# Operator Scaling via Geodesically Convex Optimization, Invariant Theory and Polynomial Identity Testing*

Zeyuan Allen-Zhu
Microsoft Research AI
zeyuan@csail.mit.edu

Ankit Garg
Microsoft Research New England
garga@microsoft.com

Yuanzhi Li
Princeton University
yuanzhil@cs.princeton.edu

Rafael Oliveira
University of Toronto
rafael@cs.toronto.edu

Avi Wigderson
Institute for Advanced Study
avi@math.ias.edu

## ABSTRACT

We propose a new second-order method for geodesically convex optimization on the natural hyperbolic metric over positive definite matrices. We apply it to solve the *operator scaling* problem in time polynomial in the input size and *logarithmic* in the error. This is an exponential improvement over previous algorithms which were analyzed in the usual Euclidean, "commutative" metric (for which the above problem is not convex). Our method is general and applicable to other settings.

As a consequence, we solve the *equivalence* problem for the left-right group action underlying the operator scaling problem. This yields a *deterministic* polynomial-time algorithm for a new class of Polynomial Identity Testing (PIT) problems, which was the original motivation for studying operator scaling.

## CCS CONCEPTS

• **Theory of computation** → **Convex optimization**; **Nonconvex optimization**; **Pseudorandomness and derandomization**;

## KEYWORDS

operator scaling, geodesic convex, positive definite, orbit-closure intersection, capacity

**ACM Reference Format:**

Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. 2018. Operator Scaling via Geodesically Convex Optimization, Invariant Theory and Polynomial Identity Testing. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3188745.3188942

*Future version of this paper can be found on arXiv: https://arxiv.org/abs/1804.01076.


*STOC'18, June 25–29, 2018, Los Angeles, CA, USA*

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188942

## 1 INTRODUCTION

Group orbits and their closures capture natural notions of *equivalence* and are studied in several fields of mathematics like group theory, invariant theory and algebraic geometry. They also come up naturally in theoretical computer science. For example, graph isomorphism, the $\mathcal{VP}$ vs $\mathcal{VNP}$ question and lower bounds on tensor rank are all questions about such notions of equivalence.

In this paper, we focus on the *orbit-closure intersection* problem, which is the most natural way to define equivalence for continuous group actions. We explore a general approach to the problem via geodesically convex optimization. As a testbed for our techniques, we design a deterministic polynomial-time algorithm for the orbit-closure intersection problem for the left-right group action. Recent results by [22, 46] have reduced this problem to polynomial identity testing (PIT), which yields a randomized polynomial-time algorithm. We derandomize this special case of PIT, perhaps surprisingly, by continuous optimization.

On the optimization side, we propose a new second-order method for geodesically convex optimization and use it to get an algorithm for *operator scaling* with time polynomial in the input bit size and *poly-logarithmic* in $1/\varepsilon$ ($\varepsilon$ is the error). In contrast, prior work [32] gives an operator-scaling algorithm that runs in time only *polynomial* in $1/\varepsilon$, which is not sufficient for an application to the general orbit-closure intersection problem.

On the PIT side, we have continued the line of research initiated by Mulmuley [59] to the study of problems in algebraic geometry and invariant theory from an algorithmic perspective in order to develop and sharpen tools to attack the PIT problem. Our result adds to the growing list of this agenda [29, 32, 45, 46], and continues the paper [32] in building optimization tools for PIT problems (at least for those arising from invariant theory). Could it be possible that the eventual solution to PIT will lie in optimization (perhaps very wishful thinking)?

Below is an outline of the rest of the introduction. We review geodesically convex optimization and explain its application to the operator-scaling problem in Section 1.1. We discuss the basics of invariant theory in Section 1.2 and an optimization approach to invariant-theoretic problems in Section 1.2.1. We discuss the left-right group action in detail and explain how the orbit-closure intersection problem for this action (for which we give the first deterministic poly-time algorithm) is a special case of PIT in Section 1.3.

In Section 1.4, we discuss the unitary equivalence problem for the left-right action. Section 2 contains an overview of the techniques we develop.

### 1.1 Geodesically Convex Optimization and Operator Scaling

Convex optimization provides the basis for efficient algorithms in a large number of scientific disciplines. For instance, to find an $\varepsilon$-approximate minimizer, the interior point method runs in time polynomial in the input and *logarithmic* in $1/\varepsilon$. Unfortunately, many problems (especially machine-learning ones) cannot be phrased in terms of convex formulations. A body of general-purpose non-convex algorithms have been recently designed with theoretical guarantees (see [3–6, 15, 62, 65]). However, their guarantees are not as good as in the convex case: they only converge to approximate local minima (some only to stationary points) and run in time *polynomial* in $1/\varepsilon$.

So one might wonder, for what generalizations of convex optimization problems, can one design optimization algorithms with guarantees comparable to convex optimization? One avenue for such a generalization is given by *geodesically convex* problems. Geodesic convexity generalizes Euclidean convexity to Riemannian manifolds [14, 36]. While there have been works on developing algorithms for optimizing geodesically convex functions [1, 63, 72, 76, 78, 79], the theory is still incomplete in terms of what is the best computational complexity.

We focus on geodesically convex optimization over the space of positive definite (PD) matrices endowed with a different geometry than the Euclidean one. This specific geometry on PD matrices is well studied, see [39, 58, 70, 76].

