final solution $\mathbf{X}_T$ is integral with high probability. As $\gamma$ goes to 0, the trajectory of the $i$-th coordinate of $\mathbf{X}_t$ closely approximates a Brownian motion started at 0, and stopped when it hits one of the boundary values $\{-1, +1\}$. Importantly, the trajectories of different coordinates are correlated according to the SDP solution. A precise definition of the algorithm is given in Section 2.1.

The algorithm for MAX-2SAT (and MAX-DICUT) is essentially the same, modulo using the covariance matrix from the appropriate standard SDP relaxation, and starting the process at the marginals for the corresponding variables. We explain this in greater detail below.

### 1.2.2 Overview of the Analysis for Max-Cut.

In order to analyze this algorithm, it is sufficient to understand the probability that an edge $(i, j)$ is cut as a function of the angle $\theta$ between the vectors $\mathbf{w}_i$ and $\mathbf{w}_j$. Thus, we can focus on the projection $((\mathbf{X}_t)_i, (\mathbf{X}_t)_j)$ of $\mathbf{X}_t$. We observe that $((\mathbf{X}_t)_i, (\mathbf{X}_t)_j)$ behaves like a discretization of correlated 2-dimensional Brownian motion started at $(0, 0)$, until the first time $\tau$ when it hits the boundary of the square $[-1, 1]^2$. After $\tau$, $((\mathbf{X}_t)_i, (\mathbf{X}_t)_j)$ behaves like a discretization of a 1-dimensional Brownian motion restricted to one of the sides of the square. From now on we will treat the process as being continuous, and ignore the discretization, which only adds an arbitrarily small error term in our analysis. It is convenient to apply a linear transformation to the *correlated* Brownian motion $((\mathbf{X}_t)_i, (\mathbf{X}_t)_j)$ so that it behaves like a *standard* 2-dimensional Brownian motion $\mathbf{B}_t$ started at $(0, 0)$. We show that this linear transformation maps the square $[-1, 1]^2$ to a rhombus $\mathbb{S}$ centered at $\mathbf{0}$ with internal angle $\theta$; we can then think of $\tau$ as the first time $\mathbf{B}_t$ hits the boundary of $\mathbb{S}$. After time $\tau$, the transformed process is distributed like a 1-dimensional Brownian motion on the side of the rhombus that was first hit. To analyze this process, we need to understand the probability distribution of $\mathbf{B}_\tau$. The probability measure associated with this distribution is known as the *harmonic measure* on the boundary $\partial\mathbb{S}$ of $\mathbb{S}$, with respect to the starting point $\mathbf{0}$. These transformations and connections are explained in detail in Section 2.2.

The harmonic measure has been extensively studied in probability theory and analysis. The simplest special case is the harmonic measure on the boundary of a disc centered at $\mathbf{0}$ with respect to the starting point $\mathbf{0}$. Indeed, the central symmetry of the disc and the Brownian motion implies that it is just the uniform measure. A central fact we use is that harmonic measure in 2 dimensions is preserved under conformal (i.e. angle-preserving) maps. Moreover, such maps between polygons and the unit disc have been constructed explicitly using complex analysis, and, in particular, are given by the Schwarz-Christoffel formula [2]. Thus, the Schwarz-Christoffel formula gives us an explicit formulation of sampling from the harmonic measure on the boundary $\partial\mathbb{S}$ of the rhombus: it is equivalent to sampling a uniformly random point on the boundary of the unit disc $\mathbb{D}$ centered at the origin, and mapping this point via a conformal map $F$ that sends $\mathbb{D}$ to $\mathbb{S}$. Using this formulation, in Section 2.3 we show how to write the probability of cutting the edge $(i, j)$ as an elliptic integral.

Calculating the exact value of elliptic integrals is a challenging problem. Nevertheless, by exploiting the symmetry in the MAX-CUT objective, we relate our particular elliptic integral to integrals of the incomplete beta and hypergeometric functions. We further simplify these integrals and bring them into a tractable form using several key identities from the theory of special functions. Putting everything together, we get a precise closed form expression for the probability that the Sticky Brownian Motion algorithm cuts a given edge in Theorem 2.1, and, as a consequence, we obtain the claimed guarantees for Max-Cut in Theorems 1.1 and 2.4.

### 1.2.3 Overview of the Analysis for Max-2SAT.

The algorithm for Max-2SAT is almost identical to the Max-Cut algorithm, except that the SDP solution is asymmetric, in the following sense. We can think of the SDP as describing the mean and covariance of a "pseudo-distribution" over the assignments to the variables. In the case of Max-Cut, we could assume that, without loss of generality, the mean of each variable (i.e. one-dimensional marginal) is 0 since $S$ and $\overline{S}$ are equivalent solutions. However, this is not the case for Max-2SAT. We use this information, and instead of starting the diffusion process at the center of the cube, we start it at the point given by the marginals. For convenience, and also respecting standard convention, we work in the cube $[0,1]^n$ rather than $[-1,1]^n$. Here, in the final solution $X_T$, if $(X_T)_i = 0$ we set the $i$-th variable to *true* and if $(X_T)_i = 1$, we set it to *false*. We again analyze each clause $C$ separately, which allows us to focus on the diffusion process projected to the coordinates $((X_t)_i, (X_t)_j)$, where $i$ and $j$ are the variables appearing in $C$. However, the previous approach of using the Schwarz-Christoffel formula to obtain precise bounds on the probability does not easily go through, since it relies heavily on the symmetry of the starting point of the Brownian motion. It is not clear how to extend the analysis when we change the starting point to a point other than the center, as the corresponding elliptic integrals appear to be intractable.

Instead, we appeal to a classical connection between diffusion processes and partial differential equations [31, Chapter 9]. Recall that we are focusing on a single clause $C$ with variables $i$ and $j$, and the corresponding diffusion process $((X_t)_i, (X_t)_j)$ in the unit square $[0,1]^2$ starting at a point given by the marginals and stopped at the first time $\tau$ when it hits the boundary of the square; after that time the process continues as a one-dimensional Brownian motion on the side of the square it first hit. For simplicity let us assume that both variables appear un-negated in $C$. The probability that $C$ is satisfied then equals the probability that the process ends at one of the points $(0,1)$, $(1,0)$ or $(0,0)$. Let $u : [0,1]^2 \to [0,1]$ be the function which assigns to $(x,y)$ the probability that this happens when the process is started at $(x,y)$. Since on the boundary $\partial[0,1]^2$ of the square our process is a one-dimensional martingale, the value of $u(x,y)$ is easy to compute on $\partial[0,1]^2$, and in fact equals $1-xy$. Then, in the interior of the square, we have $u(x,y) = \mathbb{E}[u((X_\tau)_i, (X_\tau)_j)]$. It turns out that this identifies $u$ as the unique solution to an elliptic partial differential equation (PDE) $\mathcal{L}u = 0$ with the Dirichlet boundary condition $u(x,y) = 1 - xy \ \ \forall (x,y) \in \partial[0,1]^2$. In our case, the operator $\mathcal{L}$ just corresponds to Laplace's operator $\mathcal{L}[u] = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ after applying a linear transformation to the variables and the domain. This connection between our rounding algorithm and PDEs is explained in Section 3.2.

Unfortunately, it is still not straightforward to solve the obtained PDE analytically. We deal with this difficulty using two natural approaches. First, we use the *maximum principle* of elliptic PDE's [20], which allows us to bound the function $u$ from below. In particular, if we can find a function $g$ such that $g(x,y) \leq u(x,y) = 1 - xy$ on the boundary of the square, and $\mathcal{L}g \geq 0$ in the interior, then the maximum principle tells us that $g(x,y) \leq u(x,y)$ for all $x, y$ in the square. We exhibit simple low-degree polynomials which satisfy the boundary conditions by design, and use the sum of squares proof system to certify non-negativity under the operator $\mathcal{L}$. In Section 3.3, we use this method to show that Sticky Brownian Motion rounding achieves approximation ratio at least 0.8749.

Our second approach is to solve the PDE numerically to a high degree of accuracy using finite element methods. We use this approach in Section 5.1 to numerically obtain results showing a 0.921 approximation ratio for Max-2SAT.

### 1.2.4 Extensions of Sticky Brownian Motion.

**Using different slowdown functions.** Recall that in the Sticky Brownian Motion rounding each increment is proportional to $\mathbf{\Delta X}_t$ sampled from a Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{W}_t$. The covariance is derived from the SDP: for example, in the case of Max-Cut, it is initially set to be the Gram matrix of the vectors produced by the SDP solution. Then, whenever a coordinate $(\mathbf{X}_t)_i$ reaches $\{-1,+1\}$, we simply zero-out the corresponding row and column of $\mathbf{W}_t$. This process can be easily modified by varying how the covariance matrix $\mathbf{W}_t$ evolves with time. Instead of zeroing out rows and columns of $\mathbf{W}_t$, we can smoothly scale them based on how far $(\mathbf{X}_{t-1})_i$ is from the boundary values $\{-1,1\}$. A simple way to do this, in the case of the Max-Cut problem, is to set

$$(\mathbf{W}_t)_{ij} = (1 - (\mathbf{X}_{t-1})_i^2)^{\alpha/2}(1 - \mathbf{X}_{t-1})_j^2)^{\alpha/2}\mathbf{w}_i \cdot \mathbf{w}_j,$$

for a constant $0 \le \alpha < 2$. Effectively, this means that the process is slowed down smoothly as it approaches the boundary of the cube $[-1,+1]^n$. This modified diffusion process, which we call Sticky Brownian Motion with Slowdown, still converges to $\{-1,+1\}^n$ in finite time. Once again, the probability of cutting an edge $(i,j)$ of our input graph can be analyzed by focusing on the two-dimensional projection $((\mathbf{X}_t)_i, (\mathbf{X}_t)_j)$ of $\mathbf{X}_t$. Moreover, we can still use the general connection between diffusion processes and PDE's mentioned above. That is, if we write $u(x,y) : [-1,1]^2 \to [0,1]$ for the probability that edge $(i,j)$ is cut if the process is started at $(x,y)$, then $u$ can be characterized as the solution of an elliptic PDE with boundary conditions $u(x,y) = \frac{1-xy}{2} \quad \forall (x,y) \in \partial[-1,1]^2$. We solve this PDE numerically using the finite element method to estimate the approximation ratio for a fixed value of the parameter $\alpha$, and then we optimize over $\alpha$. At the value $\alpha =$1.61 our numerical solution shows an approximation ratio that matches the Goemans-Williamson approximation of MAX-CUT up to the first three digits after the decimal point. We also analyze an analogous algorithm for MAX-2SAT and show that for $\alpha =$1.61 it achieves an approximation ratio of 0.929. The detailed analysis of the slowed down Sticky Brownian Motion rounding is given in Section 5.1.

**A higher-dimensional version.** We also consider a higher-dimensional version of the Sticky Brownian Motion rounding, in which the Brownian motion evolves in $n+1$ dimensions rather than $n$. This rounding is useful for asymmetric problems like MAX-DICUT[4] in which the SDP produces non-uniform marginals, as we discussed above in the context of MAX-2SAT. Such an SDP has a vector $\mathbf{w}_0$ in addition to $\mathbf{w}_1, \ldots, \mathbf{w}_n$, and the marginals are given by $\mathbf{w}_0 \cdot \mathbf{w}_i$. Now, rather than using the marginals to obtain a different starting point, we consider the symmetric Sticky Brownian Motion process starting from the center but using all the $n+1$ vectors $\mathbf{w}_0, \ldots, \mathbf{w}_n$. At the final step $T$ of the process, in the case of MAX-DICUT, the variables whose value is equal to $(\mathbf{X}_T)_0$ are assigned to the left side of the cut, and the variables with the opposite value are assigned to the right side of the cut. Thus, for an edge $i \to j$ to be cut, it must be the case that $(\mathbf{X}_T)_i = (\mathbf{X}_T)_0$ and $(\mathbf{X}_T)_j = 1 - (\mathbf{X}_T)_0$. While analyzing the probability that this happens is a question about Brownian motion in three rather than two dimensions, we reduce it to a two-dimensional question via the inclusion-exclusion principle. After this reduction, we can calculate the probability that an edge is cut by using the exact formula proved earlier for the MAX-CUT problem. Our analysis, which is given in Section 5.2, shows that this $(n+1)$-dimensional Sticky Brownian Motion achieves an approximation of 0.79 for MAX-DICUT. Moreover, combining the two ideas, of changing the covariance matrix at each step, as well as performing the $n+1$-dimensional Sticky Brownian Motion, achieves a ratio of 0.81.

#### 1.2.5 Overview of the Analysis for Max-Cut-SC.

The starting point for our algorithm for the MAX-CUT-SC problem is a stronger SDP relaxation derived using the Sum of Squares (SoS) hierarchy. Similar relaxations were previously considered in [8, 34] for the MAX-BISECTION problem. In addition to giving marginal values and a covariance matrix for a "pseudo-distribution" over feasible solutions, the SoS SDP makes it possible to condition on small sets of variables. The global correlation rounding method [13, 24] allows us to choose variables to condition on so that, after the conditioning, the covariance matrix has small entries on average. Differing from previous works [8, 34], we then run the Sticky Brownian Motion rounding defined by the resulting marginals and covariance matrix. We can analyze the weight of cut edges using the PDE approach outlined above. The main new challenge is to bound the amount by which the side constraints are violated. To do so, we show that Sticky Brownian Motion concentrates tightly around its mean, and, in particular, it satisfies sub-Gaussian concentration in directions corresponding to sets of vertices. Since the mean of the Sticky Brownian Motion is given by the marginals, which satisfy all side constraints, we can bound how much constraints are violated via the concentration and a union bound. To show this key concentration property, we use the fact that the covariance that defines the diffusion has small entries, and that Brownian Motion is a martingale. Then the concentration inequality follows, roughly, from a continuous analogue of Azuma's inequality. The detailed analysis is given in Section 4. We again remark that such sub-Gaussian concentration bounds are not known to hold for the random hyperplane rounding method or its generalizations as considered in [8, 34].

