

Algorithmic and Optimization Aspects of Brascamp-Lieb Inequalities, via Operator Scaling

Ankit Garg

Microsoft Research New England
Cambridge, Massachusetts, USA
garga@microsoft.com

Rafael Oliveira

Princeton University
Princeton, New Jersey, USA
rmo@cs.princeton.edu

Leonid Gurvits

City College of New York
New York City, New York, USA
l.n.gurvits@gmail.com

Avi Wigderson

Institute for Advanced Studies
Princeton, New Jersey, USA
avi@ias.edu

ABSTRACT

The celebrated Brascamp-Lieb (BL) inequalities and their reverse form of Barthe are an important mathematical tool, unifying and generalizing numerous inequalities in analysis, convex geometry and information theory, with many used in computer science. While their structural theory is very well understood, far less is known about computing their main parameters (which we later define below). Prior to this work, the best known algorithms for any of these optimization tasks required at least exponential time. In this work, we give polynomial time algorithms to compute: (1) Feasibility of BL-datum, (2) Optimal BL-constant, (3) Weak separation oracle for BL-polytopes.

What is particularly exciting about this progress, beyond the better understanding of BL-inequalities, is that the objects above naturally encode rich families of optimization problems which had no prior efficient algorithms. In particular, the BL-constants (which we efficiently compute) are solutions to *non-convex* optimization problems, and the BL-polytopes (for which we provide efficient membership and separation oracles) are linear programs with *exponentially many* facets. Thus we hope that new combinatorial optimization problems can be solved via reductions to the ones above, and make modest initial steps in exploring this possibility.

Our algorithms are obtained by a simple efficient reduction of a given BL-datum to an instance of the *Operator Scaling* problem defined by Gurvits. To obtain the results above, we utilize the two (very recent and different) algorithms for the operator scaling problem (by Garg et al. and Ivanyos et al.). Our reduction implies algorithmic versions of many of the known structural results on BL-inequalities, and in some cases provide proofs that are different or simpler than existing ones. Further, the analytic properties of the algorithm by Garg et al. provide new, effective bounds on the magnitude and continuity of BL-constants, with applications to non-linear versions of BL-inequalities; prior work relied on compactness, and thus provided no bounds.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

STOC'17, Montreal, Canada

© 2017 Copyright held by the owner/author(s). 978-1-4503-4528-6/17/06...\$15.00
DOI: 10.1145/3055399.3055458

On a higher level, our application of operator scaling algorithm to BL-inequalities further connects analysis and optimization with the diverse mathematical areas used so far to motivate and solve the operator scaling problem, which include commutative invariant theory, non-commutative algebra, computational complexity and quantum information theory.

CCS CONCEPTS

• **Theory of computation** → *Mathematical optimization*;

KEYWORDS

Optimization, Brascamp-Lieb, Operator Scaling, Inequalities.

ACM Reference format:

Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. 2017. Algorithmic and Optimization Aspects of Brascamp-Lieb Inequalities, via Operator Scaling. In *Proceedings of 49th Annual ACM SIGACT Symposium on the Theory of Computing, Montreal, Canada, June 2017 (STOC'17)*, 13 pages. DOI: 10.1145/3055399.3055458

1 INTRODUCTION

This long introduction summarizes the main contributions of the paper and the intuitions behind many of the definitions, results and proofs. We start with the main object of this paper: introducing the Brascamp-Lieb inequalities, and surveying known structural results and our new algorithmic results. We then describe how to instantiate the operator scaling algorithm of [29] directly in the context of BL-inequalities, in a way that does not require previous familiarity with it. The properties of this algorithm lead to further structural results, as well as highlight it as a provably efficient instance of the general *alternate minimization*¹ heuristic. Finally, we discuss several known and potential applications of Brascamp-Lieb inequalities in computer science and optimization, which further motivate this and future studies.

¹sometimes also called alternating minimization.

1.1 The Brascamp-Lieb Inequalities: Basic Notions

Many important inequalities, including Hölder’s inequality, Loomis-Whitney inequality, Young’s convolution inequality, hypercontractivity inequalities and many others are all special cases of the extremely general Brascamp-Lieb inequalities, introduced by these authors in [14, 40]. Yet many others, including Prekopa-Lindler inequality, versions of Brunn-Minkowski inequality and others are special cases of Barthe’s reverse form of Brascamp-Lieb [6]. We discuss below only the original form, and only for Euclidean spaces, for simplicity of exposition. The notation we use is taken mostly from [11], an excellent source both for background on this topic, as well as the state-of-art on the basic questions in this field. As is common, we often use BL as abbreviation for Brascamp-Lieb.

A *Brascamp-Lieb datum* is specified by two m -tuples: one of linear transformations $\mathbf{B} = (B_1, B_2, \dots, B_m)$, with $B_j : \mathbb{R}^n \rightarrow \mathbb{R}^{n_j}$ (along with (n, n_1, \dots, n_m) , the vector of *dimensions*), and another of non-negative reals $\mathbf{p} = (p_1, p_2, \dots, p_m)$ (which is a vector of *exponents*). This combined information is denoted by (\mathbf{B}, \mathbf{p}) . A *Brascamp-Lieb inequality*² with the datum above states that for *every* tuple of m non-negative functions, $f = (f_1, f_2, \dots, f_m)$, which are integrable according to the Lebesgue measure, we have the following inequality.

$$\int_{x \in \mathbb{R}^n} \prod_{j=1}^m (f_j(B_j x))^{p_j} dx \leq C \prod_{j=1}^m \left(\int_{x_j \in \mathbb{R}^{n_j}} f_j(x_j) dx_j \right)^{p_j}$$

where $C \in (0, \infty]$ is independent of the functions f . Of course, we really get an interesting inequality if C is finite. In that case, we call the datum (\mathbf{B}, \mathbf{p}) *feasible*, and we denote smallest C for which this inequality holds by $BL(\mathbf{B}, \mathbf{p})$, called the *BL-constant*.

Many familiar special cases of this inequality are listed in [11]; we describe only a very simple one here, which hopefully makes concrete and intuitive the complicated formal expression above and its constituents. A special case of the Loomis-Whitney inequality asserts that *the volume of every measurable set S in the unit cube is at most the square root of the product of the areas of its projections on the three coordinate planes*. In this theorem, the linear transformations $B_j : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ are defined by the simple projections $B_1(x, y, z) = (y, z)$, $B_2(x, y, z) = (x, z)$ and $B_3(x, y, z) = (x, y)$, the functions f_j are the indicator functions of these three projections of the set S , and the exponent vector is $\mathbf{p} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. The corresponding BL-constant in this case happens to be 1 (this case is an important situation which will be important later). The reader may be familiar with an entropy³ version of the inequality above, namely that $H(X, Y, Z) \leq \frac{1}{2}(H(X, Y) + H(Y, Z) + H(X, Z))$ for every random variable (X, Y, Z) . Indeed, BL-inequalities may be viewed as entropy subadditivity inequalities [15, 41].

There has been extensive work on precisely understanding (and computing) when a given datum (\mathbf{B}, \mathbf{p}) is feasible, and if it is, determining the BL-constant $BL(\mathbf{B}, \mathbf{p})$. Clean characterizations exist for both questions, which clarify that they are both decidable (have a finite algorithm). Let us overview these characterizations and then

²While this is the common name of the inequality below found in the literature, we note that only a restricted form appears in the original [14] paper, and this general form appeared only in the later paper of Lieb [40].

³Appropriately defined for these continuous variables.

turn to the computational complexity of the existing algorithms and our new ones. The first, regarding feasibility, is from [11], and the second, on the optimal constant, is from [40]. Before doing so, we make two comments, one about the “reverse BL-inequalities” and the other regarding input size when analyzing algorithms.

Reverse BL inequalities. This informal comment simply clarifies that all results in this paper stated for BL-inequalities hold for their reverse form. In [6], Barthe introduced a reverse form of the Brascamp-Lieb inequalities, sometimes called RBL inequalities, which turn out to generalize some known inequalities not captured by the the original BL inequalities. These RBL inequalities take the *same* data as standard BL inequalities, and have an optimal RBL constant associated with each. All questions raised above for the BL inequalities, like feasibility, computation of the constant and its continuity properties are relevant in this reverse setting. However, they are actually equivalent to the original ones for a simple reason: Barthe [6] proved that for any feasible datum the optimal BL constant and the optimal RBL constant multiply to 1, and when the BL datum is infeasible the RBL constant is 0. In short, these two optimal constants determine each other. Thus all structural results above translate to the reverse setting, and so do all our algorithmic results.

Input size conventions. Before we start, let us formalize the input conventions to all algorithms, and the parameters we use to measure their complexity. The input to all algorithms will be a BL datum (\mathbf{B}, \mathbf{p}) , and the “size” of each part will be measured differently. The entries of the matrices B_j will be rational numbers, given in binary. We will let $b = b(\mathbf{B})$ denote their total binary length (note that in particular $b \geq nm$, and so upper bounds the “combinatorial size” of the input). The vector \mathbf{p} will be given as a sequence of rationals with a common denominator, namely $p_j = c_j/d$ with c_j, d integers. We use this convention for two reasons. First, our algorithms will use this integer representation, and their complexity will depend on $d = d(\mathbf{p})$ (while in many cases d is only polynomial in the dimensions n, m of the problem, it can definitely be as large as exponential in other cases). Second, in the context of *quiver representations*, which is very relevant to this study, it is natural to use these integer “weights” c_j and d .

Summarizing, the two size parameters for a BL datum (\mathbf{B}, \mathbf{p}) will be b and d as above.

1.2 The Brascamp-Lieb Inequalities: Known and New Results

Testing feasibility (and more). The following theorem of Bennet et al [10, 11] precisely characterizes when a given BL datum is feasible. This work will provide a different proof of this important theorem (see Corollary 3.5).

THEOREM 1.1 ([11]). *The datum (\mathbf{B}, \mathbf{p}) is feasible if and only if the following conditions hold:*

- (1) $n = \sum_j p_j n_j$.
- (2) $\dim(V) \leq \sum_j p_j \dim(B_j V)$ for all subspaces V of \mathbb{R}^n .

Note that these are simply linear conditions on the exponent vector \mathbf{p} , albeit defined by infinitely many subspaces V . However,

as the coefficients are integers in $[n]$, there are at most n^m different inequalities, and so the linear maps \mathbf{B} define a polytope $P_{\mathbf{B}}$ (sometimes called the *BL polytope*) in \mathbb{R}^n , such that (\mathbf{B}, \mathbf{p}) is feasible iff $\mathbf{p} \in P_{\mathbf{B}}$. The BL polytopes $P_{\mathbf{B}}$ have received a lot of attention. For example, their vertices were characterized for the “rank-1 case” (namely, when all dimensions $n_j = 1$) by Barthe [6], and this was extended to other cases by Valdimarsson [48]. In the same paper, Valdimarsson considers the question of generating the inequalities defining $P_{\mathbf{B}}$, and gives a finite algorithm to do so (needless to say, after they are given, feasibility testing becomes a linear programming problem). No explicit upper bound on the complexity of this algorithm is given, but it is at least exponential in m . The same holds for the algorithm in [17] for generating the inequalities defining $P_{\mathbf{B}}$.