An $n \times n$ (complex) matrix $M$ is positive definite (PD) if it is Hermitian (i.e., $M = M^\dagger$) and all of its eigenvalues are strictly positive. We write $M \succ 0$ to denote that $M$ is PD. The *geodesic path* from any matrix $A \succ 0$ to matrix $B \succ 0$ is a function $\gamma$ that maps $[0, 1]$ to PD matrices, satisfies $\gamma(0) = A$ and $\gamma(1) = B$, and is locally distance minimizing (w.r.t. an appropriate metric). A function $F(M)$ is *geodesically convex* iff the univariate function $F(\gamma(t))$ is convex in $t$ for any PD matrices $A$ and $B$.

In the Euclidean metric, shortest paths are straight lines and such a path is $\gamma(t) = (1-t)A + tB$. In this case, geodesic convexity reduces to classical convexity.

In the standard Riemannian metric over PD matrices, the geodesic path becomes $\gamma(t) = A^{1/2}(A^{-1/2}BA^{-1/2})^t A^{1/2}$. It should be noted that there does not seem to be any global change of variables that would turn geodesically convex functions into Euclidean convex ones; the change of variables is local and varies smoothly over the manifold.

**Operator Scaling.** An example of an optimization problem which is geodesically convex but not convex arises in the problem of *operator scaling* [32, 37]. A tuple of matrices $(A_1, \dots, A_m)$ defines a positive *operator*[1] $T(X) = \sum_i A_i X A_i^\dagger$, mapping PSD matrices to PSD matrices. The so-called *capacity* of operator $T$ is defined by:

$$\mathrm{cap}(T) \stackrel{\text{def}}{=} \inf_{X \succ 0, \det(X)=1} \det(T(X)) \ .$$

The name "operator scaling" comes from the fact that if the infimum $X^*$ is attainable, then by defining $Y^* = T(X^*)^{-1}$ and re-scaling $\widetilde{A}_i = (Y^*)^{1/2} A_i (X^*)^{1/2}$, we have

$$\textstyle\sum_i \widetilde{A}_i \widetilde{A}_i^\dagger = I \quad \text{and} \quad \sum_i \widetilde{A}_i^\dagger \widetilde{A}_i = I \ .$$

This is also known as saying that the new operator

$$\widetilde{T}(X) \stackrel{\text{def}}{=} \textstyle\sum_{i=1}^m \widetilde{A}_i X \widetilde{A}_i^\dagger \text{ is } \textit{doubly stochastic}.$$

Before our work, the only known algorithmic approach to solve the above capacity optimization problem was by Gurvits [37] in 2004. His algorithm is a natural extension of Sinkhorn's algorithm, which was proposed in 1964 [69] for the simpler task of *matrix scaling*. A complete analysis of Gurvits' algorithm was done in Garg et al. [32]. Unfortunately, Gurvits' algorithm (and Sinkhorn's algorithm too) run in time $\mathrm{poly}(n, \log M, 1/\varepsilon)$, where $M$ denotes the largest magnitude of an entry of $A_i$,[2] and $\varepsilon$ is the desired accuracy. The polynomial dependency on $1/\varepsilon$ is poor and slows down the downstream applications (such as orbit-closure intersection).

**Remark 1.1.** *A special case of operator scaling is the* matrix scaling *problem (cf. [7, 18] and references therein). In matrix scaling, we are given a real matrix with non-negative entries, and asked to re-scale its rows and columns to make it doubly stochastic. In this very special case, one* can *make a change of variables in the appropriate capacity, and make it convex in the Euclidean metric. This affords standard convex optimization techniques, and for this special case, algorithms running in time* $\mathrm{poly}(n, \log M, \log 1/\varepsilon)$ *are known [7, 18, 49, 56].*

It is known that for every positive operator $T$, $\log(\det(T(X)))$ is geodesically convex in $X$ [70]. Also, it is simple to verify that $\log(\det(X))$ is geodesically linear (i.e., both convex and concave)[3]. Hence, if we define the following alternative objective (removing the hard constraint on $\det(X)$)

$$\mathrm{logcap}(X) = \log\det(T(X)) - \log\det X \qquad (1.1)$$

then it is geodesically convex over PD matrices $X$. Note that if $\mathrm{cap}(T) > 0$, then $\inf_{X \succ 0} \mathrm{logcap}(X) = \log(\mathrm{cap}(T))$.

Our main result is an algorithm which $\varepsilon$-approximates capacity and runs in time polynomial in $n, m, \log M$ and $\log(1/\varepsilon)$.

**Theorem M1** (informal). *For every $\varepsilon > 0$, there is a deterministic* $\mathrm{poly}(n, m, \log M, \log(1/\varepsilon))$*-time algorithm that finds $X_\varepsilon \succ 0$ satisfying* $\mathrm{logcap}(X_\varepsilon) - \log(\mathrm{cap}(T)) \le \varepsilon$.

Because the problem is non-convex, geodesic convexity plays an important role in getting such an algorithm with a polynomial dependency on $\log(1/\varepsilon)$. Our algorithm is a geodesic generalization of the "box-constrained Newton's method" recently introduced in two independent works [7, 18]. In each iteration, our algorithm expands the objective into its second-order Taylor expansion (up to a *geodesic diameter* 1/2), and then solves it via Euclidean convex optimization.

[1] It is also known as a *completely positive* operator.

[2] One can assume $A_i$'s are integral or complex integral without loss of generality.

[3] This should be contrasted with the fact that $\log(\det(X))$ is a concave function in the Euclidean geometry.