### 1.3 Related Work.

Due to space constraints, this section is omitted from here. Details are given in the full version of the paper.

## 2 Brownian Rounding for Max-Cut via Conformal Mappings

In this section, we use MAX-CUT as a case study for the method of rounding a semi-definite relaxation via Sticky

[4] The input for MAX-DICUT is a directed graph $G = (V,E)$ and the goal is to find a cut $S \subseteq V$ that maximizes the number of edges going from $S$ to $\overline{S}$.

Brownian Motion. Recall, in an instance of the Max-Cut problem we are given a graph $G = (V, E)$ with edge weights $a : E \to \mathbb{R}_+$ and the goal is to find a subset $S \subset V$ that maximizes the total weight of edges crossing the cut $(S, V \setminus S)$, i.e., $a(\delta(S)) := \sum_{\{u,v\}\in E: u\in S, v\notin S} a_{uv}$. We first introduce the standard semi-definite relaxation for the problem and introduce the sticky Brownian rounding algorithm. To analyze the algorithm, we use the invariance of Brownian motion with respect to conformal maps, along with several identities of special functions.

**2.1 SDP Relaxation and Sticky Brownian Rounding Algorithm.** Before we proceed, we recall again the SDP formulation for the Max-Cut problem, famously studied by Goemans and Williamson [23].

$$\begin{aligned} \max \quad & \sum_{e=(i,j)\in E} a(e)\frac{(1-\mathbf{w}_i\cdot\mathbf{w}_j)}{2} & \\ s.t. \quad & \mathbf{w}_i\cdot\mathbf{w}_i = 1 & \forall i = 1, ..., n. \end{aligned}$$

We now describe the Sticky Brownian Motion rounding algorithm specialized to the Max-Cut problem. Let $\mathbf{W}$ denote the positive semi-definite correlation matrix defined by the vectors $\mathbf{w}_1, \dots, \mathbf{w}_n$, *i.e.*, for every $1 \le i, j \le n$ we have that: $\mathbf{W}_{i,j} = \mathbf{w}_i \cdot \mathbf{w}_j$. Given a solution $\mathbf{W}$ to the semi-definite program, we perform the following rounding process: start at the origin and perform a Brownian motion inside the $[-1, 1]^n$ hypercube whose correlations are governed by $\mathbf{W}$. Additionally, the random walk is *sticky*: once a coordinate reaches either $-1$ or $+1$ it is fixed and does not change anymore.

Formally, we define a random process $\{\mathbf{X}_t\}_{t\ge 0}$ as follows. We fix $\mathbf{X}_0 = \mathbf{0}$. Let $\{\mathbf{B}_t\}_{t\ge 0}$ be standard Brownian motion in $\mathbb{R}^n$ started at the origin,[5] and let $\tau_1 = \inf\{t : \mathrm{x}_0 + \mathbf{W}^{1/2}\mathbf{B}_t \notin [-1, 1]^n\}$ be the first time $\mathrm{x}_0 + \mathbf{W}^{1/2}\mathbf{B}_t$ exits the cube. With probability 1, you can assume that $\tau_1$ is also the first time that the process lies on the boundary of the cube. Here $\mathbf{W}^{1/2}$ is the principle square root of $\mathbf{W}$. Then, for all $0 \le t \le \tau_1$ we define

$$\mathbf{X}_t = \mathrm{x}_0 + \mathbf{W}^{1/2}\mathbf{B}_t.$$

This defines the process until the first time it hits a face of the cube. From this point on, we will force it to stick to this face. Let $A_t = \{i : (\mathbf{X}_t)_i \ne \pm 1\}$ be the active coordinates of the process at time $t$, and let $F_t = \{\mathbf{x} \in [-1, 1]^n : x_i = (\mathbf{X}_t)_i \forall i \in A_t\}$ be the face of the cube on which $\mathbf{X}_t$ lies at time $t$. With probability 1, $F_{\tau_1}$ has dimension $n - 1$. We define the covariance matrix $(\mathbf{W}_t)_{ij} = \mathbf{W}_{ij}$ when $i, j \in A_t$, and $(\mathbf{W}_t)_{ij} = 0$ otherwise. Then we take $\tau_2 = \inf\{t \ge \tau_1 : \mathbf{X}_{\tau_1} + \mathbf{W}_{\tau_1}^{1/2}(\mathbf{B}_t - \mathbf{B}_{\tau_1}) \notin F_{\tau_1}\}$ to be the first time that Brownian motion started at $\mathbf{X}_{\tau_1}$ with covariance given by $\mathbf{W}_{\tau_1}$ exits the face $F_{\tau_1}$. Again, with probability 1, we can assume that this is also the first time the process lies on the boundary of $F_{\tau_1}$. For all $\tau_1 < t \le \tau_2$ we define

$$\mathbf{X}_t = \mathbf{X}_{\tau_1} + \mathbf{W}_{\tau_1}^{1/2}(\mathbf{B}_t - \mathbf{B}_{\tau_1}).$$

Again, with probability 1, dim $F_{\tau_2} = n - 2$. The process is defined analogously from here on. In general, $\tau_i = \inf\{t \ge \tau_{i-1} : \mathbf{X}_{\tau_{i-1}} + \mathbf{W}_{\tau_{i-1}}^{1/2}(\mathbf{B}_t - \mathbf{B}_{\tau_{i-1}}) \notin F_{\tau_{i-1}}\}$ is (with probability 1) the first time that the process hits a face of the cube of dimension $n - i$. Then for $\tau_{i-1} < t \le \tau_i$ we have $\mathbf{X}_t = \mathbf{X}_{\tau_{i-1}} + \mathbf{W}_{\tau_{i-1}}^{1/2}(\mathbf{B}_t - \mathbf{B}_{\tau_{i-1}})$. At time $\tau_n$, $\mathbf{X}_{\tau_n} \in \{-1, 1\}^n$, so the process remains fixed, i.e. for any $t \ge \tau_n$, $\mathbf{X}_t = \mathbf{X}_{\tau_n}$. The output of the algorithm then corresponds to a cut $S \subseteq V$ defined as follows:

$$S = \{i \in V : (\mathbf{X}_{\tau_n})_i = 1\}.$$

We say that a pair of nodes $\{i, j\}$ is *separated* when $|S \cap \{i, j\}| = 1$.

**Remark:** While we have defined the algorithm as a continuous diffusion process, driven by Brownian motion, a standard discretization will yield a polynomial time algorithm that achieves the same guarantee up to an error that is polynomially small. Such a discretization was outlined in the Introduction. An analysis of the error incurred by discretizing a continuous diffusion process in this way can be found, for example, in [21] or the book [22]. More sophisticated discrete simulations of such diffusion processes are also available, and can lead to better time complexity as a function of the error. One example is the Walk on Spheres algorithm analyzed by Binder and Braverman [15]. This algorithm allows us to draw a sample $\mathbf{X}_\tau$ from the continuous diffusion process, stopped at a random time $\tau$, such that $\mathbf{X}_\tau$ is within distance $\varepsilon$ from the boundary of the cube $[-1, 1]^n$. The time necessary to sample $\mathbf{X}_\tau$ is polynomial in $n$ and $\log(1/\varepsilon)$. We can then round $\mathbf{X}_\tau$ to the nearest point on the boundary of the cube, and continue the simulation starting from this rounded point. It is straightforward to show, using the fact that the probability to cut an edge is continuous in the starting point of our process, that if we set $\varepsilon = o(n^{-1})$, then the approximation ratio achieved by this simulation is within an $o(1)$ factor from the one achieved by the continuous process. In the rest of the paper, we focus on the continuous process since our methods of analysis are naturally amenable to it.

**2.2 Analysis of the Algorithm.** Our aim is to analyze the expected value of the cut output by the Sticky

[5] We will always assume that a standard Brownian motion starts at the origin. See the full version for a precise definition.

Brownian Motion rounding algorithm. Following Goemans and Williamson [23], we aim to bound the probability an edge is cut as compared to its contribution to the SDP objective. Theorem 2.1 below gives an *exact* characterization of the probability of separating a pair of vertices $\{i, j\}$ in terms of the gamma function and hypergeometric functions. We refer to the full version for the definitions of these functions and a detailed exposition of their basic properties.

Theorem 2.1. *The probability that the Sticky Brownian Motion rounding algorithm will separate a pair $\{i, j\}$ of vertices for which $\theta = cos^{-1}(\mathbf{w}_i \cdot \mathbf{w}_j)$ equals*

$$1 - \frac{\Gamma(\frac{a+1}{2})}{\Gamma(\frac{1-a}{2})\Gamma(\frac{a}{2}+1)^2} \cdot {}_3F_2\left[\begin{matrix}\frac{1+a}{2}, & \frac{1+a}{2}, & \frac{a}{2} \\ \frac{a}{2}, & \frac{a}{2}+1 & \end{matrix}; 1\right],$$

*where $a = \theta/\pi$, $\Gamma$ is the gamma function, and ${}_3F_2$ is the hypergeometric function.*

Theorem 1.1 will now follow from the following corollary of Theorem 2.1. The corollary follows from numerical estimates of the gamma and hypergeometric functions.

Corollary 2.1. *For any pair $\{i, j\}$, the probability that the pair $\{i, j\}$ is separated is at least $0.861 \cdot \frac{1-\mathbf{w}_i \cdot \mathbf{w}_j}{2}$.*

We now give an outline of the proof of Theorem 2.1. The plan is to first show that the desired probability can be obtained by analyzing the two-dimensional standard Brownian motion starting at the center of a rhombus. Moreover, the probability of separating $i$ and $j$ can be computed using the distribution of the first point on the boundary that is hit by the Brownian motion. Conformal mapping and, in particular, the Schwarz-Christoffel formula, allows us to obtain a precise expression for such a distribution and thus for the separation probability, as claimed in the theorem. We now expand on the above plan.

First observe that to obtain the probability $i$ and $j$ are separated, it is enough to consider the 2-dimensional process obtained by projecting to the $i^{th}$ and $j^{th}$ coordinates of the vector $\mathbf{X}_t$. Projecting the process onto these coordinates, we obtain a process $\tilde{\mathbf{X}}_t \in \mathbb{R}^2$ that can be equivalently defined as follows. Let

$$\tilde{\mathbf{W}} = \begin{pmatrix} 1 & \cos(\theta) \\ \cos(\theta) & 1 \end{pmatrix},$$

where $\theta$ is the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$. Let $\mathbf{B}_t$ be standard Brownian motion in $\mathbb{R}^2$ started at 0, and let $\tau = \inf\{t : \tilde{\mathbf{W}}^{1/2}\mathbf{B}_t \notin [-1, 1]^t\}$ be the first time the process hits the boundary of the square. Then for all $0 \le t \le \tau$ we define $\tilde{\mathbf{X}}_t = \tilde{\mathbf{W}}^{1/2}\mathbf{B}_t$. Any coordinate $k$ for which $(\tilde{\mathbf{X}}_\tau)_k \in \{\pm 1\}$ remains fixed from then on, i.e. for all $t > \tau$, $(\tilde{\mathbf{X}}_t)_k = (\tilde{\mathbf{X}}_\tau)_k$. The coordinate $\ell$ that is not fixed at time $\tau$ (one exists with probability 1) continues to perform one-dimensional Brownian motion started from $(\tilde{\mathbf{X}}_\tau)_\ell$ until it hits $-1$ or $+1$, at which point it also becomes fixed. Let $\sigma$ be the time this happens; it is easy to show that $\sigma < \infty$ with probability 1, and, moreover, $\mathbb{E}[\sigma] < \infty$. We say that the process $\{\tilde{\mathbf{X}}_t\}_{t \ge 0}$ is *absorbed* at the vertex $\tilde{\mathbf{X}}_\sigma \in \{-1, 1\}^2$.