We give a polynomial time algorithm for the feasibility problem, and much more. Our algorithm also gives a “separation oracle” (namely finds a violated inequality when infeasible). Recall again that the exponents \mathbf{p} in the BL datum (\mathbf{B}, \mathbf{p}) are given by $p_j = c_j/d$ where c_j, d are integers.

THEOREM 1.2 (COROLLARY 3.6 REPHRASED). *There is an algorithm that on input (\mathbf{B}, \mathbf{p}) of binary length b and common denominator d runs in time $\text{poly}(b, d)$ and provides the following:*

Membership oracle: Tests if $\mathbf{p} \in P_{\mathbf{B}}$.

Separation oracle: When $\mathbf{p} \notin P_{\mathbf{B}}$, provides a violated inequality, namely a basis for a vector space V in \mathbb{R}^n such that $\dim(V) > \sum_j p_j \dim(B_j V)$.

We believe that the ability to efficiently optimize over such a wide family of polytopes with exponentially many facets should be useful for different optimization problems via appropriate reductions to this setting. We will give some simple examples of such reductions in Section 7. One concrete challenge is e.g. trying to find such a reduction which embeds the Edmonds polytope [26] of perfect matchings in a general (non-bipartite) graph into some BL polytope.

It is an interesting open problem if one can improve the algorithm to depend polynomially on $\log d$ instead of d . This would allow for optimization over the BL polytopes in polynomial time via the ellipsoid algorithm. We believe even the current separation oracle for BL polytopes with polynomial dependence on d should allow for approximate optimization over BL polytopes but we haven’t been able to prove this yet.

Characterizing the BL constant. The following theorem of Lieb [40] characterizes the BL constant $\text{BL}(\mathbf{B}, \mathbf{p})$ in all cases where the BL datum is feasible. The heart of the proof establishes that the optimal constant in any BL inequality is attained (or approached) by plugging in density functions of appropriate centered Gaussians. For such densities, the BL constant has a nice expression via the following identity:

$$\int_{\mathbb{R}^n} \exp(-\pi x^T A x) dx = \text{Det}(A)^{-1/2}$$

THEOREM 1.3 ([40]). *Assume (\mathbf{B}, \mathbf{p}) is feasible. Then*

$$\text{BL}(\mathbf{B}, \mathbf{p}) = \left[\sup \frac{\prod_j (\det X_j)^{p_j}}{\det \left(\sum_j p_j B_j^\dagger X_j B_j \right)} \right]^{1/2}$$

where the supremum is taken over all choices of psd matrices X_j in dimension n_j , and B_j^\dagger denotes the adjoint map to B_j .

Thus the BL constant for given BL datum is a solution to an optimization problem. However, as defined, it is not even convex. We are not aware of any algorithms to compute the BL constant in general. Moreover, to the best of our knowledge no explicit bounds on the BL constant (when finite) in terms of the BL datum were known. We resolve both issues: provide an explicit bound and give a polynomial time algorithm for computing the BL constant to any accuracy.

THEOREM 1.4 (COROLLARY 4.9 REPHRASED). *For any feasible BL datum (\mathbf{B}, \mathbf{p}) that has binary length b and common denominator d , it holds that $\text{BL}(\mathbf{B}, \mathbf{p}) \leq \exp(O(b \log(bd)))$.*

THEOREM 1.5 (THEOREM 5.2 REPHRASED). *There is an algorithm that on input (\mathbf{B}, \mathbf{p}) of binary length b and common denominator d , and an accuracy parameter $\epsilon > 0$, runs in time $\text{poly}(b, d, 1/\epsilon)$ and computes a factor $(1 + \epsilon)$ -approximation of $\text{BL}(\mathbf{B}, \mathbf{p})$. Furthermore, the algorithm outputs a scaling \mathbf{B}' which is almost geometric i.e. $\text{BL}(\mathbf{B}', \mathbf{p}) \leq 1 + \epsilon$.*

1.3 Efficient Computation of the BL-constant via Scaling

The algorithm underlying the proof of this theorem will be shown (via a reduction) to be a special case of the operator scaling algorithm of Gurvits [30], which the current authors analyzed and proved to be polynomial time in [29].

In this subsection, we explicitly describe the algorithm in this special case of BL-inequalities only, without referring to [29]. It will be instructive to see how notion of scaling, which naturally exists in the theory of BL inequalities, can be used algorithmically. Moreover, it will become clear how the algorithm implies (known and new) structural consequences to the BL theory. This description will also help to motivate our reduction in the technical chapters which follow (which will provide the proof for its run-time).

Let us return to the BL constant, and to an important family of BL data called *geometric*⁴ defined below. It was introduced by Ball [2] and extended by Barthe [6].

Definition 1.6. A BL datum (\mathbf{B}, \mathbf{p}) is called *geometric* if it satisfies the following conditions (with I_k denoting the $k \times k$ identity matrix).

Projection: $\forall j \in [m], B_j$ is a projection, i.e. $B_j B_j^\dagger = I_{n_j}$.

Isotropy: $\sum_j p_j B_j^\dagger B_j = I_n$

Ball proved that for geometric BL datum the constant is always one (the general version is due to Barthe)⁵.

THEOREM 1.7 ([2, 6]). *For every geometric BL datum (\mathbf{B}, \mathbf{p}) have $\text{BL}(\mathbf{B}, \mathbf{p}) = 1$.*

A simple and natural action of the general linear groups on the Euclidean spaces involved (which simply performs a basis change

⁴The analogous term in the operator scaling setting will be called *doubly stochastic*.

⁵The analogous theorem for operator scaling is that the capacity of doubly stochastic operators is always 1. Another family of inequalities where the constant is always 1 are the “discrete” BL inequalities of [18].

in each space) yields an equivalence relation on BL data⁶. Specifically, we say that (\mathbf{B}, \mathbf{p}) and $(\mathbf{B}', \mathbf{p}')$ are *equivalent* if there exist matrices $C \in GL_n(\mathbb{R})$, $C_j \in GL_{n_j}(\mathbb{R})$ (which are called *intertwining transformations*⁷ in [11]) such that $B'_j = C_j^{-1} B_j C$ for all j . It is easy to compute the effect of such action on the BL constant.

THEOREM 1.8 ([11]). *Assume (\mathbf{B}, \mathbf{p}) and $(\mathbf{B}', \mathbf{p}')$ are equivalent via C, C_j as above. Then*

$$\text{BL}(\mathbf{B}', \mathbf{p}') = \frac{\prod_j (\det(C_j))^{p_j}}{\det(C)} \text{BL}(\mathbf{B}, \mathbf{p}).$$

where \det is the determinant polynomial on matrices of the appropriate dimension.

Equivalence and this simple, efficiently computable formula suggests a natural path to computing BL constants. Given BL datum (\mathbf{B}, \mathbf{p}) , compute an equivalent *geometric* datum $(\mathbf{B}', \mathbf{p}')$ (if one exists), and the intertwining transformations relating the two. The main questions are, is there such a geometric equivalent datum, and how to compute these transformations. The theory of (quantum) operator scaling [29, 30] suggests that a simple, greedy procedure will work. This algorithm will underly (most of) the statements in Theorem 1.2 and Theorem 1.5. As mentioned, we will explain the connection and reduction to operator scaling in the next sections, and here describe informally how it is applied to the BL setting.

The scaling algorithm. The key idea of *greedy, iterative* scaling, which goes back to Sinkhorn [47] in the (classical) matrix scaling, is that when trying to achieve a pair of conditions as in the definition of geometric BL datum, we should try to satisfy them one at a time! Say we are given (\mathbf{B}, \mathbf{p}) and assume for example that the isotropy condition is not met. A non-triviality condition (otherwise (\mathbf{B}, \mathbf{p}) is not feasible) implies that $C \triangleq \sum_j p_j B_j^\dagger B_j$ is invertible, and so we can set $B_j \leftarrow B_j C^{-1/2}$ to get a new datum which satisfies isotropy. So, we “only” have to fix the projection property; lets do it. Again, non-triviality implies that each of $C_j \triangleq B_j B_j^\dagger$ is invertible, and we can now set $B_j \leftarrow C_j^{-1/2} B_j$, satisfying the projection property. Of course, we may now have ruined isotropy. No problem: lets fix it again in the same way, and repeat alternately fixing the unsatisfied property. The magic is that this sequence of *normalization* steps converges, and moreover, converges in polynomial time, to a geometric datum, whenever the original datum (\mathbf{B}, \mathbf{p}) is feasible! Let us describe the BL algorithm more precisely and then state its properties formally.

Input: A BL datum (\mathbf{B}, \mathbf{p})
Set $\mathbf{B}^0 = \mathbf{B}$.

Repeat for $i = 1$ to t normalization steps.

- **Isotropy:** if i is odd, normalize \mathbf{B}^i from \mathbf{B}^{i-1} .
- **Projection:** if i is even, normalize \mathbf{B}^i from \mathbf{B}^{i-1} .

Output: \mathbf{B}_t .

Algorithm 1: The BL scaling algorithm

⁶This group action naturally calls for a study of BL from an invariant theory viewpoint, which indeed exists in a much more general context that will be relevant to us in several ways, namely that of *quiver representations*. An extensive survey is [22].

⁷They will be called *scaling matrices* in the operator scaling setting.

Algorithm 1 defines a dynamics in the space of BL data which are restricted to stay in one equivalence class. If (\mathbf{B}, \mathbf{p}) is the original datum which is not already geometric, it defines a sequence $\mathbf{B}^0 = \mathbf{B}, \mathbf{B}^1, \mathbf{B}^2, \dots$ which alternately satisfies the isotropy or projection properties (note that the exponent vector \mathbf{p} is used, but not changed in this process). We now state the main three properties which underlie the analysis.

THEOREM 1.9. *For every (\mathbf{B}, \mathbf{p}) the following holds:*

Upper bound: *If (\mathbf{B}, \mathbf{p}) is feasible, then*

$$\text{BL}(\mathbf{B}, \mathbf{p}) \leq \exp(O(b \log(bd))).$$

Lower bound: *If (\mathbf{B}, \mathbf{p}) is either isotropy-normalized or projection-normalized, then $\text{BL}(\mathbf{B}, \mathbf{p}) \geq 1$.*

Progress per step: *Let $(\mathbf{B}', \mathbf{p})$ denote the result of applying either isotropy-normalization or projection-normalization to (\mathbf{B}, \mathbf{p}) . Then (as long as $\text{BL}(\mathbf{B}, \mathbf{p}) \geq 1 + \epsilon$), we have*

$$\text{BL}(\mathbf{B}', \mathbf{p}) \leq (1 - \text{poly}(\epsilon/nd)) \text{BL}(\mathbf{B}, \mathbf{p}).$$

We will not need to prove this theorem, as it will follow via our reduction from the analogous theorem in the more general operator scaling setting⁸. However, let us say a few words here about what goes into each item above using BL language. Property (3) on progress per step is the simplest: it follows from a robust version of the AM-GM inequality. Property (2) follows from plugging in densities of spherical Gaussians. Property (1) is the hardest, in that it requires degree bounds on the generating polynomial invariants in the null-cone of the quiver associated to the group action above. We will briefly explain this mouthful in section 4, and refer the reader to [29] for details. In section 4, we will also given an improved and simplified analysis of the main theorem from [29] and derive the desired upper bounds on the BL constant via our main reduction. All in all, it is worth stressing again that beyond this non-trivial use of algebra, on the analytic side the most general BL inequalities follow from the most basic one, AM-GM.