Although we consider a specific application to operator scaling, our algorithm is in fact a general second-order method and applies to any geodesically convex problem (over PD matrices) that satisfies a particular robustness property. This robustness property is much weaker than self-concordance, and was introduced in the Euclidean space by [7, 18]. We believe that our method applies in a similar way to other metrics, and thus may be of much more general applicability.

In contrast, some previous results (e.g. [78, 79]) only analyze *first-order* methods for geodesically convex functions, and thus cannot achieve polynomial dependency on $\log(1/\varepsilon)$ for operator scaling[4]. We hope that more methods from the Euclidean setting would be transported into the geodesic settings and find applications in invariant theory, machine learning, or more broadly in the future.

## 1.2 Invariant Theory, Orbits and Orbit-Closures

We start with a short introduction to the basic concepts of invariant theory, focusing on the various notions of *equivalence* under group actions.

Invariant theory [16] is the study of group actions on vector spaces (more generally algebraic varieties) and the functions (usually polynomials) that are left *invariant* under these actions. It is a rich mathematical field in which computational methods are sought and well developed (see [21, 71]). While significant advances have been made in computational problems involving invariant theory, most algorithms still require exponential time (or longer).

Let $G$ be a group which acts *linearly*[5] on a vector space $V$. (In other words, $V$ is a *representation* of $G$.) Invariant theory is nicest when the underlying field is $\mathbb{C}$ and the group $G$ is either finite, the general linear group $\mathsf{GL}_n(\mathbb{C})$, the special linear group $\mathsf{SL}_n(\mathbb{C})$, or a direct product of these groups. Throughout this paper, whenever we say group, we refer to one of these groups because they are general enough to capture most interesting aspects of the theory.

**Invariant Polynomials.** Invariant polynomials are polynomial functions on $V$ left invariant by the action of $G$. Two simplest examples are

- The symmetric group $G = \mathcal{S}_n$ acts on $V = \mathbb{C}^n$ by permuting the coordinates. In this case, the invariant polynomials are *symmetric* polynomials, and are generated by the $n$ elementary symmetric polynomials.
- Group $G = \mathsf{SL}_n(\mathbb{C}) \times \mathsf{SL}_n(\mathbb{C})$ acts on $V = \mathsf{Mat}_n(\mathbb{C}) = \mathbb{C}^{n\times n}$ by a change of bases of the rows and columns, namely left-right multiplication: that is, $(A, B)$ maps $X$ to $AXB^{\dagger}$. Here, $\det(X)$ is an invariant polynomial and in fact every invariant polynomial must be a univariate polynomial in $\det(X)$.

The above phenomenon that the ring of invariant of polynomials (denoted by $\mathbb{C}[V]^G$) is generated by a finite number of invariant polynomials is not a coincidence. The *finite generation theorem* due to Hilbert [40, 41] states that, for a large class of groups (including the groups mentioned above), the invariant ring must be finitely generated. These two papers of Hilbert are highly influential and laid the foundations of commutative algebra. In particular, "finite basis theorem" and "Nullstellansatz" were proved as "lemmas" on the way towards proving the finite generation theorem!

**Orbits and Orbit-Closures.** The *orbit* of a vector $v \in V$, denoted by $O_v$, is the set of all vectors obtained by the action of $G$ on $v$. The *orbit-closure* of $v$, denoted by $\overline{O}_v$, is the closure (under the Euclidean topology[6]) of the orbit $O_v$. For actions of continuous groups (like $\mathsf{GL}_n(\mathbb{C})$), it is more natural to look at orbit-closures. Call points in the same orbit (or orbit-closure in the continuous setting) *equivalent* under the action of the group. Many fundamental problems in theoretical computer science (and many more across mathematics) can be phrased as questions about such equivalence. Here are some familiar examples:

- Graph isomorphism problem can be phrased as checking if the orbits of two graphs are the same or not, under the action of the symmetric group permuting the vertices.
- Geometric complexity theory (GCT) [60] formulates a variant of $\mathcal{VP}$ vs. $\mathcal{VNP}$ question as checking if the (padded) permanent lies in the orbit-closure of the determinant (of an appropriate size), under the action of the general linear group on polynomials induced by its natural linear action on the variables.
- Border rank (a variant of tensor rank) of a 3-tensor can be formulated as the minimum dimension such that the (padded) tensor lies in the orbit-closure of the unit tensor, under the natural action of $\mathsf{GL}_r(\mathbb{C}) \times \mathsf{GL}_r(\mathbb{C}) \times \mathsf{GL}_r(\mathbb{C})$. In particular, this captures the complexity of matrix multiplication.

**Orbit-closure Intersection.** We study the orbit-closure intersection problem. Given two vectors $v_1, v_2 \in V$, we want to decide whether $\overline{O}_{v_1} \cap \overline{O}_{v_2} \neq \varnothing$. By definition, invariant polynomials are constant on the orbits (and thus on orbit-closures as well). Thus, if $\overline{O}_{v_1} \cap \overline{O}_{v_2} \neq \varnothing$, then $p(v_1) = p(v_2)$ for all invariant polynomials $p \in \mathbb{C}[V]^G$. A remarkable theorem due to Mumford says that the converse is also true (for a large class of groups including the ones we discussed above).

**Theorem 1.2** ([61]). *Fix an action of a group $G$ on a vector space $V$. Given two vectors $v_1, v_2 \in V$, we have $\overline{O}_{v_1} \cap \overline{O}_{v_2} \neq \varnothing$ if and only if $p(v_1) = p(v_2)$ for all $p \in \mathbb{C}[V]^G$.*

The above theorem gives us a way to test if two orbit-closures intersect. However, in most cases, efficient constructions of invariant polynomials (in the sense of succinct descriptions of Mulmuley [59], see also [29]) are not available. In cases where they are available (as we will see is the case for left-right action in Section 1.3), the orbit-closure intersection problem reduces to *polynomial identity testing* that can be solved by randomized poly-time algorithms.