Observation 2.1. *The probability that the algorithm separates vertices $i$ and $j$ equals*

$$\Pr\left[\{\tilde{\mathbf{X}}_t\}_t \textit{ is absorbed in } \{(+1, -1), (-1, +1)\}\right].$$

With an abuse of notation, we denote $\tilde{\mathbf{X}}_t$ by $\mathbf{X}_t$ and $\tilde{\mathbf{W}}$ by $\mathbf{W}$ for the rest of the section which is aimed at analyzing the above probability. We also denote by $\rho = \cos(\theta)$ the correlation between the two coordinates of the random walk, and call the two-dimensional process just described a $\rho$-correlated walk. It is easier to bound the probability that $i$ and $j$ are separated by transforming the $\rho$-correlated walk inside $[-1, 1]^2$ into a standard Brownian motion inside an appropriately scaled rhombus. We do this by transforming $\{\mathbf{X}_t\}_{t \ge 0}$ linearly into an auxiliary random process $\{\mathbf{Y}_t\}_{t \ge 0}$ which will be sticky inside a rhombus (see Figures (1a)-(1b)). Formally, given the random process $\{\mathbf{X}_t\}_{t \ge 0}$, we consider the process $\mathbf{Y}_t = \mathbf{O} \cdot \mathbf{W}^{-1/2} \cdot \mathbf{X}_t$, where $\mathbf{O}$ is a rotation matrix to be chosen shortly. Recalling that for $0 \le t \le \tau$ the process $\{\mathbf{X}_t\}_{0 \le t \le \tau}$ is distributed as $\{\mathbf{W}^{1/2}\mathbf{B}_t\}_{0 \le t \le \tau}$, we have that, for all $0 \le t \le \tau$,

$$\mathbf{Y}_t = \mathbf{O} \cdot \mathbf{B}_t \equiv \mathbf{B}_t.$$

Above $\equiv$ denotes equality in distribution, and follows from the invariance of Brownian motion under rotation. Applying $\mathbf{O}\mathbf{W}^{-1/2}$ to the points inside $[-1, 1]^2$, we get a rhombus $\mathbb{S}$ with vertices $b_1, \ldots, b_4$, which are the images of the points $(+1, -1), (+1, +1), (-1, +1), (-1, -1)$, respectively. We choose $\mathbf{O}$ so that $b_1$ lies on the positive $x$-axis and $b_2$ on the positive $y$-axis. Since $\mathbf{O}\mathbf{W}^{-1/2}$ is a linear transformation, it maps the interior of $[-1, 1]^2$ to the interior of $\mathbb{S}$ and the sides of $[-1, 1]^2$ to the sides of $\mathbb{S}$. We have then that $\tau$ is the first time $\mathbf{Y}_t$ hits the boundary of $\mathbb{S}$, and that after this time $\mathbf{Y}_t$ sticks to the side of $\mathbb{S}$ that it first hit and evolves as (a scaling of) one-dimensional Brownian motion restricted to this side, and started at $\mathbf{Y}_\tau$. The process then stops evolving at the time $\sigma$ when $\mathbf{Y}_\sigma \in \{b_1, \ldots, b_4\}$. We say that $\{\mathbf{Y}_t\}_{t \ge 0}$ is absorbed at $\mathbf{Y}_\sigma$.

The following lemma, whose proof appears in the full version, formalizes the main facts we use about this transformation.

LEMMA 2.1. *Applying the transformation* $\mathbf{OW}^{-1/2}$ *to* $\{\mathbf{X}_t\}_{t\geq 0}$ *we get a new random process* $\{\mathbf{Y}_t\}_{t\geq 0}$ *which has the following properties:*

1. *If* $\mathbf{X}_t$ *is in the interior/boundary/vertex of* $[-1,1]^2$ *then* $\mathbf{Y}_t$ *is in the interior/boundary/vertex of* $\mathbb{S}$*, respectively.*

2. $\mathbb{S}$ *is a rhombus whose internal angles at* $b_1$ *and* $b_3$ *are* $\theta$*, and at* $b_2$ *and* $b_4$ *are* $\pi - \theta$*. The vertex* $b_1$ *lies on the positive* $x$*-axis, and* $b_2, b_3, b_4$ *are arranged counter-clockwise.*

3. *The probability that the algorithm will separate the pair* $\{i, j\}$ *is exactly* $\Pr[\mathbf{Y}_t$ *is absorbed in* $b_1$ *or* $b_3]$*.*

In the following useful lemma we show that, in order to compute the probability that the process $\{\mathbf{Y}_t\}_{t\geq 0}$ is absorbed in $b_1$ or $b_3$, it suffices to determine the distribution of the first point $\mathbf{Y}_\tau$ on the boundary $\partial\mathbb{S}$ that the process $\{\mathbf{Y}_t\}_{t\geq 0}$ hits. This distribution is a probability measure on $\partial\mathbb{S}$ known in the literature as *the harmonic measure* (with respect to the starting point 0). We denote it by $\mu_{\partial\mathbb{S}}$. The statement of the lemma follows.

LEMMA 2.2.

$$\Pr[\mathbf{Y}_t \textit{ is absorbed in } b_1 \textit{ or } b_3] = 4 \cdot \int_{b_1}^{b_2} 1 - \frac{\|p - b_1\|}{\|b_2 - b_1\|} d\mu_{\partial\mathbb{S}}(p).$$

*Proof.* Since both $\mathbb{S}$ and Brownian motion are symmetric with respect to reflection around the coordinate axes, we see that $\mu_{\partial\mathbb{S}}$ is the same as we go from $b_1$ to $b_2$ or $b_4$, and as we go from $b_3$ to $b_2$ or $b_4$. Therefore,

$$\Pr[\text{pair } \{i,j\} \text{ is separated}] = 4 \cdot \Pr[\text{pair } \{i,j\} \text{ is separated} \mid \mathbf{Y}_\tau \text{ lies on } [b_1, b_2]].$$

The process $\{\mathbf{Y}_t\}_{\tau \leq t \leq \sigma}$ is a one-dimensional martingale, so $\mathbb{E}[\mathbf{Y}_\sigma | \mathbf{Y}_\tau] = \mathbf{Y}_\tau$ by the optional stopping theorem [29, Proposition 2.4.2]. If we also condition on $\mathbf{Y}_\tau \in [b_1, b_2]$, we have that $\mathbf{Y}_\sigma \in \{b_1, b_2\}$. An easy calculation then shows that the probability of being absorbed in $b_1$ conditional on $\mathbf{Y}_\tau$ and on the event $\mathbf{Y}_\tau \in [b_1, b_2]$ is exactly $\frac{\|\mathbf{Y}_\tau - b_2\|}{\|b_2 - b_1\|} = 1 - \frac{\|\mathbf{Y}_\tau - b_1\|}{\|b_2 - b_1\|}$. Then,

$$\Pr[\text{pair } \{i,j\} \text{ is separated} \mid \mathbf{Y}_\tau \in [b_1, b_2]] = \mathbb{E}\left[1 - \frac{\|\mathbf{Y}_\tau - b_1\|}{\|b_2 - b_1\|}\right] = \int_{b_1}^{b_2} 1 - \frac{\|p - b_1\|}{\|b_2 - b_1\|} d\mu_{\partial\mathbb{S}}(p).$$

This proves the lemma. □

To obtain the harmonic measure directly for the rhombus $\mathbb{S}$ we appeal to conformal mappings. We use the fact that the harmonic measure can be defined for any simply connected region $U$ in the plane with 0 in its interior. More precisely, let $\mathbf{B}_t$ be standard 2-dimensional Brownian motion started at 0, and $\tau(U) = \inf\{t : \mathbf{B}_t \notin U\}$ be the first time it hits the boundary of $U$. Then $\mu_{\partial U}$ denotes the probability measure induced by the distribution of $\mathbf{B}_{\tau(U)}$, and is called the harmonic measure on $\partial U$ (with respect to 0). When $U$ is the unit disc centered at 0, the harmonic measure is uniform on its boundary because Brownian motion is invariant under rotation. Then the main idea is to use conformal maps to relate harmonic measures on the different domains, namely the disc and our rhombus $\mathbb{S}$.

### 2.3 Conformal Mapping.

Before we proceed further, it is best to transition to the language of complex numbers and identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$. A complex function $F : U \to V$ where $U, V \subseteq \mathbb{C}$ is conformal if it is holomorphic (i.e. complex differentiable) and its derivative $f'(x) \neq 0$ for all $x \in U$. The key fact we use about conformal maps is that they preserve harmonic measure. Below we present this theorem from Mörters and Peres [29] specialized to our setting. In what follows, $\mathbb{D}$ will be the unit disc in $\mathbb{C}$ centered at 0.

THEOREM 2.2. *[29, p. 204, Theorem 7.23]. Suppose* $F_\theta$ *is a conformal map from the unit disk* $\mathbb{D}$ *to* $\mathbb{S}$*. Let* $\mu_{\partial\mathbb{D}}$ *and* $\mu_{\partial\mathbb{S}}$ *be the harmonic measures with respect to* 0*. Then* $\mu_{\partial\mathbb{D}} \circ F_\theta^{-1} = \mu_{\partial\mathbb{S}}$*.*

Thus the above theorem implies that in our setting, the probability that a standard Brownian motion will first hit any segment $S$ of the boundary of $\mathbb{D}$ is the same as the probability of the standard Brownian motion first hitting its image under $F_\theta$, i.e. $F_\theta(S)$ in $\partial\mathbb{S}$.

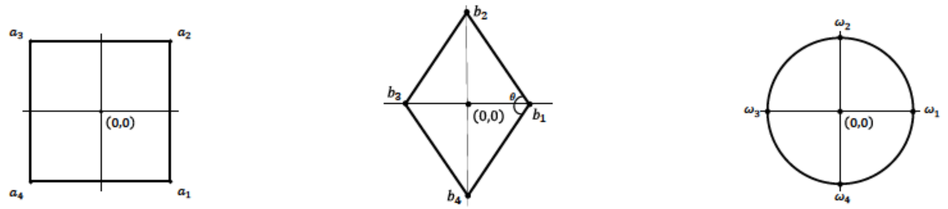
To complete the picture, the Schwarz-Christoffel formula gives a conformal mapping from the unit disc $\mathbb{D}$ to $\mathbb{S}$ that we utilize.

THEOREM 2.3. *[2, Theorem 5, Section 2.2.2] Define the function* $F_\theta(\omega)$ *by*

$$F_\theta(\omega) = \int_{s=0}^{\omega} f_\theta(s) ds = \int_{s=0}^{\omega} (1-s)^{-(1-\theta/\pi)}(1+s)^{-(1-\theta/\pi)}(s-i)^{-\theta/\pi}(s+i)^{-\theta/\pi} ds.$$

*Then, for some real number* $c > 0$*,* $cF_\theta(\omega)$ *is a conformal map from the unit-disk* $\mathbb{D}$ *to the rhombus* $\mathbb{S}$*.*

The conformal map has some important properties which will aid us in calculating the probabilities. We



(a) $\{\mathbf{X}_t\}_{t\geq 0}$ in $[-1,1]^2$ square (b) $\{\mathbf{Y}_t\}_{t\geq 0}$ in $\mathbb{S}$ (c) $\{\mathbf{B}_t\}_{t\geq 0}$ in $\mathbb{D}$

Figure 1: Figure (a) depicts $\{\mathbf{X}_t\}_{t\geq 0}$ in the $[-1,1]^2$ square, Figure (b) depicts $\{\mathbf{Y}_t\}_{t\geq 0}$ in the rhombus $\mathbb{S}$, and Figure (c) depicts $\{\mathbf{B}_t\}_{t\geq 0}$ in the unit disc $\mathbb{D}$. The linear transformation $\mathbf{W}^{-1/2}$ transforms the $[-1,1]^2$ square to $\mathbb{S}$ (Figure (a) to Figure (b)), whereas the conformal mapping $F_\theta$ transforms $\mathbb{D}$ to $\mathbb{S}$ (Figure (c) to Figure (b)).

collect them in the following lemma, which follows from standard properties of the Schwarz-Christoffel integral [2], and is easily verified.

Lemma 2.3. *The conformal map $cF_\theta(\omega)$ has the following properties:*

1. *The four points located at $\{1, i, -1, -i\}$ map to the four vertices $\{b_1, \ldots, b_4\}$ of the rhombus $\mathbb{S}$, respectively.*
2. *The origin maps to the origin.*
3. *The boundary of the unit-disk $\mathbb{D}$ maps to the boundary of $\mathbb{S}$. Furthermore, the points in the arc from 1 to $i$ map to the segment $[b_1, b_2]$.*

Define the function $r : [0, \pi/2] \to \mathbb{R}$ as $r(\phi) := |F_\theta(e^{i\phi}) - F_\theta(1)|$.

Lemma 2.4. *The probability that vertices $\{i, j\}$ are separated, given that the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$ is $\theta$, is*

$$\frac{2}{\pi}\int_0^{\pi/2} 1 - \frac{r(\phi)}{r(\pi/2)}d\phi$$

*Proof.* Rewriting the expression in Lemma 2.2 in complex number notation, we have

$$\Pr[\{i,j\} \text{ separated}] = 4\cdot\int_{b_1}^{b_2} 1 - \frac{|z - b_1|}{|b_2 - b_1|}d\mu_{\partial\mathbb{S}}(z) = $$
$$4\cdot\int_{b_1}^{b_2} 1 - \frac{|z - cF_\theta(1)|}{c|F_\theta(i) - F_\theta(1)|}d\mu_{\partial\mathbb{S}}(z).$$

Since the conformal map $F_\theta$ preserves the harmonic measure between the rhombus $\mathbb{S}$ and the unit-disk $\mathbb{D}$ (see Theorem 2.2) and by Lemma 2.3, the segment from $b_1$ to $b_2$ is the image of the arc from 1 to $i$ under $cF_\theta$, we can rewrite the above as

$$= 4\cdot\int_0^{\pi/2} 1 - \frac{|cF_\theta(e^{i\phi}) - cF_\theta(1)|}{c|F_\theta(i) - F_\theta(1)|}d\mu_{\partial\mathbb{D}}(e^{i\phi}).$$

The harmonic measure $\mu_{\partial\mathbb{D}}$ on the unit-disk is uniform due to the rotational symmetry of Brownian motion.

$$= 4\cdot\int_0^{\pi/2} 1 - \frac{|cF_\theta(e^{i\phi}) - cF_\theta(1)|}{c|F_\theta(i) - F_\theta(1)|}\frac{d\phi}{2\pi}.$$

Simplifying the above, we see that the right hand side above equals

$$\frac{2}{\pi}\cdot\int_0^{\pi/2} 1 - \frac{|F_\theta(e^{i\phi}) - F(1)|}{|F_\theta(i) - F(1)|}d\phi = \frac{2}{\pi}\cdot\int_0^{\pi/2} 1 - \frac{r(\phi)}{r(\pi/2)}d\phi.$$

This completes the proof. □

To calculate the approximation ratio exactly, we will make use of the theory of special functions. While these calculations are technical, they are not trivial. To aid the reader, we give a brief primer in the full version and refer them to the work of Andrews et al. [4], Beals and Wong [14] for a more thorough introduction.