This analysis above implies that we can choose the number t of steps in the BL scaling algorithm to be $\text{poly}(b, d, 1/\epsilon)$ so that $\text{BL}(\mathbf{B}^i, \mathbf{p}) \leq 1 + \epsilon$ for some $1 \leq i \leq t$ (whenever the initial datum $(\mathbf{B}^0, \mathbf{p})$ is feasible). Testing if the lower bound is met ($\text{BL}(\mathbf{B}^i, \mathbf{p}) \leq 1 + \epsilon$) decides feasibility⁹, thus proving the first item in Theorem 1.2. For proving the second item we will employ a very different, beautiful combinatorial algorithm for operator scaling (only the decision version of it), that was developed after ours by Ivanyos, Qiao and Subrahmanyam [35]. Their algorithm relies on even tighter degree bounds from invariant theory which were proved very recently [21], and has several advantages over the algorithm in [29]; in particular it can find violated dimension inequalities. Going back to Theorem 1.5 on approximating the BL constant, note that it suffices to multiply together the (respective) normalizing matrices used in all steps of the BL scaling algorithm, using the formula in Theorem 1.8.

We now return to the scaling algorithm above, and discuss connections and applications of its structure and properties.

⁸Where statements made here regarding the BL constants will be replaced by statements about a related notion called the *capacity* of an operator.

⁹It is not immediately clear how to test $\text{BL}(\mathbf{B}^i, \mathbf{p}) \leq 1 + \epsilon$. However this can be done by checking closeness to the geometric position in an appropriate metric, similar to the distance to doubly stochasticity for operators that will be defined later.

The BL constant, scaling and alternate minimization. The problem of computing the BL constant, as it is formulated here, is a non-convex problem. We presented an efficient algorithm for this optimization problem which may be viewed as an instance of a general heuristic called *alternate minimization*. Let us describe and discuss this method.

In general, alternate minimization is a very general technique introduced by [19], which was devised to solve problems of the following kind. Given some universe U , a (distance) function $d : U \times U \rightarrow \mathbb{R}_+$ and two subsets $X, Y \subseteq U$, find $(\mathbf{x}^*, \mathbf{y}^*) = \arg \min_{\mathbf{x} \in X, \mathbf{y} \in Y} d(\mathbf{x}, \mathbf{y})$. We note that such problems are in general NP-hard, even over convex sets X, Y and natural distances d . Nevertheless, alternate minimization is widely used to solve special cases of such problems in practice.¹⁰

An alternate minimization algorithm for the problem above starts with an arbitrary point $(x_0, y_0) \in X \times Y$ and generates a sequence of points (x_i, y_i) as follows. In even steps i , $x_{i+1} = x_i$, and y_{i+1} is chosen to be the value of y which minimizes $d(x_i, y)$ over the second coordinate alone (an optimization problem that is often far simpler). In odd steps the roles of x and y are reversed. It is easy to cast the scaling algorithm above as a process of this nature. It is also easy to see that it enjoys some nice properties in general situations. For example, if both X and Y are strictly convex, and d is a metric, then this process converges to the optimum. Of course, the speed of convergence is the main question. We list below some other, possibly familiar instances of alternate minimization for which such convergence is not known.

The famous Lemke-Howson algorithm for finding a Nash equilibrium for 2-player games [39] is of that nature. In that algorithm one starts with an arbitrary strategy for both players, and then one alternates fixing the strategy of one, and finding the best response for the other via linear programming. Another such example is the work of Zafeiriou and Petrou [49] on computing non-negative tensor factorization. As above, the analysis of such alternating minimization algorithms just provide convergence of the procedure, without proving *how fast* it converges to the minimum, with the hope that this heuristic will lead to quick convergence on instances arising in practice. Of course, there are examples, e.g. the alternate minimization algorithms for matrix completion in [32, 37] which prove rapid convergence under certain conditions on the inputs.

In this work, as well as in [29], we prove that our alternate minimization algorithm not only converges, but also converges to the infimum in polynomial time. Our proof is based on the introduction of a potential function which measures how much progress we make in every step of the alternating minimization algorithm. In our case the potential function is the very BL constant we are trying to optimize (which is a special case of the so-called *capacity* we use as potential in our operator scaling algorithm [29]). As it happens, the analysis of convergence relies on a combination of algebraic tools, in particular from (the commutative) invariant theory and (the non-commutative) theory of skew fields, as explained in that paper. We hope that such potential functions as well as the methods to analyze their progress, which we call *capacity methods*, will

¹⁰Note that such problems can be defined with more than two components, and the alternate minimization approach below can be extended to such problems, but we stick here with two.

be of use in the analyses of other problems which use alternate minimization, especially when the result of optimizing each of the solution component in individual optimization steps may be viewed as the action of a group. We are currently exploring this direction for alternate minimization with more than two components, and these lead to interesting algebraic questions related to questions that arise in invariant theory as well as in Geometric Complexity Theory of Mulmuley and Sohoni [43, 44].

Bounds and continuity of the BL constant, and non-linear BL inequalities. Another important property of the BL scaling algorithm above, evident from its form, is that it is smooth! Namely, the orbits under the algorithm of two sufficiently close BL data will remain close to each other for the duration of the algorithm. This implies that the BL constants of both will be close as well, and moreover this can be quantified! This simple observation gives a much stronger result than existing qualitative results on the continuity of BL constants which has recently received significant attention. This is yet another example highlighting the usefulness of algorithmic results in mathematics. We briefly overview the motivation and known results before stating ours.

How smooth (or regular) the BL constant is (in terms of the BL data) is of course a natural question, especially as the expression in Lieb's theorem 1.3 is not even convex. Further motivation to study these question arises from a variety of non-linear variants of the BL inequalities, in which the maps B_j are non-linear, but at least smooth enough so as to be approximable by linear ones in a small ball. Some such variants, needed in specific applications, were proved directly. The papers [7, 9] ask this question in full generality, and demonstrate the importance of such smoothness conditions of the BL constant for general non-linear extensions of BL inequalities. These papers also exposit the many diverse applications of such non-linear versions in analysis, number theory and other areas (possibly the most impressive recent ones are the applications of "Kakeya-type" BL inequalities to number theory in [12, 13], resolving long standing open problems including the "Vinogradov mean-value conjecture").

Local boundedness, a condition weaker than continuity, is established in [9].

THEOREM 1.10 ([9]). *If (\mathbf{B}, \mathbf{p}) is feasible, then there are $\delta > 0$ and $C \leq \infty$ such that for all $(\mathbf{B}', \mathbf{p})$ such that $\|\mathbf{B}' - \mathbf{B}\|_2 \leq \delta$ we have $\text{BL}(\mathbf{B}', \mathbf{p}) \leq C$.*

This was followed recently by the paper [8], which proves continuity of the BL constant.

THEOREM 1.11 ([8]). *The BL constant $\text{BL}(\mathbf{B}, \mathbf{p})$ is continuous in \mathbf{B} . Namely, for every feasible (\mathbf{B}, \mathbf{p}) and $\epsilon > 0$, there exists a $\delta > 0$ such that if $\|\mathbf{B}' - \mathbf{B}\|_2 \leq \delta$, then we have that $|\text{BL}(\mathbf{B}', \mathbf{p}) - \text{BL}(\mathbf{B}, \mathbf{p})| \leq \epsilon$.*

In both theorems above, no quantitative bounds are given. This is not surprising, as the proofs of both use compactness. On the other hand, the availability of a smooth algorithm, together with the quantitative analysis of its continuity provided in [29], imply via our reduction the following quantitative bound on the continuity¹¹ of the BL constant (giving a multiplicative approximation).

¹¹While Theorem 1.12 only implies continuity of $\text{BL}(\mathbf{B}, \mathbf{p})$ for rational \mathbf{B} and \mathbf{p} , we describe in Section 6 how our proof can be extended to the case of irrational \mathbf{B} and speculate how this might also extend to irrational \mathbf{p} .

THEOREM 1.12 (THEOREM 6.4 REPHRASED). *For every parameters $b, d, \epsilon > 0$ there exists a*

$$\delta = \delta(b, d, \epsilon) \leq \exp\left(-\frac{\text{poly}(b, d)}{\epsilon^3}\right)$$

such that for every BL data (\mathbf{B}, \mathbf{p}) of size b, d and every BL data $(\mathbf{B}', \mathbf{p})$ such that $\|\mathbf{B}' - \mathbf{B}\|_2 \leq \delta$, we have that if $\text{BL}(\mathbf{B}, \mathbf{p})$ is finite, then $\text{BL}(\mathbf{B}', \mathbf{p})$ is finite. Furthermore,

$$(1 - \epsilon)\text{BL}(\mathbf{B}, \mathbf{p}) \leq \text{BL}(\mathbf{B}', \mathbf{p}) \leq (1 + \epsilon)\text{BL}(\mathbf{B}, \mathbf{p})$$

We hope this explicit bound will be useful to make non-linear BL inequalities quantitative, and to their applications.

1.4 Applications of Brascamp-Lieb in Computer Science, Optimization and Beyond

Besides applications to other areas of mathematics as we mentioned above, the structural theory of BL inequalities (sometimes with the connection to operator scaling) have encountered several applications in computer science and optimization. Below is a (partial) survey of these applications.

Many diverse applications exist even for the special case of rank-1 BL setting (when all the n_i 's are 1), which has been studied extensively. In this case, the BL polytopes are exactly the *basis polytopes* [6]. More specifically, if the maps B_i are given by $B_i(x) = \langle v_i, x \rangle$, then the BL polytope is the convex hull of the following set:

$$\{I_I : I \subset [m], |I| = n, (v_i)_{i \in I} \text{ forms a basis for } \mathbb{R}^n\}$$

Hence Theorem 1.2 for this special case is well known. Furthermore, computing the BL constant in this special case can be formulated as a convex problem and hence Theorem 1.5 in this special case can be obtained by the ellipsoid algorithm [31, 33] ([31] in fact solves a more general problem). We list some of the applications of this setting.