Our optimization approach (see Section 1.2.1) yields *deterministic* poly-time algorithms, and we believe it should work even in settings where efficient constructions of invariants are not available. We describe a general approach next before describing a concrete application for the left-right action in Section 1.3.

[4] Capacity is not strongly geodesically convex.

[5] That is $g \cdot (v_1 + v_2) = g \cdot v_1 + g \cdot v_2$ and $g \cdot (cv) = cg \cdot v$.

[6] It turns out mathematically, it is more natural to look at closure under the Zariski topology. However, for the group actions we study, the Euclidean and Zariski closures match by a theorem due to Mumford [61].

#### 1.2.1 *Optimization approach to invariant-theoretic problems.*

We review an optimization approach to invariant-theoretic problems that comes out of the classical works in geometric invariant theory [51, 61]. We start with the *null-cone membership* problem, which is well defined for any group action. A vector $v \in V$ is said to be in the *null cone* if the orbit-closures of $v$ and 0 intersect. Then the null-cone membership problem is to test if a vector $v$ is in the null cone. This is a special case of the orbit-closure intersection problem.

Given a vector $v \in V$, consider the optimization problem which finds *a vector of minimum* $\ell_2$*-norm* in the orbit-closure of $v$:

$$N(v) = \inf_{g \in G} \|g \cdot v\|_2^2 \tag{1.2}$$

It is easy to see that $v$ is in the null cone iff $N(v) = 0$. For most group actions (think of $G = \mathrm{GL}_n(\mathbb{C})$ for concreteness), the function $f_v(g) = \|g \cdot v\|_2^2$ is not convex in the Euclidean geometry but is geodescially convex (e.g. see [35, 77]). A consequence of geodesic convexity is the so-called Kempf-Ness theorem [51], that states that any critical point (i.e., point with zero gradient) of $f_v(g)$ must be a global minimum. This brings us to *moment maps.*

**Moment map.** Informally, the moment map $\mu_G(v)$ is the gradient of $f_v(g)$ at $g = id$, the identity element of $G$. The Kempf-Ness theorem draws the following beautiful connection between the moment map and $N(v)$. It is a duality theorem which greatly generalizes linear programming duality to a "non-commutative" setting.

**Theorem 1.3** (Kempf and Ness [51])**.** *Fix an action of group $G$ on a vector space $V$ and let $v \in V$.*

- $N(v) > 0 \iff \exists$ *non-zero* $w \in \overline{O}_v$ *s.t.* $\mu_G(w) = 0$.
- *The infimum in* $N(v)$ *is attainable* $\iff O_v$ *is closed* $\iff \exists$ *non-zero* $w \in O_v$ *s.t.* $\mu_G(w) = 0$.
- $N(v) > 0 \implies \exists$ *unique non-zero* $w \in \overline{O}_v$ *s.t.* $\mu_G(w) = 0$. *Uniqueness is upto the action of a maximal compact subgroup $K$ of $G$.*[7]

Theorem 1.3 gives an optimization route to null-cone membership (which was used in [13, 32]): it suffices to find a $w \in \overline{O}_v$ satisfying $\mu_G(w) = 0$.[8] Of course, one cannot hope to compute $w$ exactly as it may not have finite bit-size. Instead, one can hope that 'computing it approximately' will suffice, but how accurate do we need to approximate this vector? We will shortly return to this. First let us discuss if this optimization approach be extended to orbit-closure intersection[9]. The extension is provided by the following theorem due to Mumford [61]:

**Theorem 1.4** ([61])**.** *If* $\overline{O}_{v_1} \cap \overline{O}_{v_2} \neq \emptyset$, *then there is a unique closed orbit in* $\overline{O}_{v_1} \cap \overline{O}_{v_2}$.

[7] Maximal compact subgroups of the groups we care about are simple to describe. For $\mathrm{GL}_n(\mathbb{C})$, a maximal compact subgroup is the unitary group $\mathrm{U}_n(\mathbb{C})$. For $\mathrm{SL}_n(\mathbb{C})$, it is the special unitary group $\mathrm{SU}_n(\mathbb{C})$.

[8] This yields a "scaling problem" of the variety of "matrix scaling" and "operator scaling", and leads naturally to alternating minimization heuristics for special classes of groups.

[9] One could also consider an optimization problem which tries to minimize the distance between the two orbit-closures, something like $\inf_{g,h \in G} \|g \cdot v_1 - h \cdot v_2\|_2^2$. It is not clear if this optimization problem has nice properties like geodesic convexity.

The above theorem essentially follows from Hilbert's Nullstellansatz and the fact that closed orbits are algebraic varieties[10], and hence separated by a polynomial. Theorems 1.3 and 1.4 imply:

**Corollary 1.5.** *Suppose* $N(v_1) > 0$ *and* $N(v_2) > 0$. *If* $\overline{O}_{v_1} \cap \overline{O}_{v_2} \neq \emptyset$, *then there is a unique non-zero* $w \in \overline{O}_{v_1} \cap \overline{O}_{v_2}$ *(upto the action of a maximal compact subgroup $K$) s.t.* $\mu_G(w) = 0$.