The proof of Theorem 2.1, will follow from the following key claims whose proofs are available in the full version. Letting $a = \theta/\pi$ and $b = 1 - a$, we have

Claim 2.1. $r(\phi) = \frac{\beta_{\sin^2\phi}(a/2, b/2)}{4}$, *when $\phi \in [0, \pi/2]$.*

Claim 2.2.

$$4\cdot\int_0^{\pi/2} r(\phi)d\phi = \frac{\beta(a/2 + 1/2, 1/2)}{2a}\cdot {}_3F_2\left[\begin{matrix}\frac{1+a}{2}, & \frac{1+a}{2}, & \frac{a}{2} \\ \frac{a}{2}, & \frac{a}{2}+1 & \end{matrix}; 1\right].$$

### 2.4 Asymptotic Calculation for $\theta$ close to $\pi$.

We consider the case when the angle $\theta = (1-\epsilon)\cdot\pi$ as $\epsilon \to 0$. The hyperplane-rounding algorithm separates such an edge by $\theta/\pi$, and hence has a separation probability of $1 - \epsilon$. We show a similar asymptotic behaviour for the Brownian rounding algorithm, albeit with slightly worse constants. We defer the proof to the full version.

Theorem 2.4. *Given an edge $\{i, j\}$ with $\cos^{-1}(\mathbf{w}_i^T\mathbf{w}_j) = \theta = (1-\epsilon)\pi$, the Sticky Brownian Motion rounding will cut the edge with probability at least $1 - \left(\frac{4}{\pi}\epsilon + O(\epsilon^2)\right)$.*

## 3 Brownian Rounding for Max-2SAT via Partial Differential Equations

In this section we use Max-2SAT as a case study for extending the Sticky Brownian Motion rounding method to other constraint satisfaction problems besides Max-Cut. In the Max-2SAT problem we are given $n$ variables $z_1, \ldots, z_n$ and $m$ clauses $C_1, \ldots, C_m$, where the $j^{\text{th}}$ clause is of the form $y_{j_1} \vee y_{j_2}$ ($y_j$ is a literal of $z_j$, *i.e.*, $z_j$ or $\overline{z}_j$). The goal is to assign to each variable $z_i$ a value of true or false so as to maximize the number of satisfied clauses.

### 3.1 Semi-definite Relaxation and Brownian Rounding Algorithm.

The standard SDP relaxation used for Max-2SAT is the following:

$$\max \quad \sum_{j=1}^{m} (1 - \mathbf{v}_{j_1} \cdot \mathbf{v}_{j_2})$$

$$\text{s.t.} \quad \mathbf{v}_0 \cdot \mathbf{v}_0 = 1 \tag{3.1}$$

$$\mathbf{v}_0 \cdot \mathbf{v}_i = \mathbf{v}_i \cdot \mathbf{v}_i \qquad \forall i = -n, \ldots, n \tag{3.2}$$

$$\mathbf{v}_i \cdot \mathbf{v}_{-i} = 0 \qquad \forall i = 1, \ldots, n \tag{3.3}$$

$$\mathbf{v}_0 \cdot (\mathbf{v}_i + \mathbf{v}_{-i}) = 1 \qquad \forall i = 1, \ldots, n \tag{3.4}$$

$$1 \geq \mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_j \cdot \mathbf{v}_0 - \mathbf{v}_i \cdot \mathbf{v}_j \quad \forall i, j = -n, \ldots, n \tag{3.5}$$

$$\mathbf{v}_i \cdot \mathbf{v}_0 \geq \mathbf{v}_i \cdot \mathbf{v}_j \qquad \forall i, j = -n, \ldots, n \tag{3.6}$$

$$\mathbf{v}_i \cdot \mathbf{v}_j \geq 0 \qquad \forall i, j = -n. \ldots, n \tag{3.7}$$

In the above $\mathbf{v}_0$ is a unit vector that denotes the *false* assignment (constraint 3.1), whereas a zero vector denotes the *true* assignment. We use the standard notation that $\mathbf{v}_i$ denotes the literal $z_i$ and $\mathbf{v}_{-i}$ denotes the literal $\overline{z}_i$. Therefore, $\mathbf{v}_i \cdot \mathbf{v}_{-i} = 0$ for every $i = 1, \ldots. n$ (constraints 3.3 and 3.4) since $z_i$ needs to be either true or false. The remainder of the constraints (constraints 3.5, 3.6 and 3.7) are equivalent to the $\ell_2^2$ triangle inequalities over all triples of vectors that include $\mathbf{v}_0$.

When trying to generalize the Brownian rounding algorithm for Max-Cut presented in Section 2 to Max-2SAT, there is a problem: unlike Max-Cut the Max-2SAT problem is *not* symmetric. Specifically, for Max-Cut both $S$ and $\overline{S}$ are equivalent solutions having the same objective value. However, for Max-2SAT an assignment to the variables $z_1 = \alpha_1, \ldots, z_n = \alpha_n$ is *not* equivalent to the assignment $z_1 = \overline{\alpha}_1, \ldots, z_n = \overline{\alpha}_n$ (here $\alpha_i \in \{0, 1\}$ and $\overline{\alpha}_i = 1 \oplus \alpha_i$). For example, if $\mathbf{v}_i \cdot \mathbf{v}_0 = 1$ then we would like the Brownian rounding algorithm to always assign $z_i$ to false. The Brownian rounding for Max-Cut cannot handle such a requirement. In order to tackle the above problem we incorporate $\mathbf{v}_0$ into both the starting point of the Brownian motion and the covariance matrix.

Let us now formally define the Brownian rounding algorithm for Max-2SAT. For simplicity of presentation denote for every $i = 1, \ldots, n$ by $x_i$ the *marginal value* of $z_i$, formally: $x_i := \mathbf{v}_i \cdot \mathbf{v}_0$. Additionally, let $\mathbf{w}_i$ be the (unique) unit vector in the direction of the projection of $\mathbf{v}_i$ to the subspace orthogonal to $\mathbf{v}_0$, *i.e.*, $\mathbf{w}_i$ satisfies $\mathbf{v}_i = x_i \mathbf{v}_0 + \sqrt{x_i - x_i^2}\, \mathbf{w}_i$.[6] Similarly to Max-Cut, our Sticky Brownian Motion rounding algorithm performs a random walk in $\mathbb{R}^n$, where the $i^{\text{th}}$ coordinate corresponds to the variable $z_i$. For simplicity of presentation, the random walk is defined in $[0, 1]^n$ as opposed to $[\pm 1]^n$, where 1 denotes false and 0 denotes true.[7] Unlike Max-Cut, the starting point $\mathbf{X}_0$ is not the center of the cube. Instead, we use the marginals, and set $(\mathbf{X}_0)_i := x_i$. The covariance matrix $\mathbf{W}$ is defined by $\mathbf{W}_{i,j} := \mathbf{w}_i \cdot \mathbf{w}_j$ for every $i, j = 1, \ldots, n$, and similarly to Max-Cut, let $\mathbf{W}^{1/2}$ be the principle square root of $\mathbf{W}$. Letting $\{\mathbf{B}_t\}_{t \geq 0}$ denote standard Brownian motion in $\mathbb{R}^n$, we define $\tau_1 = \inf\{t : \mathbf{W}^{1/2}\mathbf{B}_t + \mathbf{X}_0 \notin [0, 1]^n\}$ to be the first time the process hits the boundary of $[0, 1]^n$. Then, for all times $0 \leq t \leq \tau_1$, the process $\mathbf{X}_t$ is defined as

$$\mathbf{X}_t = \mathbf{W}^{1/2}\mathbf{B}_t + \mathbf{X}_0.$$

After time $\tau_1$, we force $\mathbf{X}_t$ to stick to the face $F_1$ hit at time $\tau_1$: i.e. if $(\mathbf{X}_{\tau_1})_i \in \{0, 1\}$, then we fix it forever, by zeroing out the $i$-th row and column of the covariance matrix of $\mathbf{W}$ for all future time steps. The rest of the process is defined analogously to the one for Max-Cut: whenever $\mathbf{X}_t$ hits a lower dimensional face of $[0, 1]^n$, it is forced to stick to it until finally a vertex is reached, at which point $\mathbf{X}_t$ stops changing. We use $\tau_i$ for the first time that $\mathbf{X}_t$ hits a face of dimension $n - i$; then, $\mathbf{X}_{\tau_n} \in \{0, 1\}^n$.

The output of the algorithm corresponds to the collection of the variables assigned a value of true $T \subseteq \{1, \ldots, n\}$:

$$T = \{i : (\mathbf{X}_{\tau_n})_i = 0\},$$

whereas implicitly the collection of variables assigned a value of false are $\{i : (\mathbf{X}_{\tau_n})_i = 1\}$.

### 3.2 Analysis of the Algorithm.

Our goal is to analyze the expected value of the assignment produced by the Sticky Brownian Motion rounding algorithm. Similarly to previous work, we aim to give a lower bound on the probability that a fixed clause $C$ is satisfied. Unfortunately, the conformal mapping approach

[6] It is easy to see that $x_{-i} = 1 - x_i$ and $\mathbf{w}_{-i} = -\mathbf{w}_i$ for every $i = 1, \ldots, n$.

[7] We note that the Brownian rounding algorithm for Max-2SAT can be equivalently defined in $[-1, 1]^n$, however, this will incur some overhead in the notations which we would like to avoid.

described in Section 2 does not seem to be easily applicable to the extended Sticky Brownian Motion rounding described above for MAX-2SAT, because our calculations for MAX-CUT relied heavily on the symmetry of the starting point of the random walk. We propose a different method for analyzing the Brownian rounding algorithm that is based on partial differential equations and the maximum principle. We prove analytically the following theorem which gives a guarantee on the performance of the algorithm. We also note that numerical calculations show that the algorithm in fact achieves the better approximation ratio of 0.921 (see Section 5.1 for details).

THEOREM 3.1. *The Sticky Brownian Motion rounding algorithm for* MAX-2SAT *achieves an approximation of at least* 0.8749.

### 3.2.1 Analysis via Partial Differential Equations and Maximum Principle.

As mentioned above, our analysis focuses on the probability that a single clause $C$ with variables $\{z_i, z_j\}$ is satisfied. We assume the variables are not negated. This is without loss of generality as the algorithm and analysis are invariant to the sign of the variable in the clause.

For simplicity of notation we denote by $x$ the marginal value of $z_i$ and by $y$ the marginal value of $z_j$. Thus, $\mathbf{v}_i = x\mathbf{v}_0 + \sqrt{x - x^2}\mathbf{w}_i$ and $\mathbf{v}_j = y\mathbf{v}_0 + \sqrt{y - y^2}\mathbf{w}_j$. Projecting the random process $\{\mathbf{X}\}_{t\geq 0}$ on the $i$ and $j$ coordinates of the random process, we obtain a new process $\{\tilde{\mathbf{X}}_t\}_{t\geq 0}$ where $\tilde{\mathbf{X}}_0 = (x, y)$. Let $\tilde{\mathbf{W}} = \begin{pmatrix} 1 & \cos(\theta) \\ \cos(\theta) & 1 \end{pmatrix}$, where $\theta$ is the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$. Then $\tilde{\mathbf{X}}_t = \tilde{\mathbf{X}}_0 + \tilde{\mathbf{W}}^{1/2}\mathbf{B}_t$ for all $0 \leq t \leq \tau$, where $\tau = \inf\{t : \tilde{\mathbf{X}}_0 + \tilde{\mathbf{W}}^{1/2}\mathbf{B}_t \notin [0,1]^2\}$ is the first time the process hits the boundary of the square. After time $\tau$, the process $\tilde{\mathbf{X}}_t$ performs a one-dimensional standard Brownian motion on the first side of the square it has hit, until it hits a vertex at some time $\sigma$. After time $\sigma$ the process stays fixed. Almost surely $\sigma < \infty$, and, moreover, it is easy to show that $\mathbb{E}\sigma < \infty$. We say that $\{\tilde{\mathbf{X}}_t\}_{t\geq 0}$ is absorbed at $\tilde{\mathbf{X}}_\sigma \in \{0,1\}^2$.

OBSERVATION 3.1. *The probability that the algorithm satisfies the clause* $\{z_i, z_j\}$ *equals*

$$\Pr\left[\tilde{\mathbf{X}}_\sigma \textit{ is absorbed in } \{(0,0),(0,1),(1,0)\}\right].$$

We abuse notation slightly and denote $\tilde{\mathbf{X}}_t$ by $\mathbf{X}_t$ and $\tilde{\mathbf{W}}$ by $\mathbf{W}$ for the rest of the section which is aimed at analyzing the above probability. We also denote $\rho = \cos(\theta)$.