- In [33], Hardt and Moitra provide an application of scaling the rank-1 BL datum to a geometric position (which they call isotropic) to the problem of *robust subspace recovery*.
- Forster [27] used the existence of a scaling to geometric position for rank-1 BL datum (sometimes called Barthe's theorem) to give the first nontrivial lower bounds for the *sign rank* of an explicit matrix (in particular the Hadamard matrix), and thereby prove the first nontrivial *unbounded error communication complexity* lower bounds for the inner product function.
- In [24], Dvir et. al. use Barthe's theorem as part of the first super-quadratic lower bounds on the size of 3-query *LCCs* (*Locally Correctable Codes*) over the Real numbers.
- Recently, Nikolov and Singh [45] use a optimization problem similar to the rank-1 BL constant to give a polynomial time algorithm for approximating the maximum value of the determinant of a submatrix of a psd matrix under partition constraints.

The class of BL polytopes is a rich class of polytopes. As mentioned above, the rank-1 case gives rise to basis polytopes. The rank-2 case (when all n_i 's are 1 or 2) has been completely characterized by Valdimarsson [48]. An important special case is given by the *linear matroid intersection* polytopes which we describe next. If

(v_1, \dots, v_m) and (w_1, \dots, w_m) are two sets of vectors in \mathbb{R}^n , then the matroid intersection polytope given by these collections is the convex hull of the set:

$$\{I_I : I \subset [m], |I| = n, (v_i)_{i \in I}, (w_i)_{i \in I} \text{ both are basis for } \mathbb{R}^n\}$$

Let us define the maps $B_i : \mathbb{R}^{2n} \rightarrow \mathbb{R}^2$ in the following way:

$$B_i(x, y) = (\langle v_i, x \rangle, \langle w_i, y \rangle)$$

Then the BL polytope $P_{\mathbf{B}}$ is exactly the corresponding matroid intersection polytope. In particular, the bipartite matching polytopes are a special case of rank-2 BL polytopes.

We will give proofs of these statements in Section 7, with the hope of demonstrating the simplicity by which such exponential size linear programs of this form capture (simple) combinatorial optimization problems. As mentioned before, it remains a very interesting challenge to see if the general matching polytopes (or in general the linear matroid matching polytopes) are a special case of BL polytopes, for which it may still suffice to stay in the rank-2 case. We note a peculiar discrepancy between operator scaling and BL-inequalities when encoding the matroid intersection problem. In the case of operator scaling, this problem was one of the first special cases considered in Gurvits' paper [30], and he showed that one could encode it by an operator scaling problem with matrices of rank 1. Here, when encoded as a BL polytope, we use rank-2 projectors, and believe it is impossible to do in rank 1. Even though BL is a special case of Operator scaling (as our main reduction shows) the relative power of these two formulations with respect to rank is not clear.

Characterizing vertices of BL-polytopes for projectors of rank 3 and higher is a very interesting question, as is the case of rank-2 matrices in the formulation of operator scaling.

BL-inequalities (and operator scaling) have recently found applications in computational and combinatorial geometry in the paper of Dvir et al. [23]. They generalize the rank lower bounds for design matrices given in [3, 25] to the setting of block matrices. This result allows them to obtain sharp bounds on Sylvester-Gallai type theorems for arrangements of subspaces, to obtain new incidence bounds for high-dimensional line/curve arrangements, as well as to prove structural rigidity results in the projective setting.

Beyond *applying* BL-inequalities, we note that in recent years computer scientists started getting interested in *proving* them efficiently. Efficiency here should be taken to mean in the Sum-of-Squares (SoS) framework, a hierarchy of semi-definite programs that is perhaps the most powerful general algorithmic technique for a variety of optimization problems. The paper [4], generalizing from many particular examples on the power of SoS algorithms, suggests a general framework for rounding solutions in this system, in which a crucial element is efficient SoS proofs of inequalities (which in all known applications are special cases of BL-inequalities). This viewpoint has led to many new recent SoS algorithms [5, 34, 42], and naturally also lead to the question of efficiently proving (in this sense) all BL-inequalities. This was taken up by Lei and Sheng [38], who have (in a very general setting) SoS proofs of degree $\text{poly}(d)$ for BL-inequalities that have denominator d in its exponent \mathbf{p} -vector. This yields an algorithm for testing feasibility that is *exponential* in d . Our algorithms depend polynomially in d , and raise the question of relating operator scaling and SoS algorithms, and whether such

a relation can help improve the bounds above. Needless to say, relating the power of these two optimization methods is motivated for many other reasons!

Dictionary. We give here a basic dictionary describing translation between central notions in the theory of Brascamp-Lieb inequalities and completely positive operators, denoted here by CPO. This will be useful in reading the technical sections below. The basic notions concerning completely positive operators will be defined in Section 2.

Table 1: Dictionary

Brascamp-Lieb inequalities	CPO
BL datum	Kraus operators
Intertwining transformations	Scaling operations
Geometric BL datum	Doubly stochastic
BL constant	Capacity

Organization of the paper. All but the last two sections are devoted to providing background and then proving the main technical results of this paper. In Section 2, we will extend these results to *rectangular* completely positive operators via a simple reduction. In Section 3, we describe our main reduction, from BL data to (rectangular) completely positive operators, and prove Theorem 1.2. In Section 4, we describe an upper bound on the Brascamp-Lieb constants and prove Theorem 1.4. In Section 5, we show how to approximate Brascamp-Lieb constants by using the main reduction above and our operator scaling algorithm (proving Theorem 1.5). In Section 6, we use the smoothness of the scaling algorithm to give explicit bounds for the continuity of Brascamp-Lieb constants and prove Theorem 1.12. In Section 7 (which is independent from all others) we exemplify the power of exponentially large linear programs arising as BL-polytopes to capture certain simple combinatorial optimization problems. Finally, we conclude with some open problems in Section 8. Due to space constraints, we will omit the proofs of some intermediate lemmas. The full version of this paper can be found at [28].

2 RECTANGULAR OPERATORS

In this section we extend the definitions and main results of the theory of square operators, presented in [29, 30], to rectangular operators, which we will define below.

We define a rectangular operator as any linear map $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$. Choi's characterization [16] extends to rectangular operators as well. And this is how we define completely positive rectangular operators.

Definition 2.1 (Rectangular Completely Positive Operators). An operator (or map) $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ is called completely positive if there are $n_2 \times n_1$ complex matrices A_1, \dots, A_m s.t. $T(X) = \sum_{i=1}^m A_i X A_i^\dagger$. The matrices A_1, \dots, A_m are called Kraus operators of T (and they are not unique). T is called completely positive trace preserving (cptp) if in addition, $\text{tr}(T(X)) = \text{tr}(X)$ for all X . This is equivalent to the condition $T^*(I) = \sum_{i=1}^m A_i^\dagger A_i = I$.

The following is a natural extension of the definition of doubly stochastic operators to the rectangular case.

Definition 2.2. If $T(X) = \sum_{i=1}^m A_i X A_i^\dagger$ is a completely positive operator, we define its dual T^* by $T^*(X) = \sum_{i=1}^m A_i^\dagger X A_i$. We say T is doubly stochastic if $T\left(\frac{n_2}{n_1} I_{n_1}\right) = I_{n_2}$ and T is trace preserving. This is the same as

$$T\left(\frac{n_2}{n_1} I_{n_1}\right) = I_{n_2} \quad \text{and} \quad T^*(I_{n_2}) = I_{n_1}.$$

The following is a natural extension of the rank decreasing property to the rectangular case. For a matrix X of dimension n we defined its *fractional-rank* as $\text{Rank}(X)/n$.

Definition 2.3 (Fractional-Rank Decreasing Operators). A completely positive operator T is fractional-rank decreasing if there exists an $X \geq 0$ s.t. $\frac{\text{rank}(T(X))}{n_2} < \frac{\text{rank}(X)}{n_1}$.

The following is a natural extension of capacity to the rectangular case.

Definition 2.4 (Capacity). The capacity of a completely positive operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$, denoted by $\text{cap}(T)$, is defined as

$$\begin{aligned} \text{cap}(T) &= \inf \left\{ \frac{\text{Det}\left(\frac{n_2}{n_1} T(X)\right)}{\text{Det}(X)^{\frac{n_2}{n_1}}} : X > 0 \right\} \\ &= \inf \left\{ \text{Det}\left(\frac{n_2}{n_1} T(X)\right) : X > 0, \text{Det}(X) = 1 \right\} \end{aligned}$$

Definition 2.5 (Operator Scaling). An operator T' is called an operator scaling of T if there exist invertible matrices B, C (which are $n_2 \times n_2$ and $n_1 \times n_1$ respectively) s.t.

$$T'(X) = B \cdot T\left(C \cdot X \cdot C^\dagger\right) \cdot B^\dagger$$

Alternatively, if A_1, \dots, A_m are Kraus operators for T , then the matrices BA_1C, \dots, BA_mC are Kraus operators for T' .

Definition 2.6 (Distance to doubly stochasticity).

$$\text{ds}(T) = \text{tr} \left[\left(T\left(\frac{n_2}{n_1} I_{n_1}\right) - I_{n_2} \right)^2 \right] + \text{tr} \left[\left(T^*(I_{n_2}) - I_{n_1} \right)^2 \right]$$

The next two operations are natural extensions of the right and left normalizations to the case of rectangular operators.

Definition 2.7 (Right Normalization). Given a completely positive operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$, define its *right normalization* T_R as follows:

$$T_R(X) = T\left(T^*(I_{n_2})^{-1/2} \cdot X \cdot T^*(I_{n_2})^{-1/2}\right)$$

Note that $T_R^*(I_{n_2}) = I_{n_1}$.

Definition 2.8 (Left Normalization). Given a completely positive operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$, define its *left normalization* T_L as follows:

$$T_L(X) = \frac{n_1}{n_2} \cdot T\left(I_{n_1}\right)^{-1/2} \cdot T(X) \cdot T\left(I_{n_1}\right)^{-1/2}$$

Note that $T_L\left(\frac{n_2}{n_1} I_{n_1}\right) = I_{n_2}$.

Algorithm 2 is a natural extension of Algorithm G [29, 30] to the case of rectangular operators.

Input: Completely positive operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$. Perform right and left normalizations starting from $T_0 = T$ alternately for t steps. Let T_j be the operator after j steps.
Output: T_t

Algorithm 2: Algorithm G for rectangular operators

We will now see how from a rectangular operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$, we can define a square operator $\tilde{T} : M_{n_1 n_2}(\mathbb{C}) \rightarrow M_{n_1 n_2}(\mathbb{C})$ that captures most of the properties of T (we can also embed in dimension $\text{lcm}(n_1, n_2)$ but since it only saves a quadratic factor, we ignore this to simplify notation). \tilde{T} is intended to act on block diagonal matrices (the off-diagonal blocks are ignored).