**Corollary 1.6.** *In other words, orbit-closure intersection reduces to*

- *computing* $w_1 \in \overline{O}_{v_1}$ *and* $w_2 \in \overline{O}_{v_2}$ *satisfying* $\mu_G(w_1) = \mu_G(w_2) = 0$ *and*
- *testing if* $w_1$ *and* $w_2$ *are in the same orbit of the action of the maximal compact subgroup $K$.*

Again one cannot hope to compute $w_1$ and $w_2$ exactly as they may not have finite bit-sizes. Instead, one can hope that 'computing them approximately' will suffice, but how accurate do we need to approximate these vectors?

For null-cone membership, in some cases [13, 32, 33, 37], it suffices to calculate $\varepsilon$-accurate vectors in $\mathrm{poly}(n, m, 1/\varepsilon)$ time. For the orbit-closure intersection, we need a faster $\mathrm{poly}(n, m, \log 1/\varepsilon)$-time algorithm because the distance between two non-intersecting orbit-closures could be exponentially small in $n, m, \log(M)$ (see Section 2.4). This is what our algorithm for capacity minimization (see Theorem M1) achieves. We remark that the optimization problem $N(v)$ for the left-right group action (described next), after elementary transformations, translates directly to the capacity optimization problem. The role of $v$ is played by the tuple of matrices $(A_1, \ldots, A_m)$ which define the completely positive operator $T$.

## 1.3 Left-Right Group Action and Polynomial Identity Testing

In this section, we introduce the left-right group action, describe its invariants, and explain how to reduce its orbit-closure intersection to a special case of polynomial identity testing. Finally, we use our operator-scaling algorithm to derandomize this special case of polynomial identity testing.

Left-right action is a generalization of the basic action we saw in Section 1.2. The group $G = \mathrm{SL}_n(\mathbb{C}) \times \mathrm{SL}_n(\mathbb{C})$ acts simultaneously on a *tuple* of matrices by left-right multiplication. That is $(C, D)$ sends $(Z_1, \ldots, Z_m)$ to $(CZ_1D^\dagger, \ldots, CZ_mD^\dagger)$. The following theorem characterizes the invariants for left-right action.

**Theorem 1.7** ([2, 24, 25, 67])**.** *The invariants for the left-right action are generated by polynomials of the form* $\det\left(\sum_{i=1}^m E_i \otimes Z_i\right)$, *where $E_i$ are $d \times d$ complex matrices for an arbitrary $d$.*[11]

In remarkable progress recently, Derksen and Makam [22] proved polynomial bounds on the dimension $d$ that one needs to form a generating set (previous bounds were exponential but held for more general group actions [20]). Formally, they proved

[10] For the Zariski closure, this statement follows from the definition but it is true for Euclidean closure as well due to Mumford's theorem that Zariski and Euclidean closures match for the groups we are studying.

[11] Here the matrices $Z_i$ have entries which are disjoint formal variables.

**Theorem 1.8** ([22]). *The invariants for the left-right action are generated by polynomials of the form* $\det\left(\sum_{i=1}^{m} E_i \otimes Z_i\right)$*, where* $E_i$ *are* $d \times d$ *complex matrices for* $d \le n^5$.

Using Theorem 1.2, this reduces the orbit-closure intersection problem for the left-right action to the following special case of polynomial identity testing (PIT).

**Corollary 1.9.** *The orbit-closures of the two tuples* $(A_1, \ldots, A_m)$ *and* $(B_1, \ldots, B_m)$ *intersect under the left-right action iff* $\det\left(\sum_{i=1}^{m} Y_i \otimes A_i\right) \equiv \det\left(\sum_{i=1}^{m} Y_i \otimes B_i\right)$ *for all* $d \le n^5$*. Here, the matrices* $Y_i$ *are* $d \times d$ *with disjoint sets of variables.*

Corollary 1.9 implies a randomized poly-time algorithm for the orbit-closure problem for the left-right action (randomly picking the entries of the $Y_i$ using the Schwartz-Zippel lemma). Using our algorithm for capacity minimization in Section 1.1 and the invariant-theory framework in Section 1.2.1, we show

> **Theorem M2** (informal). *There is a deterministic polynomial-time algorithm for the orbit-closure intersection problem for the left-right action.*

This generalizes the results in [32, 46] where deterministic poly-time algorithms were designed for the null-cone problem. We refer the readers to those papers for applications of the null-cone problem in non-commutative algebra, analysis, and quantum information theory.

Designing a deterministic algorithm for PIT is a major open problem in complexity theory with applications to circuit lower bounds [48]. There has been extensive work on designing deterministic algorithms for identity testing for restricted computational models (e.g. [26, 30, 50, 53, 66]).[12] However, the above results in PIT (corresponding to the null-cone or orbit-closure intersection problems) give rise to very different class of polynomials for which we can *now* solve PIT in deterministic poly-time. This is part of a bigger agenda proposed by Mulmuley [59] to study PIT problems arising in algebraic geometry and invariant theory.

The other novel aspect of the PIT algorithms in [32] and the current paper is that they are based on continuous optimization whereas the original problems are purely algebraic. It is perhaps not surprising that optimization approaches are now coming back to PIT, since many of the fundamental combinatorial optimization problems like bipartite matching, general matching, linear matroid intersection, and linear matroid parity are special cases of PIT [27, 57].