Our next step is fixing $\theta$ and analyzing the probability of satisfying the clause for all possible values of marginals $x$ and $y$. Indeed, for different $x$ and $y$ but the same $\theta$, the analysis only needs to consider the same random process with a different starting point. Observe that not all such $x, y$ are necessarily feasible for the SDP: we characterize which ones are feasible for a given $\theta$ in Lemma 3.3. But considering all $x, y$ allows us to handle the probability in Observation 3.1 analytically.

For any $0 \leq x \leq 1, 0 \leq y \leq 1$, let $u(x, y)$ denote the probability of starting the random walk at the point $(x, y)$ and ending at one of the corners $(0,0)$, $(0,1)$ or $(1,0)$. This captures the probability of a clause being satisfied when the walk begins with marginals $(x, y)$ (and angle $\theta$). We can easily calculate this probability exactly when either $x$ or $y$ are in the set $\{0,1\}$. We obtain the following easy lemma whose proof appears in the full version.

LEMMA 3.1. *For* $\phi(x, y) = 1 - xy$, *we have*

$$u(\mathbf{x}) = \phi(\mathbf{x}) \quad \textit{for all } \mathbf{x} \in \partial[0,1]^2. \tag{3.8}$$

*Moreover, for all* $\mathbf{x}$ *in the interior of the square* $[0,1]^2$, $u(\mathbf{x}) = \mathbb{E}^{\mathbf{x}}[\phi(\mathbf{X}_\tau)]$, *where* $\mathbb{E}^{\mathbf{x}}$ *denotes expectation with respect to starting the process at* $\mathbf{X}_0 = \mathbf{x}$.

Next we use the fact that Brownian motion gives a solution to the Dirichlet boundary problem. While Brownian motion gives a solution to Laplace's equation ([29] chapter 3), since our random process is a diffusion process, we need a slightly more general result[8]. We state the following result from [31], specialized to our setting, that basically states that given a diffusion process in $[0,1]^2$ and a function $\phi$ on the boundary, the extension of the function defined on the interior by the expected value of the function at the first hitting point on the boundary is characterized by an elliptic partial differential equation.

THEOREM 3.2. ([31] THEOREM 9.2.14) *Let* $D = (0,1)^2 \subseteq \mathbb{R}^2$, $\mathbf{\Sigma} \in \mathbb{R}^{2\times 2}$ *and let* $a_{11}, a_{12}, a_{21}, a_{22}$ *be defined as follows*

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \frac{1}{2}\mathbf{\Sigma}\mathbf{\Sigma}^\top.$$

*For any* $\mathbf{x} \in D$, *consider the process* $\mathbf{X}_t = \mathbf{X}_0 + \mathbf{\Sigma}\mathbf{B}_t$ *where* $\mathbf{B}_t$ *is standard Brownian motion in* $\mathbf{R}^2$. *Let* $\tau = \inf\{t : \mathbf{X}_t \notin D\}$. *Given a bounded continuous function* $\phi : \partial D \to \mathbb{R}$, *define the function* $u : D \to \mathbb{R}$ *such that*

$$u(\mathbf{x}) = \mathbb{E}^{\mathbf{x}}\left[\phi(\mathbf{X}_\tau)\right],$$

*where* $\mathbb{E}^{\mathbf{x}}$ *denotes the expected value when* $\mathbf{X}_0 = \mathbf{x} \in \mathbb{R}^2$. *I.e.,* $u(\mathbf{x})$ *is the expected value of* $\phi$ *when first hitting*

[8] This result can also be derived from Theorem 3.12 in [29] after applying a linear transformation to the variables.

*$\partial D$ conditioned on starting at point $\mathbf{x}$. Consider the uniformly elliptic partial differential operator $\mathcal{L}$ in $D$ defined by:*

$$\mathcal{L} = \sum_{i,j=1}^{2} a_{ij}\frac{\partial^2}{\partial x_i \partial x_j}.$$

*Then $u \in C^2(D)$ is the unique solution to the partial differential equation[9]:*

$$\begin{aligned} &\mathcal{L}u = 0 && \text{in } D \\ &\lim_{\substack{\mathbf{x}\to\mathbf{y}\\ \mathbf{x}\in D}} u(\mathbf{x}) = \phi(\mathbf{y}) && \text{for all } \mathbf{y} \in \partial D. \end{aligned}$$

We instantiate our differential equation by choosing $\mathbf{\Sigma} = \mathbf{W}^{1/2}$ and thus $a_{ij}$ are the entries of $\mathbf{W}$. It is important to note that all $a_{ij}$s are independent of the starting point $\mathbf{x} \in [0,1]^2$. Thus, we obtain that $u$ is the unique function satisfying the following partial differential equation:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 2\rho\frac{\partial^2 u}{\partial x \partial y} = 0 \qquad & \forall (x,y) \in \operatorname{Int}[0,1]^2 \\ u(x,y) = (1 - xy) \qquad & \forall (x,y) \in \partial[0,1]^2. \end{aligned}$$

Above, and in the rest of the paper, we use $\operatorname{Int} D$ to denote the interior of a set $D$, and $\partial D$ to denote its boundary.

It remains to solve the above partial differential equation (PDE) that will allow us to calculate $u(x,y)$ and give the probability of satisfying the clause.

**3.3 Maximum Principle.** Finding closed form solutions general PDE's is challenging and, there is no guarantee any solution would be expressible in terms of simple functions. However, to find a good approximation ratio, it suffices for us to find good lower-bounds on the probability of satisfying the clause. I.e. we need to give a lower bound on the function $u(x,y)$ from the previous section over those $(x,y)$ that are feasible. Since the PDE's generated by our algorithm are elliptic (a particular kind of PDE), we will use a property of elliptic PDE's which will allow us to produce good lower-bounds on the solution at any given point. More precisely, we use the following theorem from Gilbarg and Trudinger [20].

Let $\mathcal{L}$ denote the operator $\mathcal{L} := \sum_{ij} a_{ij}\frac{\partial^2}{\partial_i \partial_j}$, and we say that $\mathcal{L}$ is an elliptic operator if the coefficient matrix $A = [a_{ij}]_{i,j}$ is positive semi-definite.

We restate a version of Theorem 3.1 in Gilbarg and Trudinger [20] that shows how the maximum principle can be used to obtain lower bounds on $u(x,y)$. Here $\bar{D}$ denotes the closure of $D$.

Theorem 3.3. (Maximum Principle) *Let $\mathcal{L}$ be elliptic on a bounded domain $D$ and suppose $\mathcal{L}[g](x) \geq 0 \;\; \forall x \in D$ for some $g \in C^2(D) \cap C^0(\bar{D})$. Then the maximum of $g$ on $D$ is achieved on $\partial D$, that is,*

$$\sup_{x\in D} g(x) = \sup_{x\in\partial D} g(x).$$

Theorem 3.3 has the following corollary that allows us to obtain lower bounds on $u(x,y)$.

Corollary 3.1. *Let $\mathcal{L}$ be elliptic on a bounded domain $D$ and for some $u, g \in C^2(D) \cap C^0(\bar{D})$.*

1. *$\mathcal{L}[g](x) \geq \mathcal{L}[u](x) \quad \forall x \in D$*
2. *$g(x) \leq u(x) \quad \forall x \in \partial D$,*

*then $g(x) \leq u(x) \forall x \in D$.*

We refer the reader to [20] for a formal proof. Thus, it is enough to construct candidate functions $g : [0,1]^2 \to \mathbb{R}$ such that

$$\begin{aligned} \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} + 2\rho\frac{\partial^2 g}{\partial x \partial y} \geq 0 \qquad & \forall (x,y) \in \operatorname{Int}[0,1]^2 \\ g(x,y) \leq (1 - xy) \qquad & \forall (x,y) \in \partial[0,1]^2 \end{aligned}$$

Then we obtain that $g(x,y) \leq u(x,y)$ for all $(x,y) \in [0,1]^2$. In what follows we construct many different such function each of which works for a different range of the parameter $\theta$ (equivalently, $\rho$).

**3.4 Candidate Functions for Maximum Principle.** We now construct feasible candidates to the maximum principle as described in Corollary 3.1. We define the following functions:

1. $g_1(x,y) = 1 - xy - \cos(\theta)\sqrt{x - x^2}\sqrt{y - y^2}$.
2. $g_2(x,y) = 1 - xy - 2\cos(\theta)(x - x^2)(y - y^2)$.
3. $g_3(x,y) = 1 - xy - \frac{1}{2}(1 + 5\cos(\theta))(x - x^2)(y - y^2)(x + y)(2 - x - y)$.

The following lemma shows that the above functions satisfy the conditions required for the application of the maximum principle (its proof is available in the full version).

Lemma 3.2. *Each of $g_1, g_2, g_3$ satisfies the boundary conditions, i.e. $g_i(x,y) = u(x,y)$ for all $x,y \in \partial[0,1]^2$ and for all values $\theta$. Moreover, we have the following for each $(x,y) \in [0,1]^2$:*

1. *If $1 \geq \cos(\theta) \geq 0$, then $\mathcal{L}g_1 \geq 0$.*
2. *If $0 \geq \cos(\theta) \geq -\frac{1}{2}$, then $\mathcal{L}g_2 \geq 0$.*
3. *If $-\frac{1}{2} \geq \cos(\theta) \geq -1$, then $\mathcal{L}g_3 \geq 0$.*

[9] $u \in C^k(D)$ means that $u$ has a continuous $k^{\text{th}}$ derivative over $D$, and $u \in C^0(D)$ means that $u$ is continuous.

While some of these proofs are based on simple inequalities, proving others requires us to use sum of squares expressions. For example, to show $\mathcal{L}g_3 \geq 0$, we consider $\mathcal{L}g_3 = p(x, y, \cos(\theta))$ as a polynomial in $x, y$ and $\cos(\theta)$. Replacing $z = \cos(\theta)$, our aim is to show $p(x, y, z) \geq 0$ if $0 \leq x, y \leq 1$ and $-1 \leq z \leq -\frac{1}{2}$. Equivalently, we need to show $p(x, y, z) \geq 0$ whenever $r_1(x, y, z) := x - x^2 \geq 0$, $r_2(x, y, z) := y - y^2 \geq 0$ and $r_3(x, y, z) := -(z + \frac{1}{2}) \geq 0$ and $r_4(x, y, z) := (z + 1) \geq 0$. We show this by obtaining polynomials $q_i(x, y, z)$ for $i = 0, 1, 2, 3, 4$ such that each $q_i$ is a sum of squares polynomial of fixed degree and we have

$$p(x, y, z) = q_0(x, y, z) + \sum_{i=1}^{4} q_i(x, y, z) r_i(x, y, z).$$

Observe that the above polynomial equality proves the desired result by evaluating the RHS for every $0 \leq x, y \leq 1$ and $-1/2 \geq z \geq -1$. Clearly, the RHS is non-negative: each $q_i$ is non-negative since it is a sum of squares and each $r_i$ is non-negative in the region we care about, by construction. We mention that we obtain these proofs via solving a semi-definite program of fixed degree (at most 6) for each of the $q_i$ polynomials (missing details are available in the full version).

Let us now focus on the approximation guarantee that can be proved using the above functions $g_1$, $g_2$, and $g_3$. The following lemma compares the lower bounds on the probability of satisfying a clause, as given by $g_1$, $g_2$, and $g_3$, to the SDP objective. Recall that the contribution of any clause with marginals $x$ and $y$ and angle $\theta$ to the SDP's objective is given by: $1 - xy - \cos(\theta)\sqrt{x - x^2}\sqrt{y - y^2}$. We denote this contribution by $\mathrm{SDP}(x, y, \theta)$. It is important to note that not all triples $(x, y, \theta)$ are feasible (recall that $\theta$ is the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$), due to the triangle inequalities in the SDP. This is summarized in the following lemma.

LEMMA 3.3. *Let $x, y, \theta$ be as defined by a feasible pair of vectors $v_i$ and $v_j$. Then they must satisfy the following constraints:*

1. $0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq \theta \leq \pi$.
2. $\cos(\theta) \geq -\sqrt{\frac{xy}{(1-x)(1-y)}}$.
3. $\cos(\theta) \geq -\sqrt{\frac{(1-x)(1-y)}{xy}}$.

Finally, we prove the following lemma which proves an approximation guarantee of 0.8749 for MAX-2SAT via the PDE and the maximum principle approach. As before, these proofs rely on explicitly obtaining sum of squares proofs as discussed above. We remark that these proofs essentially aim to obtain $\frac{7}{8} = 0.875$-approximation but errors of the order $10^{-5}$ allow us to obtain a slightly worse bound using this methods. The details appear in the full version.

LEMMA 3.4. *Consider any feasible triple $(x, y, \theta)$ satisfying the condition in Lemma 3.3. We have the following.*

1. *If* $1 \geq \cos(\theta) \geq 0$, *then* $g_1(x, y) \geq 1 \cdot \mathrm{SDP}(x, y, \theta)$.
2. *If* $0 \geq \cos(\theta) \geq -\frac{1}{2}$, *then* $g_2(x, y) \geq 0.8749 \cdot \mathrm{SDP}(x, y, \theta)$.
3. *If* $-\frac{1}{2} \geq \cos(\theta) \geq -1$, *then* $g_3(x, y) \geq 0.8749 \cdot \mathrm{SDP}(x, y, \theta)$.

## 4 Max-Cut with Side Constraints (Max-Cut-SC)

In this section we describe how to apply the Sticky Brownian Motion rounding and the framework of Raghavendra and Tan [34] to the MAX-CUT-SC problem in order to give a bi-criteria approximation algorithm whose running time is non-trivial even when the the number of constraints is large.