CONSTRUCTION 2.9. Let $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ be an operator. Given an $n_1 n_2 \times n_1 n_2$ psd matrix X , view it as a $n_2 \times n_2$ block matrix, where each block $X_{i,j}$ is $n_1 \times n_1$. Define $\tilde{T}(X)$ as $n_1 \times n_1$ block diagonal matrix where each diagonal block is the same $n_2 \times n_2$ matrix: $\frac{1}{n_1} \sum_{i=1}^{n_2} T(X_{i,i})$. In other words, we have that

$$\tilde{T}(X) = I_{n_1} \otimes \left(\frac{1}{n_1} \sum_{i=1}^{n_2} T(X_{i,i}) \right)$$

In terms of Kraus operators, if A_1, \dots, A_m are Kraus operators for T , then a set of Kraus operators for \tilde{T} are $\left\{ \frac{1}{n_1} E_{i,j} \otimes A_k \right\}_{i=1, j=1, k=1}^{n_1, n_2, m}$. Here $E_{i,j}$ is the elementary matrix with a 1 in position (i, j) and 0's everywhere else.

The next lemma shows some useful relations between the operators T and \tilde{T} . A proof of the lemma can be found in the full version of the paper [28].

LEMMA 2.10. If $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ is an operator and \tilde{T} is the operator obtained from Construction 2.9, the following hold:

- (1) $\text{cap}(\tilde{T}) = \text{cap}(T)^{n_1}$.
- (2) T is fractional-rank non-decreasing iff \tilde{T} is rank non-decreasing.

With Construction 2.9, most of the results about square operators carry over to the rectangular case.

THEOREM 2.11. A rectangular completely positive operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ is fractional-rank non-decreasing iff $\text{cap}(T) > 0$.

For any completely positive operator, $\text{cap}(T) > 0$ if, and only if, $\text{cap}(T^*) > 0$. A proof of the proposition can be found in the full version of the paper [28].

PROPOSITION 2.12. Let $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ be a completely positive operator. Then $\text{cap}(T)^{1/n_2} = \frac{n_2}{n_1} \text{cap}(T^*)^{1/n_1}$.

Thus, we can deduce the following corollary:

COROLLARY 2.13. For any completely positive operator $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$, $\text{cap}(T) > 0$ iff T^* is fractional-rank non-decreasing.

Similar to the square case, whenever a rectangular operator T is fractional-rank non-decreasing, there exists an operator scaling which is almost doubly stochastic.

THEOREM 2.14. Let $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ be a completely positive operator s.t. $\text{cap}(T) > 0$. Then for every $\epsilon > 0$, there exists an operator scaling T' of T s.t. $\text{ds}(T') \leq \epsilon$. Furthermore this T' can be found by running Algorithm 2 for an appropriate number of steps $t(T, \epsilon)$.

Algorithms for computing capacity of square operators can be used in a black box way to compute the capacity of rectangular operators due to Construction 2.10 and Lemma 2.10.

THEOREM 2.15. Let $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ be a completely positive operator s.t. the bit-complexity of the description of T (in terms of its Kraus operators) is b . Then there is a $\text{poly}(n_1, n_2, b)$ time (deterministic) algorithm to test whether T is fractional-rank non-decreasing. There is also a $\text{poly}(n_1, n_2, b, 1/\epsilon)$ time algorithm to compute a $(1 + \epsilon)$ -multiplicative approximation to $\text{cap}(T)$. Furthermore, the algorithm outputs a scaling T' of T which is almost doubly stochastic i.e. $\text{cap}(T') \geq 1 - \epsilon$.

Similarly, algorithms for computing witnesses to rank decreasing property of square operators can be used in a black box way to compute witnesses to fractional-rank decreasing property of rectangular operators.

THEOREM 2.16. Let $T : M_{n_1}(\mathbb{C}) \rightarrow M_{n_2}(\mathbb{C})$ be a completely positive operator s.t. the bit-complexity of the description of T (in terms of its Kraus operators) is b . Then there is a $\text{poly}(n_1, n_2, b)$ time (deterministic) algorithm to test whether T is fractional-rank non-decreasing. If T is fractional-rank decreasing, then the algorithm also outputs an $X \geq 0$ s.t. $\frac{\text{Rank}(T(X))}{n_2} < \frac{\text{Rank}(X)}{n_1}$.

3 REDUCTION FROM BRASCAMP-LIEB TO COMPLETELY POSITIVE OPERATORS

In this section we will show how each BL datum (\mathbf{B}, \mathbf{p}) corresponds to an operator $T_{(\mathbf{B}, \mathbf{p})}$ for which $\text{cap}(T_{(\mathbf{B}, \mathbf{p})}) = 1/\text{BL}(\mathbf{B}, \mathbf{p})^2$. From this correspondence, and from the connection between capacity and rank expansion of an operator, we will show how to derive the [11] conditions for finiteness of $\text{BL}(\mathbf{B}, \mathbf{p})$. Recall that in the Brascamp-Lieb setting, we are given a datum (\mathbf{B}, \mathbf{p}) consisting of m matrices B_1, \dots, B_m , where each B_i is a $n_i \times n$ matrix. We also have m non-negative rational numbers p_1, \dots, p_m s.t. $p_i = c_i/d$ for some $c_i, d \in \mathbb{N}$. Since one necessary condition for finiteness of the BL constant is that $n = \sum_{i=1}^m p_i n_i$, we will assume this equality throughout the section.¹²

We begin by recalling the characterization of the BL constant due to Lieb [40]:

THEOREM 3.1 (BRASCAMP-LIEB CONSTANT). Given a BL datum (\mathbf{B}, \mathbf{p}) , the Brascamp-Lieb constant $\text{BL}(\mathbf{B}, \mathbf{p})$ equals:

$$\text{BL}(\mathbf{B}, \mathbf{p}) = \sup_{\substack{X_i \in M_{n_i}(\mathbb{C}) \\ X_i > 0}} \left\{ \left(\frac{\prod_{i=1}^m \text{Det}(X_i)^{p_i}}{\text{Det} \left(\sum_{i=1}^m p_i B_i^\dagger X_i B_i \right)} \right)^{1/2} \right\}$$

We now describe our main reduction from the BL datum to (rectangular) completely positive operators.

CONSTRUCTION 3.2. Let $m' = \sum_{i=1}^m c_i$. We will construct $m' nd \times n$ matrices $A_1, \dots, A_{m'}$ from B_1, \dots, B_m . Essentially $A_1, \dots, A_{m'}$ will consist of c_i copies of B_i acting on different parts of the space. Consider

¹²This condition can be easily seen to hold by plugging in appropriate Gaussian inputs in the Brascamp-Lieb inequalities.

the following matrix A :

$$A = \begin{pmatrix} B_1 \\ \vdots \\ B_1 \\ B_2 \\ \vdots \\ B_2 \\ \vdots \\ \vdots \\ B_m \\ \vdots \\ B_m \end{pmatrix} \begin{matrix} \left. \vphantom{\begin{matrix} B_1 \\ \vdots \\ B_1 \\ B_2 \\ \vdots \\ B_2 \\ \vdots \\ \vdots \\ B_m \\ \vdots \\ B_m \end{matrix}} \right\} c_1 \text{ times} \\ \left. \vphantom{\begin{matrix} B_1 \\ \vdots \\ B_1 \\ B_2 \\ \vdots \\ B_2 \\ \vdots \\ \vdots \\ B_m \\ \vdots \\ B_m \end{matrix}} \right\} c_2 \text{ times} \\ \vdots \\ \left. \vphantom{\begin{matrix} B_1 \\ \vdots \\ B_1 \\ B_2 \\ \vdots \\ B_2 \\ \vdots \\ \vdots \\ B_m \\ \vdots \\ B_m \end{matrix}} \right\} c_m \text{ times} \end{matrix}$$

A has a natural block structure consisting of total m' blocks. The matrices A_j are obtained by keeping one block intact and zeroing out all other blocks. More formally, suppose j is such that $\sum_{i=1}^k c_i < j \leq \sum_{i=1}^{k+1} c_i$. Then all entries of A_j will be zero except rows

$$\begin{aligned} & \sum_{i=1}^k c_i n_i + \left(j - \sum_{i=1}^k c_i - 1 \right) \cdot n_{k+1} + 1 \text{ to} \\ & \sum_{i=1}^k c_i n_i + \left(j - \sum_{i=1}^k c_i \right) \cdot n_{k+1} \end{aligned}$$

which are occupied by B_{k+1} . Define the operator $T_{(\mathbf{B}, \mathbf{p})} : M_{nd}(\mathbb{C}) \rightarrow M_n(\mathbb{C})$ as

$$T_{(\mathbf{B}, \mathbf{p})}(X) = \sum_{j=1}^{m'} A_j^\dagger X A_j$$

Definition 3.3 (BL operator). The completely positive operator $T_{(\mathbf{B}, \mathbf{p})}$ constructed in Construction 3.2 is called the *Brascamp-Lieb operator* for datum (\mathbf{B}, \mathbf{p}) .

The next lemma relates important properties of a BL datum and its Brascamp-Lieb operator. Its proof can be found at the full version of this paper [28].

LEMMA 3.4. Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, where $p_i = c_i/d$ for $c_i, d \in \mathbb{N}$ and $B_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i}$. Moreover, assume that $n = \sum_{i=1}^m c_i n_i/d$. Let $T_{(\mathbf{B}, \mathbf{p})}$ be the Brascamp-Lieb operator corresponding to (\mathbf{B}, \mathbf{p}) . Then

- (1) If $\text{BL}(\mathbf{B}, \mathbf{p}) < \infty$, then $\text{cap}(T_{(\mathbf{B}, \mathbf{p})}) = 1/\text{BL}(\mathbf{B}, \mathbf{p})^2$
- (2) The inequality $\dim(V) \leq \sum_j p_j \dim(B_j V)$ holds for all subspaces $V \subseteq \mathbb{R}^n$ iff $T_{(\mathbf{B}, \mathbf{p})}^*$ is fractional-rank non-decreasing.

Moreover, if B_i 's are integer matrices such that $\|B_i\|_\infty \leq M$, then the same holds for the matrices $A_1, \dots, A_{m'}$.

With Construction 3.2 and Lemma 3.4, we can deduce the following corollaries for the BL datum. The next corollary is the Theorem 1.1 from the introduction proved in [11].

COROLLARY 3.5. Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, where $p_i = c_i/d$ for $c_i, d \in \mathbb{N}$ and $B_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i}$. Moreover, assume that $n =$

$\sum_{i=1}^m c_i n_i/d$. Then $\text{BL}(\mathbf{B}, \mathbf{p})$ is finite iff for every subspace $V \subseteq \mathbb{R}^n$, it holds that

$$\dim(V) \leq \sum_{i=1}^m p_i \dim(B_i(V))$$

COROLLARY 3.6. Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, where $p_i = c_i/d$ for $c_i, d \in \mathbb{N}$ and $B_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i}$. Moreover, assume that $n = \sum_{i=1}^m c_i n_i/d$. Let b be the total bit size of \mathbf{B} . Then there is a poly(b, d) time (deterministic) algorithm to check if $\text{BL}(\mathbf{B}, \mathbf{p})$ is finite. Furthermore, if $\text{BL}(\mathbf{B}, \mathbf{p})$ is not finite, then the algorithm outputs a subspace $V \subseteq \mathbb{R}^n$ s.t.

$$\dim(V) > \sum_{i=1}^m p_i \dim(B_i(V))$$

4 UPPER BOUND ON THE BRASCAMP-LIEB CONSTANT

In this section, we will prove an exponential upper bound on the BL constant $\text{BL}(\mathbf{B}, \mathbf{p})$ (Theorem 1.4). The bulk of the work is done in the first subsection, where we derive a lower bound on the capacity of square completely positive operators with integer entries, which is both stronger and has a simpler proof than the corresponding bound we proved in [29]. In the second subsection we derive the upper bound on the BL constant, with arbitrary rational entries, via its connection to capacity established in our main reduction. All proofs of this section can be found at the full version of the paper [28].