[12] It is perhaps worth pointing out that, the null cone and orbit-closure intersection problems for the *simultaneous conjugation action* can be done in deterministic time using one such computational model — read-once algebraic branching programs [29, 64]. There are also other instances of PIT that can be solved in deterministic poly-time but which do not correspond to any restricted computational models. These include papers in math studying subspaces of singular matrices [8–10, 28, 34] (after all, by Valiant's completeness theorem for determinant [73], PIT is essentially equivalent to testing if a subspace of matrices contains a non-singular matrix), PIT for subspaces of matrices spanned by rank-1 matrices [37, 42, 43] and algorithms for module isomprhism [12, 17].

## 1.4 Side Result: Unitary Equivalence Testing for the Left-Right Action

When deriving our algorithm for Theorem M2, we in fact need a subroutine for checking if two given tuples are equivalent under the left-right action: given two tuples of matrices $A = (A_1, \ldots, A_m)$ and $B = (B_1, \cdots, B_m)$, check if there exist unitary matrices $U, V$ such that $UA_iV = B_i$ for all $i$.

Recall there is a deterministic polynomial-time algorithm for this problem (for instance combining [17, Theorem 4] and [44, Proposition 15]). There has been a lot of work characterizing the conditions under which two tuples are equivalent up to unitary transformations [31, 47, 68, 75].

However, in this paper, we need an algorithm for an approximate version of the unitary equivalence problem (recall the discussion in Section 1.2.1). We develop a deterministic polynomial-time algorithm for this purpose, where the time complexity has only poly-logarithmic dependency on the approximation parameter $\varepsilon$.

> **Theorem M3** (informal). *There is a deterministic* $\mathrm{poly}(n, m, \log M, \log(1/\varepsilon))$*-time algorithm that, given two tuples* $A$ *and* $B$ *and* $\varepsilon > 0$*, outputs*
>
> - *yes if there are unitary matrices* $U, V$ *s.t.* $UAV$ *is* $\varepsilon$*-close to* $B$*; or*
> - *no if for all unitary matrices* $U, V$*,* $UAV$ *is* $\varepsilon'$*-far to* $B$*, where* $\varepsilon' = \varepsilon^{1/\mathrm{poly}(n,m)} M^{\mathrm{poly}(n,m)}$.

We believe this algorithm may be of independent interests with possibly other applications.

## 1.5 Open problems

We design an algorithm for operator scaling with time polynomial in input size and $\log(1/\varepsilon)$, and use it to give a deterministic polynomial-time algorithm for the orbit-closure intersection problem for the left-right action. We believe the recent coming together of optimization and invariant theory (from an algorithmic perspective) is a very exciting development and there are many interesting research directions and open problems in this area. We list some of the most interesting ones (from our perspective).

(1) In terms of optimization, it is interesting to design efficient algorithms for other classes of geodesically convex functions, especially with time polynomial in $\log(1/\varepsilon)$, even when the function is not strongly convex. Of particular interest is the manifold on PD matrices that is described in Section 1.1. This will directly lead to polynomial-time algorithms for testing null-cone membership for more general actions, e.g. for the natural action of $\mathrm{SL}_n(\mathbb{C}) \times \mathrm{SL}_n(\mathbb{C}) \times \mathrm{SL}_n(\mathbb{C})$ on tensors in $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \mathbb{C}^n$ (see [13] for some partial results).
(2) Design black-box PIT algorithms for testing null-cone membership and orbit-closure intersection for the left-right action, even for characteristic 0. Our algorithm is inherently white box.
(3) Design efficient deterministic algorithms for the null cone and orbit-closure intersection problems for actions, of $\mathrm{GL}_n(\mathbb{C})$ for concreteness, only assuming polynomial degree bounds on a

generating set. The tools in this paper might already be enough to tackle this general problem.

## Independent work

Independent and concurrent to this work, Derksen and Makam [23] have found a different algorithm for testing orbit-closure intersection for the left-right action. Their algorithm is conceptually simpler than ours, and does not use optimization techniques. Their algorithm works over fields of positive characteristic as well, and may be viewed as extending the null-cone membership algorithm in [46].

# 2 TECHNIQUES AND PROOF OVERVIEW

- Section 2.1 describes a high level plan for our second-order algorithm for geodesically convex optimization.
- Section 2.2 contains an overview of our optimization algorithm for operator scaling.
- Section 2.3 contains a proof overview of our diameter bound for the optimal solutions to capacity optimization.
- Section 2.4 describes our algorithm for the orbit-closure intersection problem for the left-right action.
- Section 2.5 describes an algorithm for checking approximate unitary equivalence of two tuple of matrices under the left-right action.

## 2.1 Geodesically Convex Optimization

In this section, we provide a high-level overview of our general algorithm for minimizing geodesically convex functions over a natural manifold over PD matrices. The algorithm is a geodesic analogue of the box-constrained Newton's method in [7, 18]. The box-constrained Newton's method is related to trust-region methods (see [19] and the references therein). There has been study of Riemannian/geodesic analogues of these trust-region methods [11]. As far as we know, there was no analysis previously that gave a running time polylogarithmically in the error parameter. While we apply our second-order method to a specific metric, the framework is very general and we believe applicable to many other settings.

We say that a function $F$ over PD matrices is g-convex if for every PD matrix $X$ and every Hermitian matrix $\Delta$, $F\left(X^{1/2}e^{s\Delta}X^{1/2}\right)$ is a convex function in $s$. We also assume a robustness condition on the function $F$ which essentially says that the function behaves like a quadratic function in every "small" neighborhood with respect to the metric.