**4.1 Problem Definition and Basics.** Let us recall the relevant notation and definitions. An instance of the MAX-CUT-SC problem is given by an $n$-vertex graph $G = (V, E)$ with edge weights $a : E \to \mathbb{R}_+$, as well as a collection $\mathcal{F} = \{F_1, \ldots, F_k\}$ of subsets of $V$, and cardinality bounds $b_1, \ldots, b_k \in \mathbb{N}$. For ease of notation, we will assume that $V = \{1, \ldots, n\}$. Moreover, we denote the total edge weight by $a(E) = \sum_{e \in E} a(e)$. The goal in the MAX-CUT-SC problem is to find a subset $S \subset V$ that maximizes the weight $a(\delta(S))$ of edges crossing the cut $(S, V \setminus S)$, subject to having $|S \cap F_i| = b_i$ for all $i \in [k]$. These cardinality constraints may not be simultaneously satisfiable, and moreover, when $k$ grows with $n$, checking satisfiability is NP-hard [17]. For these reasons, we allow for approximately feasible solutions. We will say that a set of vertices $S \subseteq V$ is an $(\alpha, \varepsilon)$-approximation to the MAX-CUT-SC problem if $\big||S \cap F_i| - b_i\big| \leq \varepsilon n$ for all $i \in [k]$, and $a(\delta(S)) \geq \alpha \cdot a(\delta(T))$ for all $T \subset V$ such that $|T \cap F_i| = b_i$ for all $i \in [k]$. In the remainder of this section we assume that the instance given by $G$, $\mathcal{F}$, and $b$ is satisfiable, i.e. that there exists a set of vertices $T$ such that $|T \cap F_i| = b_i$ for all $i \in [k]$. Our algorithm may fail if this assumption is not satisfied. If this happens, then the algorithm will certify that the instance was not satisfiable.

We start with a simple baseline approximation algorithm, based on independent rounding. The algorithm outputs an approximately feasible solution which cuts a constant fraction of the total edge weight. For this reason, it achieves a good bi-criteria approximation when the value of the optimal solution OPT is much smaller than $\varepsilon a(E)$. This allows us to focus on the case in which OPT is bigger than $\varepsilon a(E)$ for our main rounding algorithm. The proof of the lemma, which follows from standard arguments, is available in the full version.

LEMMA 4.1. *Suppose that* $n \geq \frac{2\ln(8k/\varepsilon)}{\varepsilon^2}$ *and* $\varepsilon \leq \frac{1}{2}$. *There exists a polynomial time algorithm that on input a satisfiable instance* $G = (V, E)$, $\mathcal{F}$, *and* $b_1, \ldots, b_k$, *as defined above, outputs a set* $S \subseteq V$ *such that, with high probability,* $a(\delta(S)) \geq \frac{\varepsilon}{2}a(E)$, *and* $\big||S \cap F_i| - b_i\big| \leq \varepsilon n$ *for all* $i \in [k]$.

**4.2 Sum of Squares Relaxation.** Our main approximation algorithm is based on a semidefinite relaxation, and the sticky Brownian motion. Let us suppose that we are given the optimal objective value OPT of a feasible solution: this assumption can be removed by doing binary search for OPT. We can then model the problem of finding an optimal feasible solution by the quadratic program

$$\begin{aligned} &\sum_{e=(i,j)\in E} a(e)(x_i - x_j)^2 \geq \mathrm{OPT} && \\ \text{s.t.}\quad &\sum_{j\in F_i} x_j = b_i && \forall i = 1, \ldots, k \\ &x_j(1 - x_j) = 0 && \forall j = 1, \ldots, k \end{aligned}$$

Let us denote this quadratic feasibility problem by $Q$. The Sum of Squares (Lasserre) hierarchy gives a semidefinite program that relaxes $Q$. We denote by $\mathrm{SoS}_\ell(Q)$ the solutions to the level-$\ell$ Sum of Squares relaxations of $Q$. Any solution in $\mathrm{SoS}_\ell(Q)$ can be represented as a collection of vectors $\mathcal{V} = \{\mathbf{v}_S : S \subseteq [n], 0 \leq |S| \leq \ell\}$. To avoid overly cluttered notation, we write $\mathbf{v}_i$ for $\mathbf{v}_{\{i\}}$; we also write $\mathbf{v}_0$ for $\mathbf{v}_\emptyset$. We need the following properties of $\mathcal{V}$, valid as long as $\ell \geq 2$.

1. $\mathbf{v}_0 \cdot \mathbf{v}_0 = 1$.

2. $\mathbf{v}_S \cdot \mathbf{v}_T = \mathbf{v}_{S'} \cdot \mathbf{v}_{T'}$ for any $S, S', T, T'$ such that $S \cup T = S' \cup T'$ and $|S \cup T| \leq k$. In particular, $\mathbf{v}_i \cdot \mathbf{v}_i = \mathbf{v}_i \cdot \mathbf{v}_0$ for any $i$.

3. For any $i$ and $j$ the following inequalities hold:

$$1 \geq \mathbf{v}_0 \cdot \mathbf{v}_i + \mathbf{v}_j \cdot \mathbf{v}_0 - \mathbf{v}_i \cdot \mathbf{v}_j \tag{4.9}$$

$$\mathbf{v}_i \cdot \mathbf{v}_0 \geq \mathbf{v}_i \cdot \mathbf{v}_j \tag{4.10}$$

$$\mathbf{v}_i \cdot \mathbf{v}_j \geq 0 \tag{4.11}$$

4. $\sum_{e=(i,j)\in E} a(e)\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \mathrm{OPT}$

5. For any $i$, there exist two solutions $\mathcal{V}^{i\to 0}$ and $\mathcal{V}^{i\to 1}$ in $\mathrm{SoS}_{\ell-1}(Q)$ such that, if we denote the vectors in $\mathcal{V}^{i\to 0}$ by $\mathbf{v}_S^0$, and the vectors in $\mathcal{V}^{i\to 1}$ by $\mathbf{v}_S^1$, we have

$$\mathbf{v}_S \cdot \mathbf{v}_0 = (1 - \mathbf{v}_i \cdot \mathbf{v}_0)\mathbf{v}_S^0 \cdot \mathbf{v}_0^0 + (\mathbf{v}_i \cdot \mathbf{v}_0)\mathbf{v}_S^1 \cdot \mathbf{v}_0^1.$$

Moreover, a solution $\mathcal{V} \in \mathrm{SoS}_\ell$ can be computed in time polynomial in $n^\ell$.

Intuitively, we think of $\mathcal{V}$ as describing a pseudo-distribution over solutions to $Q$, and we interpret $\mathbf{v}_S \cdot \mathbf{v}_T$ as the pseudo-probability that all variables in $S \cup T$ are set to one, or, equivalently, as the pseudo-expectation of $\prod_{i\in S\cup T} x_i$. Usually we cannot expect that there is any true distribution giving these probabilities. Nevertheless, the pseudo-probabilities and pseudo-expectations satisfy some of the properties of actual probabilities. For example, the transformation from $\mathcal{V}$ to $\mathcal{V}^{i\to b}$ corresponds to *conditioning* $x_i$ to $b$.

We will denote by $x_S = \mathbf{v}_S \cdot \mathbf{v}_0$ the marginal value of set $S$. In particular, we will work with the single-variable marginals $x_i = x_{\{i\}} = \mathbf{v}_i \cdot \mathbf{v}_0$, and will denote $\mathbf{x} = (x_1, \ldots, x_n)$. As before, it will be convenient to work with the component of $\mathbf{v}_i$ which is orthogonal to $\mathbf{v}_0$. We define $\widetilde{\mathbf{w}}_i = \mathbf{v}_i - x_i\mathbf{v}_0$, and $\mathbf{w}_i = \frac{1}{\|\widetilde{\mathbf{w}}_i\|}\widetilde{\mathbf{w}}_i$. Note that, by the Pythagorean theorem, $\|\widetilde{\mathbf{w}}_i\|^2 = x_i - x_i^2$, and $\mathbf{v}_i = x_i\mathbf{v}_0 + \sqrt{x_i - x_i^2}\mathbf{w}_i$. We define the matrices $\widetilde{\mathbf{W}}$ and $\mathbf{W}$ by $\widetilde{\mathbf{W}}_{i,j} := \widetilde{\mathbf{w}}_i \cdot \widetilde{\mathbf{w}}_j$ and $\mathbf{W}_{i,j} := \mathbf{w}_i \cdot \mathbf{w}_j$. We can think of $\widetilde{\mathbf{W}}$ as the covariance matrix of the pseudodistribution corresponding to the SDP solution. The following lemma, due to Barak, Raghavendra, and Steurer [13], and, independently, to Guruswami and Sinop [24], shows that any pseudodistribution can be conditioned so that the covariances are small on average.

LEMMA 4.2. *For any* $\varepsilon_0 \leq 1$, *and any* $\mathcal{V} \in \mathrm{SoS}_\ell(Q)$, *where* $\ell \geq \frac{1}{\varepsilon_0^4} + 2$, *there exists an efficiently computable* $\mathcal{V}' \in \mathrm{SoS}_{\ell-1/\varepsilon_0^4}(Q)$, *such that*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \widetilde{\mathbf{W}}_{i,j}^2 \leq \varepsilon_0^4 n^2. \tag{4.12}$$

*In particular,* $\mathcal{V}'$ *can be computed by conditioning* $\mathcal{V}$ *on* $\frac{1}{\varepsilon_0^4}$ *variables.*

**4.3 Rounding Algorithm.** For our algorithm, we first solve a semidefinite program to compute a solution in $\mathrm{SoS}_\ell(Q)$, to which we apply Lemma 4.2 with parameter $\varepsilon_0$, which we will choose later. In order to be able to apply the lemma, we choose $\ell = \left\lceil \frac{1}{\varepsilon_0^4} \right\rceil + 2$. The rounding algorithm itself is similar to the one we used for MAX-2SAT. We perform a Sticky Brownian Motion with initial covariance $\mathbf{W}$, starting at the initial point $\mathbf{X}_0 = \mathbf{x}$, i.e. at the marginals given by the SDP solution. As variables hit 0 or 1, we freeze them, and delete the corresponding row and column of the covariance matrix. The main difference from the Max-2SAT rounding is that we stop the process at time $\tau$, where $\tau$ is another parameter that we will choose later. Then, independently for each $i = 1, \dots, n$, we include vertex $i$ in the final solution $S$ with probability $(\mathbf{X}_\tau)_i$, and output $S$.

The key property of this rounding that allows us to handle a large number of global constraints is that, for any $F_i \in \mathcal{F}$, the value $\sum_{j \in F_i} (\mathbf{X}_\tau)_j$ that the fractional solution assigns to the set $F_i$ satisfies a sub-Gaussian concentration bound around $b_i$. Note that $\sum_{j \in F_i} (\mathbf{X}_t)_j$ is a martingale with expectation equal to $b_i$. Moreover, by Lemma 4.2, the entries of the covariance matrix $\widetilde{\mathbf{W}}$ are small on average, which allows us to also bound the entries of the covariance matrix $\mathbf{W}$, and, as a consequence, bound how fast the variance of the martingale increases with time. The reason we stop the walk at time $\tau$ is to make sure the variance doesn't grow too large: this freedom, allowed by the Sticky Brownian Motion rounding, is important for our analysis. The variance bound then implies the sub-Gaussian concentration of $\sum_{j \in F_i} (\mathbf{X}_\tau)_j$ around its mean $b_i$, and using this concentration we can show that no constraint is violated by too much. This argument crucially uses the fact that our rounding is a random walk with small increments, and we do not expect similarly strong concentration results for the random hyperplane rounding or its variants.

The analysis of the objective function, as usual, reduces to analyzing the probability that we cut an edge. However, because we start the Sticky Brownian Motion at $\mathbf{x}$, which may not be equal to $\mathbf{0}$, our analysis from Section 2 is not sufficient. Instead, we use the PDE based analysis from Section 3, which easily extends to the Max-Cut objective. One detail to take care of is that, because we stop the walk early, edges incident on vertices that have not reached 0 or 1 by time $\tau$ may be cut with much smaller probability than their contribution to the SDP objective. To deal with this, we choose the time $\tau$ when we stop the walk large enough, so that any vertex has probability at least $1 - \mathrm{poly}(\varepsilon)$ to have reached $\{0, 1\}$ by time $\tau$. We show that this happens for $\tau = \Theta(\log(1/\varepsilon))$. This value of $\tau$ is small enough so that we can usefully bound the variance of $\sum_{j \in F_i} (\mathbf{X}_\tau)_i$ and prove the sub-Gaussian concentration we mentioned above.

Let us recall some notation that will be useful in our analysis. We will use $\tau_i$ for the first time $t$ that $\mathbf{X}_t$ hits a face of $[0,1]^n$ of dimension $n - i$; then, $\mathbf{X}_{\tau_n} \in \{0,1\}^n$. We also use $\mathbf{W}_t$ for the covariance used at time step $t$, which is equal to $\mathbf{W}$ with rows and columns indexed by $\{i : (\mathbf{X}_t)_i \in \{0,1\}\}$ zeroed out.

As discussed, our analysis relies on a martingale concentration inequality, and the following lemma, which is proved with the methods we used above for the Max-2SAT problem. A proof sketch can be found in the full version.