4.1 Better Lower Bound on the Capacity of Integral Operators

Before diving into the proof we provide a high level plan. We want to prove that if a completely positive operator with integer entries has positive capacity, then the capacity is actually non-negligible. Our starting point (Theorem 4.1) is the statement of the improved degree upper bounds from invariant theory, for the Kronecker (Left-Right) action on tuples of matrices. This mouthful is just a simple linear algebraic condition, of *low degree*, satisfied by general complex-valued operators of positive capacity. Using Alon's Combinatorial Nullstellensatz this low degree translates also into small coefficients in the linear algebraic condition above. We then state and prove (Lemma 4.4) a Cauchy-Schwartz inequality for matrices needed next. The main result (Theorem 4.5) proceeds by applying an appropriate scaling to the original integral operator, which reduces it to one that preserves the identity matrix. The latter property (which provides a trace bound), the integrality of the original operator (which provides an integral lower determinantal bound) and the multiplicativity of capacity under scaling combine (via the above inequality) to give the desired bound. We now proceed with the details of this plan.

The following theorem is the essence of the new degree bounds [21, 35] on the semi-invariants for the left-right action.

THEOREM 4.1 (DEGREE BOUNDS [21, 35]). If A_1, \dots, A_m are $n \times n$ complex matrices s.t. the completely positive operator T_A defined by A_1, \dots, A_m has positive capacity, then there exist matrices $D_1, \dots, D_m \in M_n(\mathbb{C})$ such that

$$\text{Det}(A_1 \otimes D_1 + \dots + A_m \otimes D_m) \neq 0.$$

The previous degree bounds on (the dimension of) matrices D_i were exponential in n [20, 36], and this is one source of this improved analysis. Another source of the improved analysis of capacity is the fact that (nonzero) polynomials of degree d cannot vanish on all points with non-negative integer coordinates with sum $\leq d$. This follows from Alon's combinatorial nullstellensatz [1, Theorem 1.2].

LEMMA 4.2. *If $p(x_1, \dots, x_n) \in \mathbb{C}[x_1, \dots, x_n]$ is a (nonzero) polynomial of degree d , where the individual degree of the variable x_i is at most d_i , then there exists $(a_1, \dots, a_n) \in \mathbb{Z}_{\geq 0}^n$ such that $\sum_{i=1}^n a_i \leq d$ and $a_i \leq d_i$, for which $p(a_1, \dots, a_n) \neq 0$.*

As a corollary of Theorem 4.1 and of Lemma 4.2, we get the following:

COROLLARY 4.3. *Let A_1, \dots, A_m be $n \times n$ complex matrices s.t. the completely positive operator T_A defined by A_1, \dots, A_m has positive capacity. Then there exist matrices $D_1, \dots, D_m \in \mathcal{M}_n(\mathbb{C})$ such that*

$$\text{Det}(A_1 \otimes D_1 + \dots + A_m \otimes D_m) \neq 0.$$

Furthermore, D_1, \dots, D_m can be chosen to be matrices with non-negative integer entries s.t.

$$\sum_{j=1}^m \sum_{k,l=1}^n D_j(k,l)^2 \leq n^3$$

We will also need the following Cauchy-Schwarz like inequality, whose proof can be found at [28]:

LEMMA 4.4. *Let C_1, \dots, C_m and D_1, \dots, D_m be n dimensional complex matrices. Then*

$$\begin{aligned} \text{tr} \left[\left(\sum_{i=1}^m C_i \otimes D_i \right) \left(\sum_{j=1}^m C_j^\dagger \otimes D_j^\dagger \right) \right] &\leq \\ &\leq \text{tr} \left[\sum_{i=1}^m C_i C_i^\dagger \right] \cdot \text{tr} \left[\sum_{j=1}^m D_j D_j^\dagger \right] \end{aligned}$$

The following theorem (Theorem 2.21) is from [29]. We give a simpler proof of it in the full version of the paper [28], and we also obtain better bounds there (mostly due to the remarkable progress in degree bounds).

THEOREM 4.5 (CAPACITY OF SQUARE OPERATORS). *Suppose T_A is a completely positive operator which has positive capacity and has Kraus operators $A_1, \dots, A_m \in \mathcal{M}_n(\mathbb{Z})$. In this case:*

$$\text{cap}(T_A) \geq \exp(-2n \log(n))$$

4.2 An Exponential Upper Bound on the BL-constant

We now derive the upper bound on the BL-constant from the lower bound on capacity above via our main reduction. We do so first for BL-data with integer entries (which is independent of their magnitude!), and then for general ones with rational entries.

THEOREM 4.6 (LOWER BOUND ON BRASCAMP-LIEB CAPACITY). *Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, with $B_i \in \mathcal{M}_{n_i \times n}(\mathbb{Z})$ and $p_i = c_i/d$, where $c_i, d \in \mathbb{N}$. Let $T_{(\mathbf{B}, \mathbf{p})}$ be the Brascamp-Lieb*

operator, as defined in Construction 3.2. If $T_{(\mathbf{B}, \mathbf{p})}$ has positive capacity, then the following bound holds:

$$\text{cap}(T_{(\mathbf{B}, \mathbf{p})}) \geq \exp(-2n \log(n^2 d)).$$

From the theorem above we obtain the following upper bound on the Brascamp-Lieb constant:

COROLLARY 4.7 (UPPER BOUND ON BL CONSTANT). *Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, with $B_i \in \mathcal{M}_{n_i \times n}(\mathbb{Z})$ and $p_i = c_i/d$, where $c_i, d \in \mathbb{N}$. If $\text{BL}(\mathbf{B}, \mathbf{p}) < \infty$, then the following bound holds:*

$$\text{BL}(\mathbf{B}, \mathbf{p}) \leq \exp(4n \log(n^2 d)).$$

As a corollary, we can also get an exponential bound that is independent of d .

COROLLARY 4.8. *Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, with $B_i \in \mathcal{M}_{n_i \times n}(\mathbb{Z})$ and p_1, \dots, p_m arbitrary positive reals. If $\text{BL}(\mathbf{B}, \mathbf{p}) < \infty$, then the following bound holds:*

$$\text{BL}(\mathbf{B}, \mathbf{p}) \leq \exp(12nm \log(mn)).$$

From the bound on the BL constant for integer datum, we can also get a bound for arbitrary rational datum in terms of the bit sizes (which is a restatement of Theorem 1.4 in the introduction.).

COROLLARY 4.9. *Let (\mathbf{B}, \mathbf{p}) be a Brascamp-Lieb datum, with $B_i \in \mathcal{M}_{n_i \times n}(\mathbb{Q})$ and $p_i = c_i/d$, where each entry of B_i has bit-size at most \bar{b} and $c_i, d \in \mathbb{N}$. If $\text{BL}(\mathbf{B}, \mathbf{p}) < \infty$, then the following bound holds:*

$$\text{BL}(\mathbf{B}, \mathbf{p}) \leq \exp(4n \log(n^2 d) + n\bar{b}).$$

REMARK 4.10. *As before, d can be eliminated from the above corollary if needed but the bound in the exponential would become quadratic.*

5 COMPUTING THE BRASCAMP-LIEB CONSTANT

In this section we show how Algorithm G with truncation, described in [29, 30], computes the Brascamp-Lieb constant up to a multiplicative factor of $(1 \pm \epsilon)$. We begin by restating Theorem 3.5 from [29], which states that we can approximate the capacity of a completely positive operator.

THEOREM 5.1 (THEOREM 3.5 FROM [29]). *Let T be a completely positive operator, whose Kraus operators are given by $n \times n$ matrices $A_1, \dots, A_m \in \mathbb{Z}^{n \times n}$, such that each entry of A_i has absolute value at most M . Algorithm G, with truncation parameter*

$$P(n, 1/\epsilon, \log(M)) = \frac{1}{\epsilon} \cdot (n^{12} \log^4(Mn)) \cdot \log(n^4/\epsilon^2)$$

when applied for $t = \frac{4n^3}{\epsilon^2} \cdot (1 + 10n^2 \log(Mn))$ steps approximates $\text{cap}(T)$ within a multiplicative factor of $1 \pm \epsilon$.

Theorem 5.1 tells us how to approximate the capacity of an operator. Construction 3.2 gives us an operator whose capacity is the inverse square of the Brascamp-Lieb constant. Hence, to approximate the Brascamp-Lieb constant corresponding to datum (\mathbf{B}, \mathbf{p}) , we need only obtain operator $T_{(\mathbf{B}, \mathbf{p})}$ and use algorithm G to compute

an approximation to $\text{cap}(T_{(\mathbf{B}, \mathbf{p})})$. Then, $\text{BL}(\mathbf{B}, \mathbf{p}) = \left(\frac{1}{\text{cap}(T_{(\mathbf{B}, \mathbf{p})})} \right)^{1/2}$.

This yields the following algorithm, and Theorem 5.2 (from which Theorem 1.5 in the introduction follows by a simple scaling argument), whose proof can be found at [28]:

Input: BL datum (\mathbf{B}, \mathbf{p}) , where each $p_i = c_i/d$, with $c_i, d \in \mathbb{Z}, d \neq 0, B_i \in \mathbb{Z}^{n_i \times n}$ and approximation parameter $\epsilon > 0$. Moreover, $\|B_i\|_\infty \leq M$ for all i .

Output: $\text{BL}(\mathbf{B}, \mathbf{p})$ with multiplicative error of $(1 \pm \epsilon)$.

- (1) Check if $n = \sum_{i=1}^m p_i n_i$. If yes, go to step 2. If not, return ∞ .
- (2) Let $T_{(\mathbf{B}, \mathbf{p})}$ from Construction 3.2. Let $\tilde{T}_{(\mathbf{B}, \mathbf{p})}$ be the operator obtained from $T_{(\mathbf{B}, \mathbf{p})}$ via Construction 2.9. Go to step 3.
- (3) Apply Algorithm G on $\tilde{T}_{(\mathbf{B}, \mathbf{p})}$, with approximation parameter ϵ . Let α be the output of Algorithm G .
- (4) Return $(1/\alpha)^{1/2nd}$.