Our algorithm is quite simple. Starting with $X_0 = I$, we update $X_t$ to $X_{t+1}$ by solving a (constrained) Euclidean convex quadratic minimization problem. Define $f^t(\Delta) = F\left(X_t^{1/2}\exp(\Delta)X_t^{1/2}\right)$. Let $q^t$ be the second-order Taylor expansion of $f^t$ around $\Delta = 0$. We have $q^t$ is a convex and quadratic (in the Euclidean space) because $F$ is $g$-convex. Then, we optimize $q^t(\Delta)$ under the convex constraint $\|\Delta\|_2 \le 1/2$ (i.e., the "box" constraint). Let $\Delta_t$ be the optimal solution, and we update $X_{t+1} = X_t^{1/2}\exp(\Delta_t)X_t^{1/2}$.[13]

We prove this algorithm finds an $\varepsilon$-approximate minimizer of $F(\cdot)$ in $O(R\log(1/\varepsilon))$ iterations. Here, the diameter parameter $R$ is an upper bound on $\log\left(X_t^{-1/2}X^*X_t^{1/2}\right)$, where $X^*$ is some optimal solution for $F$.

Let us give some intuition for the proof of this. We will prove that in each iteration $F(X_t) - F(X^*)$ decreases by a multiplicative factor of roughly $1 - \Omega(1/R)$. Denote by $\Delta^* = \log\left(X_t^{-1/2}X^*X_t^{1/2}\right)$, that is, the "direction" from $X_t$ towards $X^*$. Also let $h(s) = f^t(s\Delta^*)$ and $\Delta'_t = \Delta^*/2R$.

We know $h$ is a univariate convex function due to g-convexity of $F$. Therefore,

$$F(X_t) - F(X^*) = h(0) - h(1) \le 2R\,(h(0) - h(1/2R)) = 2R\left(F(X_t) - f^t\left(\Delta'_t\right)\right) \ .$$

On the other hand, since $\|\Delta'_t\|_2 \le 1/2$, we have that $f^t\left(\Delta'_t\right) \approx g^t\left(\Delta'_t\right)$ by the robustness assumption. Therefore, our obtained solution $\Delta_t$ —which minimizes $q^t(\Delta)$ under the convex constraint $\|\Delta\|_2 \le 1/2$— will be at least no worse than $\Delta'_t$, or in symbols:

$$f^t\left(\Delta'_t\right) \approx g^t\left(\Delta'_t\right) \ge g^t(\Delta_t) \approx f^t(\Delta_t) = F(X_{t+1})$$

Combining the above two inequalities, we have $F(X_t) - F(X_{t+1}) \ge (1 - \Omega(1/R))(F(X_t) - F(X^*))$.

## 2.2 Operator Scaling via Geodesically Convex Optimization

Recall that we are given a positive operator $T(X) = \sum_{i=1}^m A_i X A_i^\dagger$, where matrices $A_i$ are $n\times n$ and whose entries are complex numbers with integer coefficients (Gaussian integers).[14] We want to solve the following optimization problem:

$$\mathrm{cap}(T) = \inf_{X\succ 0 \wedge \det(X)=1} \det(T(X))$$

Before going into our algorithm, let us first explain what is known for a commutative special case of the above optimization problem, which is called *matrix scaling*. There one is given a non-negative $n \times n$ matrix $N$ and one wants to solve the following optimization problem [38]:

$$\mathrm{cap}(N) = \inf_{x>0 \,\wedge\, x_1x_2\cdots x_n=1} \prod_{i=1}^n (Nx)_i$$

The above program is an instance of geometric programming, so one can formalize it as a convex function and apply the ellipsoid algorithm to solve it to accuracy $\varepsilon$ in time $\mathrm{poly}(n, b, \log(1/\varepsilon))$, where $b$ denotes the bit size of entries in $N$ [49]. In contrast, our operator scaling problem is not convex, and there is no analogue of ellipsoid algorithm for geodesically convex optimization.

Linial et al. [56] presented an algorithm for matrix scaling which also gives a polylogarithmic time dependency in $1/\varepsilon$. Unfortunately,

[13]There is a minor difference in the actual algorithm —where we compute $\exp(\Delta_t/e^2)$ instead of $\exp(\Delta_t)$— but we ignore the subtlety here.
[14]If $A_i$ contains rational entries then one can multiply all matrices by the common denominator.

for its natural extensions to operator scaling, we are aware of counter examples (due to matrix non-commutativity) in which their approach fails to generate similar polylogarithmic efficiency.

We apply Section 2.1 to operator scaling. Recall that

$$\mathrm{logcap}(X) = \log\det\left(\sum_i A_i X A_i^\dagger\right) - \log\det X$$

is geodesically convex over PD matrices [54, 70]. Unfortunately, in the language of Section 2.1, the diameter parameter $R$ is not polynomially bounded. In particular, the exact minimizer $X^*$ of $\mathrm{logcap}(X)$ may not even be attainable (so can be at infinity). We fix this issues in two steps.

- First, we show (see Section 2.3) that there is an (approximate) minimizer $X^*_\varepsilon$ of $\mathrm{logcap}(X)$ that has a bounded condition number. That is, $\mathrm{logcap}(X^*_\varepsilon) \le \inf_{X \succ 0} \mathrm{logcap}(X) + \varepsilon$ and $\kappa(X^*_\varepsilon) \stackrel{\text{def}}{=} \lambda_{\max}(X^*_\varepsilon)/\lambda_{\min}(X^*_\varepsilon) \le \exp(\mathrm{poly}(n, \log M, \log(1/\varepsilon))$ is bounded.
- Second, we add a regularizer $\mathrm{reg}(X) = \mathrm{Tr} X \cdot \mathrm{Tr} X^{-1}$ (which is also g-convex) to the objective. This ensures that when minimizing $F(X) = \mathrm{logcap}(X) + \mu\,\mathrm{reg}(X)$ for some sufficiently small parameter $\mu > 0$, we always have $\kappa(X) \le \exp(\mathrm{poly}(n, \log M, \log(1/\varepsilon))$.