Lemma 4.3. *For the SDP solution $\mathcal{V}$ and the Sticky Brownian Motion $\mathbf{X}_t$ described above, and for any pair $\{i, j\}$ of vertices*

$$\Pr[(\mathbf{X}_{\tau_n})_i \neq (\mathbf{X}_{\tau_n})_j] \geq 0.843 \cdot \|\mathbf{v}_i - \mathbf{v}_j\|^2.$$

The next lemma shows that the probability that any coordinate is fixed by time $t$ drops exponentially with $t$. We use this fact to argue that by time $\tau = \Theta(\log(1/\varepsilon))$ the endpoints of any edge have probability at least $1 - \mathrm{poly}(\varepsilon)$ to be fixed, and, therefore, edges are cut with approximately the same probability as if we didn't stop the random walk early, which allows us to use Lemma 4.3. The proof of this lemma, which is likely well-known, appears in the full version.

Lemma 4.4. *For any $i$, and any integer $t \geq 0$, $\Pr[\forall s \leq t : 0 < (\mathbf{X}_s)_i < 1] < 4^{-t}$.*

The following concentration inequality is our other key lemma. The statement is complicated by the technical issue that the concentration properties of the random walk depend on the covariance matrix $\mathbf{W}$, while Lemma 4.2 bounds the entries of $\widetilde{\mathbf{W}}$. When $x_i(1-x_i)$ or $x_j(1-x_j)$ is small, $\widetilde{\mathbf{W}}_{i,j}$ can be much smaller than $\mathbf{W}_{i,j}$. Because of this, we only prove our concentration bound for sets of vertices $i$ for which $x_i(1-x_i)$ is sufficiently large. For those $i$ for which $x_i(1-x_i)$ is small, we will instead use the fact that such $x_i$ are already nearly integral to prove a simpler concentration bound.

Lemma 4.5. *Let $\varepsilon_0, \varepsilon_1 \in [0,1]$, and $n \geq \frac{\varepsilon_1}{8\tau\varepsilon_0^2}$. Define $V_{>\varepsilon_1} = \{i : 2x_i(1-x_i) > \varepsilon_1\}$. For any set $F \subseteq V_{>\varepsilon_1}$, and any $t \geq 0$, the random set $S$ output by the rounding algorithm satisfies*

$$\Pr\left[\left||F \cap S| - \sum_{i \in F} x_i\right| \geq t\varepsilon_0 n\right] \leq 4\exp\left(-\frac{\varepsilon_1 t^2}{4\tau}\right).$$

We defer the proofs of Lemma 4.5 and Theorem 1.3 to the full version of our paper.

## 5 Extensions of the Brownian Rounding

In this section, we consider two extensions of the Brownian rounding algorithm. We also present numerical results for these variants showing improved performance over the sticky Brownian Rounding analyzed in previous sections.

**5.1 Brownian Rounding with Slowdown.** As noted in section 2, the Sticky Brownian rounding algorithm does not achieve the optimal value for the MAX-CUT problem. A natural question is to ask if we can modify the algorithm to achieve the optimal constant. In this section, we will show that a simple modification achieves this ratio up to at least three decimals. Our results are computer-assisted as we solve partial differential equations using finite element methods. These improvement indicate that variants of the Brownian Rounding approach offer a direction to obtain optimal SDP rounding algorithms for MAX-CUT problem as well as other CSP problems.

In the sticky Brownian motion, the covariance matrix $W_t$ is a constant, until some vertex's marginals $(\mathbf{X}_t)_i$ becomes ±1. At that point, we abruptly zero the $i^{th}$ row and column. In this section, we analyze the algorithm where we gradually dampen the step size of the Brownian motion as it approaches the boundary of the hypercube, until it becomes 0 at the boundary. We call this process a "Sticky Brownian Motion with Slowdown."

Let $(\mathbf{X}_t)_i$ denote the marginal value of vertex $i$ at time $t$. Initially $(\mathbf{X}_0)_i = 0$. First, we describe the discrete algorithm which will provide intuition but will also be useful to those uncomfortable with Brownian motion and diffusion processes. At each time step, we will take a step whose length is scaled by a factor of $(1-(\mathbf{X}_t)_i^2)^\alpha$ for some constant $\alpha$. In particular, the marginals will evolve according to the equation:

$$(\mathbf{X}_{t+dt})_i = (\mathbf{X}_t)_i + (1-(\mathbf{X}_t)_i)^2)^{\alpha/2}\cdot\left(\mathbf{w}_i\cdot\mathbf{G}_t\right)\cdot\sqrt{dt}. \tag{5.13}$$

where $\mathbf{G}_t$ is distributed according to an $n$-dimensional Gaussian and $dt$ is a small discrete step by which we advance the time variable. When $\mathbf{X}_t$ is sufficiently close to $-1$ or $+1$, we round it to the nearest one of the two: from then on it will stay fixed because of the definition of the process, i.e. we will have $(\mathbf{X}_s)_i = (\mathbf{X}_t)_i$ for all $s > t$.

More formally, $\mathbf{X}_t$ is defined as an Itô diffusion process which satisfies the stochastic differential equation

$$d\mathbf{X}_t = \mathbf{A}(\mathbf{X}_t)\cdot\mathbf{W}^{1/2}\cdot d\mathbf{B}_t \tag{5.14}$$

where $\mathbf{B}_t$ is the standard Brownian motion in $\mathbb{R}^n$ and $\mathbf{A}(\mathbf{X}_t)$ is the diagonal matrix with entries $[\mathbf{A}(\mathbf{X}_t)]_{ii} = (1-(\mathbf{X}_t)_i^2)^{\alpha/2}$. Since this process is continuous, it becomes naturally sticky when some coordinate $(\mathbf{X}_t)_i$ reaches $\{-1,1\}$.

Once again, it suffices to restrict our attention to the two dimensional case where we analyze the probability of cutting an edge $(i,j)$ and we will assume that $\tilde{\mathbf{W}} = \begin{pmatrix} 1 & \cos(\theta) \\ \cos(\theta) & 1 \end{pmatrix}$, where $\theta$ is the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$.

Let $\tau$ be the first time when $\mathbf{X}_t$ hits the boundary $\partial[-1,1]^2$. Since the walk slows down as it approaches the boundary, it is worth asking if $\mathbb{E}[\tau]$ is finite. In the full version, we show that $\mathbb{E}[\tau]$ is finite for constant $\alpha$.

Let $u(x,y)$ denote the probability of the Sticky Brownian Walk algorithm starting at $(x,y)$ cutting an edge, i.e. the walk is absorbed in either $(+1,-1)$ or $(-1,+1)$. It is easy to give a precise formula for $u$ at the boundary as the algorithm simplifies to a one-dimensional walk. Thus, $u(x,y)$ satisfies the boundary condition $\phi(x,y) = (1-xy)/2$ for all points $(x,y) \in bd[-1,1]^2$. For a given $(x,y) \in \text{Int}[-1,1]^2$, we can say

$$u(x,y) = \mathbb{E}^{(x,y)}[\phi(\tilde{\mathbf{X}}_\tau(i), \tilde{\mathbf{X}}_\tau(j))],$$

where $\mathbb{E}^{(x,y)}$ denotes the expectation of diffusion process that begins at $(x,y)$. Informally, $u(x,y)$ is the expected value of $\phi$ when first hitting $\partial[-1,1]^2$ conditioned on starting at point $(x,y)$. Observe that the probability that the algorithm will cut an edge is given by $u(0,0)$.

The key fact about $u(x,y)$ that we use is that it is the unique solution to a Dirichlet Problem, formalized in Lemma 5.1 below.

LEMMA 5.1. *Let $\mathcal{L}^\alpha$ denote the operator*

$$\begin{aligned}\mathcal{L}^\alpha =& (1-x^2)^\alpha\frac{\partial^2}{\partial x^2} \\ &+ 2\cos(\theta)(1-x^2)^{\alpha/2}(1-y^2)^{\alpha/2}\frac{\partial^2}{\partial x\partial y} \\ &+ (1-y^2)^\alpha\frac{\partial^2}{\partial y^2},\end{aligned}$$

*then the function $u(x,y)$ is the unique solution to the Dirichlet Problem:*

$$\begin{aligned}\mathcal{L}^\alpha[u](x,y) &= 0 && \forall (x,y)\in\text{Int}([-1,1]^2) \\ \lim_{\substack{(x,y)\to(\tilde{x},\tilde{y}),\\ (x,y)\in\text{Int}([-1,1]^2)}} u(x,y) &= \phi(\tilde{x},\tilde{y}) && \forall(\tilde{x},\tilde{y})\in\partial[-1,1]^2.\end{aligned}$$

The proof again uses [31, Theorem 9.2.14], however, the exact application is a little subtle and we defer the details to the full version.

**Numerical Results.** The Dirichlet problem is parameterized by two variables: the slowdown parameter $\alpha$ and the angle between the vectors $\theta$. We can numerically solve the above equation using existing solvers for any given fixed $\alpha$ and angle $\theta \in [0,\pi]$. We solve these problems for a variety of $\alpha$ between 0 and 2 and all values of $\theta$ in $[0,\pi]$ discretized to a granularity of 0.02.[10]

[10] Our code, containing the details of the implementation, is available at [1].

We observe that as we increase $\alpha$ from 0 to 2, the approximation ratio peaks around $\alpha \approx 1.61$ for all values of $\theta$. In particular, when $\alpha = 1.61$, the approximation ratio is 0.878 which matches the integrality gap for this relaxation up to three decimal points.

The Brownian rounding with slowdown is a well-defined algorithm for any 2-CSP. We investigate 3 different values of slowdown parameter, i.e., $\alpha$, and show their relative approximation ratios. We show that with a slowdown of 1.61 we achieve an approximation ratio of 0.929 for MAX-2SAT. We list these values below in Table 2.

For the MAX-CUT problem, since we start the walk at the point $(0, 0)$, we only need to investigate the performance of the rounding for all possible angles between two unit vectors which range in $[0, \theta]$ (Figure 2). In particular, we are able to achieve values that are comparable to the Goemans-Williamson bound.

| $\alpha$ | MAX-CUT | MAX-2SAT |
|---|---|---|
| 0 | 0.861 | 0.921 |
| 1 | 0.874 | 0.927 |
| 1.61 | 0.878 | 0.929 |

Table 2: Approximation ratio of Sticky Brownian Motion rounding with Slowdown for MAX-CUT and MAX-2SAT.

### 5.2 Higher-Dimension Brownian Rounding.

Our motivating example for considering the higher-dimension Brownian rounding is the MAX-DICUT problem: given a directed graph $G = (V, E)$ equipped with non-negative edge weights $a : E \to \mathbb{R}_+$ we wish to find a cut $S \subseteq V$ that maximizes the total weight of edges going out of $S$. The standard semi-definite relaxation for MAX-DICUT is the following:

$$\begin{aligned} \max \quad & \sum_{e=(i\to j)\in E} a_e \cdot \frac{(\mathbf{w}_0 + \mathbf{w}_i)\cdot(\mathbf{w}_0 - \mathbf{w}_j)}{4} \\ s.t. \quad & \mathbf{w}_i \cdot \mathbf{w}_i = 1 \qquad\qquad \forall i = 0, 1, ..., n \\ & \|\mathbf{w}_i - \mathbf{w}_j\|^2 + \|\mathbf{w}_j - \mathbf{w}_k\|^2 \geq \|\mathbf{w}_i - \mathbf{w}_k\|^2 \\ & \qquad\qquad\qquad \forall i, j, k = 0, 1, ..., n. \end{aligned}$$

In the above, the unit vector $\mathbf{w}_0$ denotes the cut $S$, whereas $-\mathbf{w}_0$ denotes $\overline{S}$. We also include the triangle inequalities which are valid for any valid relaxation.

The sticky Brownian rounding algorithm for MAX-DICUT fails to give a good performance guarantee. Thus we design a high-dimensional variant of the algorithm that incorporates the inherent asymmetry of the problem. Let us now describe the high-dimensional Brownian rounding algorithm. It is similar to the original Brownian rounding algorithm given for MAX-CUT, except that it evolves in $\mathbb{R}^{n+1}$ with one additional dimension for $\mathbf{w}_0$. Let $\mathbf{W} \in \mathbb{R}^{(n+1)\times(n+1)}$ denote the positive semi-definite correlation matrix defined by the vectors $\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_n$, *i.e.*, for every $0 \leq i, j \leq n$ we have that: $\mathbf{W}_{i,j} = \mathbf{w}_i \cdot \mathbf{w}_j$. The algorithm starts at the origin and perform a sticky Brownian motion inside the $[\pm 1]^{n+1}$ hypercube whose correlations are governed by $\mathbf{W}$.

As before, we achieve this by defining a random process $\{\mathbf{X}_t\}_{t\geq 0}$ as $\mathbf{X}_t = \mathbf{W}^{1/2}\mathbf{B}_t$, where $\{\mathbf{B}_t\}_{t\geq 0}$ is the standard Brownian motion in $\mathbb{R}^{n+1}$ starting at the origin and $\mathbf{W}^{1/2}$ is the square root matrix of $\mathbf{W}$. Additionally, we force $\{\mathbf{X}_t\}_{t\geq 0}$ to stick to the boundary of the $[\pm 1]^{n+1}$ hypercube, *i.e.*, once a coordinate of $\mathbf{X}_t$ equals either 1 or $-1$ it is fixed and remains unchanged indefinitely. This description can be formalized the same way we did for the MAX-CUT problem. Below we use $\sigma$ for the time at which $\mathbf{X}_\sigma \in \{-1, 1\}^{n+1}$, which has finite expectation.