Algorithm 3: Algorithm \tilde{G} computing the BL constant

THEOREM 5.2 (APPROXIMATING THE BRASCAMP-LIEB CONSTANT). *Given a Brascamp-Lieb datum (\mathbf{B}, \mathbf{p}) , where $p_i = c_i/d$, with $c_i, d \in \mathbb{Z}, d \neq 0, B_i \in \mathbb{Z}^{n_i \times n}$ and approximation parameter $\epsilon > 0$. Each entry of B_i has absolute value at most M . Algorithm \tilde{G} runs in time $\text{poly}(n, d, \log M, 1/\epsilon)$ and computes $\text{BL}(\mathbf{B}, \mathbf{p})$ to a multiplicative approximation factor of $(1 \pm \epsilon)$.*

6 CONTINUITY OF THE BRASCAMP-LIEB CONSTANT

In this section, we note that the explicit bounds on the continuity of capacity of completely positive operators (at rational points) in [29] directly translate to bounds on the continuity of Brascamp-Lieb constants (at rational points). We will then discuss how this can be extended to irrational points.

We start by proving a weaker version of Theorem 1.12. The following theorem is proved in [29].

THEOREM 6.1 (THEOREM 2.32 FROM [29]). *Suppose A_1, \dots, A_m and B_1, \dots, B_m are two tuples of $n \times n$ matrices such that the bit-complexity of elements of A_i 's and B_i 's is b and $\|A_i - B_i\| \leq \delta$ for all i . Let T_A be the operator defined by A_1, \dots, A_m and T_B be the operator defined by B_1, \dots, B_m . Then there exists a polynomial $P(n, b, \log(m))$ s.t. if $\delta \leq \exp(-P(n, b, \log(m)))$, then $\text{cap}(T_A) > 0$ iff $\text{cap}(T_B) > 0$. Furthermore*

$$\left(1 - \frac{P(n, b, \log(m))}{\log(1/\delta)^{1/3}}\right) \leq \frac{\text{cap}(T_A)}{\text{cap}(T_B)} \leq \left(1 + \frac{P(n, b, \log(m))}{\log(1/\delta)^{1/3}}\right)$$

Note that the above theorem doesn't imply continuity of capacity at rational points since we have assumed bounds on the bit-sizes of B_1, \dots, B_m . As an immediate corollary of the reduction from Brascamp-Lieb datum to completely positive operators (Construction 3.2) and the theorem above, we get the following:

THEOREM 6.2. *Suppose $(B_1, \dots, B_m, p_1, \dots, p_m)$ and $(\tilde{B}_1, \dots, \tilde{B}_m, p_1, \dots, p_m)$ are two tuples of BL datum. Let (n, n_1, \dots, n_m) be the dimension vector and $p_i = c_i/d$ for $1 \leq i \leq m$. Also suppose the bit-complexity of elements of B_i 's and \tilde{B}_i 's is b and $\|B_i - \tilde{B}_i\| \leq \delta$ for all i . Then there exists a polynomial $P(n, d, b, \log(m))$ s.t. if $\delta \leq$*

$\exp(-P(n, d, b, \log(m)))$, then $\text{BL}(\mathbf{B}, \mathbf{p})$ is finite iff $\text{BL}(\tilde{\mathbf{B}}, \mathbf{p})$ is finite. Furthermore

$$\left(1 - \frac{P(n, d, b, \log(m))}{\log(1/\delta)^{1/3}}\right) \leq \frac{\text{BL}(\mathbf{B}, \mathbf{p})}{\text{BL}(\tilde{\mathbf{B}}, \mathbf{p})} \leq \left(1 + \frac{P(n, d, b, \log(m))}{\log(1/\delta)^{1/3}}\right)$$

Again, the above theorem doesn't imply continuity of the BL constant at \mathbf{B} since we assume a bound on the bit sizes of \mathbf{B}' . Nonetheless, the proof of Theorem 2.32 in [29] can be modified to yield the following theorem (continuity of capacity at rational points):

THEOREM 6.3. *Suppose A_1, \dots, A_m and B_1, \dots, B_m are two tuples of $n \times n$ matrices such that the bit-complexity of elements of A_i 's is b and $\|A_i - B_i\| \leq \delta$ for all i . Let T_A be the operator defined by A_1, \dots, A_m and T_B be the operator defined by B_1, \dots, B_m . Then there exists a polynomial $P(n, b, \log(m))$ s.t. if $\delta \leq \exp(-P(n, b, \log(m)))$ and $\text{cap}(T_A) > 0$, then $\text{cap}(T_B) > 0$. Furthermore*

$$\left(1 - \frac{P(n, b, \log(m))}{\log(1/\delta)^{1/3}}\right) \leq \frac{\text{cap}(T_A)}{\text{cap}(T_B)} \leq \left(1 + \frac{P(n, b, \log(m))}{\log(1/\delta)^{1/3}}\right)$$

From this we get the following theorem (continuity of BL constant at rational points) as an immediate corollary, again due to Construction 3.2. More details can be found in the full version [28].

THEOREM 6.4 (THEOREM 1.12 RESTATED). *Suppose that (\mathbf{B}, \mathbf{p}) and $(\tilde{\mathbf{B}}, \mathbf{p})$ are two BL data. Let (n, n_1, \dots, n_m) be the dimension vector and $p_i = c_i/d$. Also suppose the bit-complexity of elements of B_i 's is b and $\|B_i - \tilde{B}_i\| \leq \delta$ for all i . Then there exists a polynomial $P(n, d, b, \log(m))$ s.t. if $\delta \leq \exp(-P(n, d, b, \log(m)))$ and $\text{BL}(\mathbf{B}, \mathbf{p})$ is finite, then $\text{BL}(\tilde{\mathbf{B}}, \mathbf{p})$ is finite. Furthermore*

$$\left(1 - \frac{P(n, d, b, \log(m))}{\log(1/\delta)^{1/3}}\right) \leq \frac{\text{BL}(\mathbf{B}, \mathbf{p})}{\text{BL}(\tilde{\mathbf{B}}, \mathbf{p})} \leq \left(1 + \frac{P(n, d, b, \log(m))}{\log(1/\delta)^{1/3}}\right)$$

Continuity at irrational points. We note that Theorem 6.3 can be extended to continuity of capacity at irrational A_1, \dots, A_m (without any explicit bounds). The changes that need to be made to the proof are outlined in the full version [28].

When \mathbf{p} is not rational, our reduction from BL datum to operators doesn't work anymore. Nonetheless, Algorithm 1 described in the introduction can be applied directly to the BL datum. Since this algorithm is continuous in \mathbf{B} , this should imply continuity of $\text{BL}(\mathbf{B}, \mathbf{p})$ in \mathbf{B} . However, it is not clear to us at the moment how long this algorithm will have to be run before the BL constant gets sufficiently close to 1. This is because the progress per step in Theorem 1.9 depends on the common denominator d of \mathbf{p} , which doesn't make sense in the case when \mathbf{p} is not rational. But perhaps the fact that the vertices of the BL polytope $P_{\mathbf{B}}$ are rational can come in handy.

7 INTERESTING SPECIAL CASES OF BL POLYTOPES

In this section, we explore special cases of BL polytopes that give rise to natural polytopes which are well studied in the combinatorial optimization literature. As mentioned before, Barthe [6] proved

that the BL polytopes corresponding to rank-1 BL datum are exactly the basis polytopes.

Now we show that some special cases of BL polytopes corresponding to rank-2 BL datum give rise to linear matroid intersection polytopes. Given a collection of vectors in \mathbb{R}^n , $\mathbf{v} = (v_1, \dots, v_m)$, the linear matroid $\mathcal{M}_{\mathbf{v}}$ associated with this collection (over the ground set $[m]$) is $\{I \subseteq [m] : (v_i)_{i \in I} \text{ are linearly independent}\}^{13}$.

Given two linear matroids $\mathcal{M}_{\mathbf{v}}$ and $\mathcal{M}_{\mathbf{w}}$, their intersection polytope $P_{\mathcal{M}_{\mathbf{v}}, \mathcal{M}_{\mathbf{w}}}$ is the convex hull of the characteristic vectors of their common bases (sometimes this is defined as the convex hull of the common independent sets) i.e.

$$P_{\mathcal{M}_{\mathbf{v}}, \mathcal{M}_{\mathbf{w}}} = \text{conv} \{1_I : (v_i)_{i \in I}, (w_i)_{i \in I} \text{ each basis of } \mathbb{R}^n\}$$

For a linear matroid $\mathcal{M}_{\mathbf{v}}$ and a set $J \subseteq [m]$, we define the subspace

$$V_J = \text{span}((v_j)_{j \in J})$$

It is well known (e.g. see Chapter 41 in [46]) $P_{\mathcal{M}_{\mathbf{v}}, \mathcal{M}_{\mathbf{w}}}$ can be described by the following linear program (with exponentially many inequalities!) on a variable vector \mathbf{p} :

$$\begin{aligned} \sum_{j=1}^m p_j &= n \\ \sum_{j \in J} p_j &\leq \dim(V_J) \quad \forall J \subseteq [m] \end{aligned} \tag{1}$$

$$\begin{aligned} \sum_{j \in J'} p_j &\leq \dim(W_{J'}) \quad \forall J' \subseteq [m] \\ \mathbf{p} &\geq 0 \end{aligned} \tag{2}$$

Now consider the following BL datum: for every $j \in [m]$, $B_j : \mathbb{R}^{2n} \rightarrow \mathbb{R}^2$ is given by

$$B_j(x, y) = (\langle v_j, x \rangle, \langle w_j, y \rangle)$$

We will show that the BL polytope $P_{\mathbf{B}}$ corresponding to this BL datum is the same as the matroid intersection polytope $P_{\mathcal{M}_{\mathbf{v}}, \mathcal{M}_{\mathbf{w}}}$. Our feasibility algorithm for BL-polytopes thus automatically gives a polynomial time algorithm for the linear matroid intersection problem (here over \mathbb{R}). Of course it was already known that this problem is in P , and moreover Gurvits [30] already noticed that operator scaling efficiently solves this problem; our hope is that such encodings are possible for other optimization problems with exponential size linear programs. A proof of the following lemma can be found in the full version of the paper [28].

LEMMA 7.1. *The BL polytope $P_{\mathbf{B}}$ corresponding to the above BL datum is the same as $P_{\mathcal{M}_{\mathbf{v}}, \mathcal{M}_{\mathbf{w}}}$.*

Since the perfect bipartite matching polytopes are a special case of linear matroid intersection polytopes, we see that the bipartite matching polytopes are a special case of BL polytopes for rank-2 BL datum. As mentioned before, it is an interesting question if the general perfect matching polytopes are a special case of BL polytopes or not.

¹³Linear matroids can be defined w.r.t. any field \mathbb{F} but we will restrict ourselves to \mathbb{R} as this is the case which embeds into BL.

8 CONCLUSION AND OPEN QUESTIONS

In this paper we address some of the central algorithmic questions regarding Brascamp-Lieb inequalities. In particular, we gave polynomial time algorithms to test membership in the BL polytopes, compute a violating inequality (separation oracle) and approximate the BL constants. These algorithms were obtained by a simple reduction from the BL data to completely positive operators and then using recent polynomial time algorithms that test whether such operators are rank decreasing and compute the capacity of such operators [29, 35]. The reduction itself provides in some cases different proofs of some of the known structural results regarding BL inequalities, in particular the characterization of feasibility.

On the algorithmic side, our work leaves some concrete open questions.

- (1) While our reduction is polynomial time in the binary description length of the matrices (\mathbf{B}) in the BL datum, it is not polynomial time in the binary description length of the exponent vector \mathbf{p} . When $p_i = \frac{c_i}{d}$, the running times of our algorithms scale as $\text{poly}(d)$. Can one improve them to get (the optimal) $\text{poly}(\log(d))$?
- (2) Our algorithm for approximating the BL constants up to an $(1 + \epsilon)$ factor runs in time $\text{poly}(1/\epsilon)$. Can one improve it to $\text{poly}(\log(1/\epsilon))$?
- (3) Can one efficiently optimize linear functions over BL polytopes?

On the structural side, our reduction may provide a higher level view of BL inequalities, simply since BL data is just a proper subset of completely positive operators, and viewing them as such may lead to better understanding. One intriguing question is whether this viewpoint leads to an even larger family of inequalities. It is obvious that for any doubly stochastic operator T , the fact that $\text{cap}(T) = 1$ implies the following determinantal inequality (that holds for all psd matrices X),

$$\text{Det} \left(\sum_{i=1}^m A_i X A_i^\dagger \right) = \text{Det}(T(X)) \geq \text{Det}(X)$$

which can be translated to an inequality on Gaussian densities. One can wonder for which operators T the Gaussian densities can be replaced by arbitrary non-negative densities, as in BL-inequalities. There are simple counterexamples showing this does not hold in full generality. On the other other hand, our reduction and the connection to quiver representation (Section ??) tells us that for operators T arising from the star quiver, these inequalities are in 1-1 correspondence to the BL-inequalities. It would be interesting to better understand what is special about star quivers in this context. In particular, do other quivers lead to new inequalities?

ACKNOWLEDGMENTS

We would like to thank Zeev Dvir for pointing us to the paper [11]. We would also like to thank Jonathan Bennett and Larry Guth for helpful comments. We thank Rohit Gurjar for suggesting that BL polytopes can encode the linear matroid intersection polytopes.

Ankit Garg's research was done when the author was a student at Princeton University. His research was supported by Mark Braverman's NSF grant CCF-1149888, Simons Collaboration on

Algorithms and Geometry, Simons Fellowship in Theoretical Computer Science and Siebel Scholarship. Rafael Oliveira's research was supported by Zeev Dvir's NSF CAREER award DMS-1451191, NSF grant CCF-1523816 and Siebel Scholarship. Avi Wigderson's research was supported by NSF grant CCF-1412958.

REFERENCES

- [1] Noga Alon. 1999. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing* 8, 1-2 (1999), 7–29.
- [2] Keith Ball. 1989. Volumes of sections of cubes and related problems. *Geometric Aspects of Functional Analysis* (1989), 251–260.
- [3] Boaz Barak, Zeev Dvir, Avi Wigderson, and Amir Yehudayoff. 2012. Fractional Sylvester-Gallai theorems. *Proceedings of the National Academy of Sciences* (2012).
- [4] Boaz Barak, Jonathan A Kelner, and David Steurer. 2014. Rounding sum-of-squares relaxations. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. ACM, 31–40.
- [5] Boaz Barak, Jonathan A Kelner, and David Steurer. 2015. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. ACM, 143–151.
- [6] Franck Barthe. 1998. On a reverse form of the Brascamp–Lieb inequality. *Inventiones Mathematicae* 134 (1998), 335–361.
- [7] Jonathan Bennett and Neal Bez. 2010. Some nonlinear Brascamp–Lieb inequalities and applications to harmonic analysis. *Journal of Functional Analysis* 259, 10 (2010), 2520–2556.
- [8] Jonathan Bennett, Neal Bez, Michael Cowling, and Taryn Flock. 2016. Behaviour of the Brascamp-Lieb constant. arXiv preprint arXiv:1605.08603 (2016).
- [9] Jonathan Bennett, Neal Bez, Taryn Flock, and Sanghyuk Lee. 2015. Stability of the Brascamp-Lieb constant and applications. arXiv preprint arXiv:1508.07502 (2015).
- [10] Jonathan Bennett, Anthony Carbery, Michael Christ, and Terence Tao. 2005. FINITE BOUNDS FOR Holder-Brascamp-Lieb MULTILINEAR INEQUALITIES. arXiv preprint arXiv:0505691 (2005).
- [11] Jonathan Bennett, Anthony Carbery, Michael Christ, and Terence Tao. 2008. The Brascamp-Lieb inequalities: finiteness, structure, and extremals. *Geometric and Functional Analysis* 17, 5 (2008), 1343–1415.
- [12] Jean Bourgain and Ciprian Demeter. 2015. Mean value estimates for Weyl sums in two dimensions. arXiv preprint arXiv:1509.05388 (2015).
- [13] Jean Bourgain, Ciprian Demeter, and Larry Guth. 2015. Proof of the main conjecture in Vinogradov's mean value theorem for degrees higher than three. *Annals of Mathematics* (2015).
- [14] Herm Brascamp and Elliot Lieb. 1976. Best Constants in Young's Inequality, Its Converse and Its Generalization to More Than Three Functions. *Advances in Mathematics* 20 (1976), 151–172.
- [15] Eric Carlen and Dario Cordero-Erausquin. 2009. Subadditivity of The Entropy and its Relation to Brascamp-Lieb Type Inequalities. *Functional Analysis* 19 (2009), 373.
- [16] M. Choi. 1975. Completely positive linear maps on complex matrices. *Linear Algebra and Its Applications* (1975), 285–290.
- [17] Michael Christ, James Demmel, Nicholas Knight, Thomas Scanlon, and Katherine Yelick. 2013. Communication lower bounds and optimal algorithms for programs that reference arrays - Part 1. arXiv preprint arXiv:1308.0068 (2013).
- [18] Michael Christ, James Demmel, Nicholas Knight, Thomas Scanlon, and Katherine Yelick. 2015. On Holder-Brascamp-Lieb inequalities for torsion-free discrete Abelian groups. arXiv preprint arXiv:1510.04190 (2015).
- [19] Imre Csiszár and Gábor Tusnády. 1984. Information geometry and alternating minimization procedures. *Stat. Decis. Suppl.* 1 (1984), 205–237.
- [20] Harm Derksen. 2001. Polynomial bounds for rings of invariants. *Proc. Amer. Math. Soc.* 129, 4 (2001), 955–964.
- [21] Harm Derksen and Visu Makam. 2015. Polynomial degree bounds for matrix semi-invariants. arXiv preprint arXiv:1512.03393 (2015).
- [22] Harm Derksen and Jerzy Weyman. 2007. The combinatorics of quiver representations. arXiv preprint arXiv:0608288 (2007).
- [23] Zeev Dvir, Ankit Garg, Rafael Oliveira, and József Solymosi. 2016. Rank bounds for design matrices with block entries and geometric applications. arXiv preprint arXiv:1610.08923 (2016).
- [24] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. 2014. Breaking the Quadratic Barrier for 3-LCC's over the Reals. *STOC* (2014).
- [25] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. 2014. IMPROVED RANK BOUNDS FOR DESIGN MATRICES AND A NEW PROOF OF KELLY'S THEOREM. *Forum of Mathematics, Sigma* 2 (2014), 24.
- [26] Jack Edmonds. 1965. Maximum matching and a polyhedron with 0,1-vertices. *Journal of research of the National Bureau of Standards* 69B (1965), 125–130.
- [27] Jürgen Forster. 2002. A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. System Sci.* 65, 4 (2002), 612–625.
- [28] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. 2016. Algorithmic aspects of Brascamp-Lieb inequalities. arXiv preprint arXiv:1607.06711 (2016).
- [29] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. 2016. A deterministic polynomial time algorithm for non-commutative rational identity testing. *FOCS* arXiv:1511.03730 (2016).
- [30] Leonid Gurvits. 2004. Classical complexity and quantum entanglement. *J. Comput. System Sci.* 69, 3 (2004), 448–484.
- [31] Leonid Gurvits and Alex Samorodnitsky. 2000. A deterministic polynomial-time algorithm for approximating mixed discriminant and mixed volume. *STOC* (2000).
- [32] Moritz Hardt. 2014. Understanding Alternating Minimization for Matrix Completion. *FOCS* (2014), 651–660.
- [33] Moritz Hardt and Ankur Moitra. 2013. Algorithms and Hardness for Robust Subspace Recovery. *COLT* (2013).
- [34] Samuel B Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. 2015. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors. arXiv preprint arXiv:1512.02337 (2015).
- [35] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. 2015. Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics. arXiv preprint arXiv:1512.03531 (December 2015).
- [36] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. 2015. Non-commutative Edmonds' problem and matrix semi-invariants. <http://arxiv.org/abs/1508.00690> (August 2015).
- [37] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. 2013. Low-rank matrix completion using alternating minimization. *STOC* (2013), 665–674.
- [38] Lei and Sheng. 2016. Personal Communication. (2016).
- [39] Carlton Lemke and J. T. Howson. 1964. Equilibrium points of bimatrix games. *SIAM J. Appl. Math.* 12, 2 (1964), 413–423.
- [40] Elliot Lieb. 1990. Gaussian Kernels have only Gaussian Maximizers. *Inventiones Mathematicae* 102 (1990), 179–208.
- [41] Jingbo Liu, Thomas Courtarde, Paul Cuff, and Sergio Verdú. 2016. Smoothing Brascamp-Lieb Inequalities and Strong Converses for Common Randomness Generation. *ISIT* (2016).
- [42] Tengyu Ma, Jonathan Shi, and David Steurer. 2016. Polynomial-time tensor decompositions with sum-of-squares. arXiv preprint arXiv:1610.01980 (2016).
- [43] Ketan Mulmuley. 2012. Geometric Complexity Theory V: Equivalence between Blackbox Derandomization of Polynomial Identity Testing and Derandomization of Noether's Normalization Lemma. *FOCS* (2012), 629–638.
- [44] Ketan Mulmuley and Milind Sohoni. 2006. Geometric Complexity Theory I: An Approach to the P. vs. NP and related problems. *SIAM J. Comput* 31, 2 (2006), 496–526.
- [45] Aleksandar Nikolov and Mohit Singh. 2016. Maximizing Determinants under Partition Constraints. *STOC* (2016).
- [46] Alexander Schrijver. 2003. *Combinatorial Optimization*. Vol. B. Springer.
- [47] R. Sinkhorn. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics* 35 (1964), 876–879.
- [48] Stefan Valdimarsson. 2010. The Brascamp-Lieb Polyhedron. *Canadian Journal of Mathematics* 62 (2010), 870–888.
- [49] Stefanos Zafeiriou and Maria Petrou. 2011. Nonnegative tensor factorization as an alternative Csiszár–Tusnády procedure: algorithms, convergence, probabilistic interpretations and novel probabilistic tensor latent variable analysis algorithms. *Data Mining and Knowledge Discovery* 22, 3 (2011), 419–466.