Unlike the Brownian rounding algorithm for MAX-CUT, we need to take into consideration the value $\mathbf{w}_0$ was rounded to, *i.e.*, $(\mathbf{X}_\sigma)_0$, since the zero coordinate indicates $S$. Formally, the output $S \subseteq V$ is defined as follows:

$$S = \{i \in V : (\mathbf{X}_\sigma)_i = (\mathbf{X}_\sigma)_0\}.$$

To simplify the rest of the presentation, let us denote $Z_i := (\mathbf{X}_\sigma)_i$ for every $i = 0, 1, \ldots, n$.

The event that an edge $(i \to j) \in E$ is an outgoing edge from $S$, *i.e.*, $i \in S$ and $j \in \overline{S}$, involves three random variables: $Z_i$, $Z_j$, and $Z_0$. Formally, the above event happens if and only if $Z_i = Z_0$ and $Z_j \neq Z_0$. We now show how events on any triplet of the random variables $Z_0, Z_1, \ldots, Z_n$ can be precisely calculated. To simplify the presentation, denote the following for every $i, j, k = 0, 1, 2, \ldots, n$ and $\alpha, \beta, \gamma \in \{\pm 1\}$:

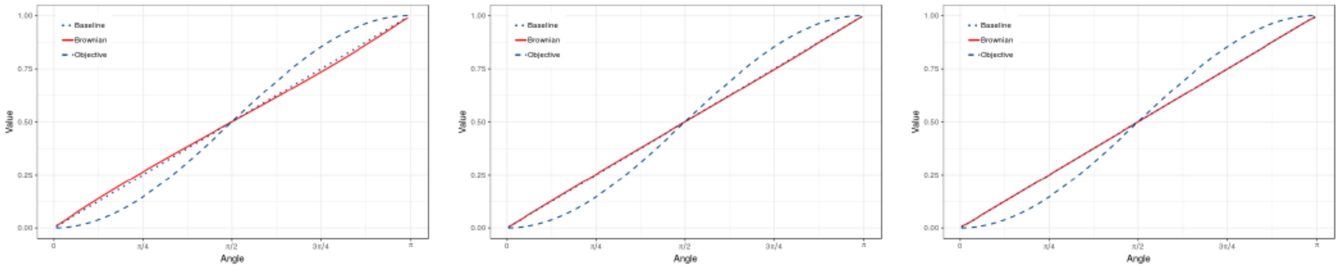
$$\begin{aligned} p_i(\alpha) &\triangleq \Pr[Z_i = \alpha] \\ p_{ij}(\alpha, \beta) &\triangleq \Pr[Z_i = \alpha, Z_j = \beta] \\ p_{ijk}(\alpha, \beta, \gamma) &\triangleq \Pr[Z_i = \alpha, Z_j = \beta, Z_k = \gamma]. \end{aligned}$$

OBSERVATION 5.1. *The following two hold:*

1. $p_i(\alpha) = p_i(-\alpha)$, $p_{ij}(\alpha, \beta) = p_{ij}(-\alpha, -\beta)$, *and* $p_{ijk}(\alpha, \beta, \gamma) = p_{ijk}(-\alpha, -\beta, -\gamma)$ *for every* $i, j, k = 0, 1, 2, \ldots, n$ *and* $\alpha, \beta, \gamma \in \{\pm 1\}$.
2. $p_i(\alpha) = 1/2$ *for every* $i = 0, 1, 2, \ldots, n$ *and* $\alpha \in \{\pm 1\}$.

The proof of Observation 5.1 follows immediately from symmetry.

The following lemma proves that every conjunction event that depends on three variables from



(a) $\alpha = 0$.

(b) $\alpha = 1$.

(c) $\alpha = 1.61$.

Figure 2: Comparing the performance of three values of the slowdown parameter for the MAX-CUT problem.

$Z_0, Z_1, Z_2, \dots, Z_n$ can be precisely calculated. (The proof is available in the full version.)

LEMMA 5.2. *For every* $i, j, k = 0, 1, 2, \dots, n$ *and* $\alpha, \beta, \gamma \in \{\pm 1\}$*:*

$$p_{ijk(\alpha,\beta,\gamma)} = \frac{1}{2}\left[p_{ij}(\alpha,\beta) + p_{ik}(\alpha,\gamma) + p_{jk}(\beta,\gamma) - \frac{1}{2}\right].$$

Let us now consider the case study problem MAX-DICUT. One can verify that an edge $(i \to j) \in E$ is a forward edge crossing the cut $S$ if and only if the following event happens: $\{Z_i = Z_0 \neq Z_j\}$ (recall that $Z_0$ indicates $S$). Thus, the value of the Brownian rounding algorithm, when considering only the edge $(i \to j)$, equals $p_{0ij}(1,1,-1) + p_{0ij}(-1,-1,1)$. Lemma 5.2 above shows that if one knows the values of $p_{ij}(\alpha,\beta)$ for every $i, j = 0, 1 \dots, n$ and $\alpha, \beta \in \{\pm 1\}$, then $p_{0ij}(1,1,-1)$ and $p_{0ij}(-1,-1,1)$ can be calculated (thus deriving the exact probability that $(i \to j)$ is a forward edge crossing $S$).

How can we calculate $p_{ij}(\alpha,\beta)$ for every $i, j = 0, 1 \dots, n$ and $\alpha, \beta \in \{\pm 1\}$? Fix some $i$, $j$, $\alpha$ and $\beta$. We note that Theorem 2.1 can be used to calculate $p_{ij}(\alpha,\beta)$. The reason is that: (1) Theorem 2.1 provides the value of $p_{ij}(-1,1) + p_{ij}(1,-1)$; (2) $p_{ij}(-1,-1) + p_{ij}(-1,1) + p_{ij}(1,-1) + p_{ij}(1,1) = 1$; and (3) $p_{ij}(-1,-1) = p_{ij}(1,1)$ and $p_{ij}(-1,1) = p_{ij}(1,-1)$ from symmetry. We conclude that using Theorem 2.1 we can exactly calculate the probability that $(i \to j)$ is a forward edge crossing $S$, and obtain that this probability equals $\frac{1}{2}(p_{0j} + p_{ij} - p_{0i})$, where $p_{ij}$ is the probability that $i$ and $j$ are separated as given by Theorem 2.1.

Similarly to MAX-2SAT, not all triplets of angles $\{\theta_{0i}, \theta_{0j}, \theta_{ij}\}$ are possible due to the triangle inequality constraints (here $\theta_{ij}$ indicates the angle between $\mathbf{w}_i$ and $\mathbf{w}_j$). Let us denote by $\mathcal{F}$ the collection of all possible triplet of angles for the MAX-DICUT problem. Then, we can lower bound the approximation guarantee of the Brownian rounding algorithm as follows:

$$\min_{(\theta_{0i},\theta_{0j},\theta_{ij}) \in \mathcal{F}} \left\{ \frac{\frac{1}{2}(p_{0j} + p_{ij} - p_{0i})}{\frac{1}{4}(1 - \cos(\theta_{0j}) + \cos(\theta_{0i}) - \cos(\theta_{ij}))} \right\}.$$

This results in the following theorem.

THEOREM 5.1. *The high dimensional Brownian rounding algorithm achieves an approximation ratio of 0.79 for the* MAX-DICUT *problem.*

We also remark that we can introduce slowdown (as discussed in Section 5.1 to the high dimensional Brownian rounding algorithm. Numerically, we show that this improves the performance to 0.79-approximation.

## 6 Acknowledgements

The authors are grateful to Assaf Naor and Ronen Eldan for sharing their manuscript with us. SA would like to thank Gantumur Tsogtgerel for bringing the Maximum Principle to our attention, and Christina C. Christara for helping us compare various numerical PDE solvers. SA and AN thank Allan Borodin for useful discussions during the initial stages of this research. GG would like to thank Anupam Gupta and Ian Tice for useful discussion, and for also pointing out the Maximum Principle.

SA and AN were supported by an NSERC Discovery Grant (RGPIN-2016-06333). MS was supported by NSF grant CCF-BSF:AF1717947. Some of this work was carried out while AN and NB were visitors at the Simons Institute program on Bridging Discrete and Continuous Optimization, partially supported by NSF grant #CCF-1740425.

## References

[1] Sepehr Abbasi-Zadeh, Nikhil Bansal, Guru Guruganesh, Aleksandar Nikolov, Roy Schwartz, and Mohit Singh. Code for PDE Solvability and Sum of Square Proofs. `https://github.com/sabbasizadeh/brownian-rounding`, 2018. 10

[2] Lars V. Ahlfors. *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable.* McGraw-Hill Book Company, second edition, 1966. 1.2.2, 2.3

[3] Noga Alon and Assaf Naor. Approximating the cut-norm via grothendieck's inequality. *SIAM J. Comput.*, 35(4):787–803, 2006. 1

[4] George E Andrews, Richard Askey, and Ranjan Roy. *Special Functions, volume 71 of Encyclopedia of Mathematics and its Applications.* Cambridge Univ. Press, Cambridge, 1999. 2.3

[5] Takao Asano and David P Williamson. Improved approximation algorithms for MAX SAT. *Journal of Algorithms*, 42 (1):173–202, 2002. 1

[6] Per Austrin. Towards sharp inapproximability for any 2-csp. *SIAM J. Comput.*, 39(6):2430–2463, 2010. 1

[7] Per Austrin and Aleksa Stankovic. Global cardinality constraints make approximating some Max-2-CSPs harder. In *APPROX-RANDOM*, volume 145 of *LIPIcs*, pages 24:1–24:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 1.1

[8] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for Max Bisection. *ACM Trans. Algorithms*, 13(1):2:1–2:27, 2016. 1, 1.1, 1.2.5

[9] Adi Avidor, Ido Berkovitch, and Uri Zwick. Improved approximation algorithms for Max NAE-SAT and Max SAT. In *International Workshop on Approximation and Online Algorithms*, pages 27–40. Springer, 2005. 1

[10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 3–10. IEEE, 2010. 1.1

[11] Nikhil Bansal, Daniel Dadush, and Shashwat Garg. An algorithm for komlós conjecture matching banaszczyk's bound. In *FOCS*, pages 788–799. IEEE Computer Society, 2016. 1.1

[12] Nikhil Bansal, Daniel Dadush, Shashwat Garg, and Shachar Lovett. The Gram-Schmidt walk: a cure for the banaszczyk blues. In *STOC*, pages 587–597. ACM, 2018. 1.1

[13] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *FOCS*, pages 472–481. IEEE Computer Society, 2011. 1.2.5, 4.2

[14] Richard Beals and Roderick Wong. *Special functions: a graduate text*, volume 126. Cambridge Univ. Press, 2010. 2.3

[15] Ilia Binder and Mark Braverman. The rate of convergence of the walk on spheres algorithm. *Geom. Funct. Anal.*, 22 (3):558–587, 2012. ISSN 1016-443X. 2.1

[16] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005. 1

[17] Moses Charikar, Alantha Newman, and Aleksandar Nikolov. Tight hardness results for minimizing discrepancy. In *SODA 2011*, pages 1607–1614. SIAM, 2011. 1.1, 4.1

[18] Ronen Eldan and Assaf Naor. Krivine diffusions attain the goemans-williamson approximation ratio. *CoRR*, abs/1906.10615, 2019. 1.1

[19] Uriel Feige and Michel Goemans. Approximating the value of two power proof systems, with applications to Max 2-SAT and Max DiCut. In *istcs*, page 0182. IEEE, 1995. 1

[20] David Gilbarg and Neil S Trudinger. *Elliptic partial differential equations of second order.* springer, 2015. 1.2.3, 3.3, 3.3

[21] Emmanuel Gobet. Weak approximation of killed diffusion using Euler schemes. *Stochastic Process. Appl.*, 87(2):167–197, 2000. ISSN 0304-4149. 2.1

[22] Emmanuel Gobet. *Monte-Carlo methods and stochastic processes.* CRC Press, Boca Raton, FL, 2016. ISBN 978-1-4987-4622-9. From linear to non-linear. 2.1

[23] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995. (document), 1, 2.1, 2.2

[24] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *FOCS*, pages 482–491. IEEE Computer Society, 2011. 1.2.5, 4.2

[25] Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775. ACM, 2002. 1

[26] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007. 1, 1.1

[27] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the MAX 2-sat and MAX DI-CUT problems. In *IPCO*, volume 2337 of *Lecture Notes in Computer Science*, pages 67–82. Springer, 2002. 1, 1.1

[28] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM Journal on Computing*, 44(5):1573–1582, 2015. 1.1

[29] Peter Mörters and Yuval Peres. *Brownian Motion.* Cambridge Series in Statistical and Probabilistic Mathematics, 2010. 2.2, 2.3, 2.2, 3.2.1, 8

[30] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *FOCS*, pages 21–30. IEEE Computer Society, 2005. 1

[31] Bernt Øksendal. *Stochastic differential equations.* Universitext. Springer-Verlag, Berlin, fourth edition, 1995. An introduction with applications. 1.2.3, 3.2.1, 3.2, 5.1

[32] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *STOC*, pages 245–254. ACM, 2008. 1

[33] Prasad Raghavendra and David Steurer. How to round any CSP. In *FOCS*, pages 586–594. IEEE Computer Society, 2009. 1

[34] Prasad Raghavendra and Ning Tan. Approximating csps with global cardinality constraints using SDP hierarchies. In *SODA*, pages 373–387. SIAM, 2012. 1, 1.1, 1.2.5, 4