Finally, since both $\kappa(X^*_\varepsilon)$ and $\kappa(X)$ are bounded, one can show that the diameter parameter $R = O(\log\kappa(X^*_\varepsilon) + \log\kappa(X))$ is also polynomially bounded. We can now apply Section 2.1 directly.

## 2.3 Bounds on Eigenvalues of Scaling Matrices

We want to bound the condition number of a minimizer $X^*$ of the $\mathrm{logcap}(X)$. Note that the infimum of $\inf_{X \succ 0}\{\mathrm{logcap}(X)\}$ may not be attainable, and in such case we want to bound the condition number of some $X^*_\varepsilon$ that satisfies $\mathrm{logcap}(X^*_\varepsilon) - \inf_{X \succ 0}\{\mathrm{logcap}(X)\} \le \varepsilon$. Let us call such $X^*_\varepsilon$ being $\varepsilon$-minimizers.

We remark that similar bounds for the simpler matrix-scaling case were derived in Kalantari and Khachiyan [49] (for $X^*$) and in Allen-Zhu et al. [7] (for the more general $X^*_\varepsilon$). Unfortunately, these *combinatorial* proofs do not apply to the operator case due to non-commutativity, even when the infimum is attainable.

We take a completely different approach by considering a symmetric formulation of capacity:[15]

$$\widetilde{\mathrm{cap}}(T) = \inf_{X, Y \succ 0, \det(X)=\det(Y)=1} \mathrm{Tr}\,[X\,T(Y)]$$

Optimal solutions for $\widetilde{\mathrm{cap}}(T)$ have direct correspondence to the optimal solutions for $\mathrm{cap}(T)$. The proof considers running gradient flow on the objective $\mathrm{Tr}\,[X\,T(Y)]$. [16] The main trick is to continuously follow the gradient but *normalized* to norm 1. That is

$$\frac{d}{dt}(X_t, Y_t) = \frac{\mathrm{grad}\,\mathrm{Tr}\,[X_t\,T(Y_t)]}{\|\mathrm{grad}\,\mathrm{Tr}\,[X_t\,T(Y_t)]\|_2}$$

where the gradient has to be defined appropriately. Then, we use several known properties of capacity [32, 55] to prove that the gradient flow converges in polynomial time with a linear convergence rate (i.e., error $\varepsilon \propto e^{-O(t)}$ where $t$ is the time). Also since the gradient has norm 1, informally, the log of the condition number of an $\varepsilon$-minimizer shall be bounded by the amount of time that the gradient flow reaches an $\varepsilon$-minimizer. This yields that there exists an $\varepsilon$-minimizer $X^*_\varepsilon$ (one reached by the continuous gradient flow) that has a bounded condition number, that is, $\kappa(X^*_\varepsilon) \stackrel{\text{def}}{=} \lambda_{\max}(X^*_\varepsilon)/\lambda_{\min}(X^*_\varepsilon) \le e^R$ where $R = \mathrm{poly}(n, \log M, \log(1/\varepsilon))$ and $\log M$ is the bit complexity of entries of the matrices $A_i$ defining the operator $T$.

Note that this is only an existential proof and one cannot algorithmically find an $\varepsilon$-minimizer using this gradient flow. Indeed, if one discretizes the gradient flow, the resulting algorithm will be a first-order method that converges in a number of iterations *polynomially* in $1/\varepsilon$ as opposed to poly-logarithmically (the objective is not strongly geodesically convex). This is why we have to design a separate algorithm (as explained in the next section) to find an $\varepsilon$-minimizer. Note that our proof strategy only yields that there exists an $\varepsilon$-minimizer that has "small" condition number. But as we will describe in the previous section, this will suffice through the use of an appropriate regularizer.

## 2.4 Orbit-Closure Intersection for Left-Right Action

In this section, we give an overview of our algorithm for orbit-closure intersection. We are given two tuples $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$, which we assume integral for simplicity. They are associated with completely positive operators $T_A$ and $T_B$:

$$T_A(X) = \sum_{i=1}^m A_i X A_i^\dagger \quad \text{and} \quad T_B(X) = \sum_{i=1}^m B_i X B_i^\dagger \ .$$

We can assume wlog that both the tuples are not in the null cone since testing null-cone membership for the left-right action is already solved in [32, 46]. This means we can assume $\mathrm{cap}(T_A) > 0$ and $\mathrm{cap}(T_B) > 0$ (as a consequence of the Kempf-Ness theorem, alternatively see [32]).

Recall from Corollary 1.6 that to test orbit-closure intersection for $A$ and $B$, it suffices to

- find tuples $C = (C_1, \ldots, C_m)$ and $D = (D_1, \ldots, D_m)$, in the orbit-closures of $A$ and $B$ respectively, that have moment map 0. For the left-right action, $C$ (or similarly $D$) has moment map 0 if there exists a scalar $\alpha$ s.t. $\sum_{i=1}^m C_i C_i^\dagger = \sum_{i=1}^m C_i^\dagger C_i = \alpha I_n$.
- test whether $C$ and $D$ are equivalent up to left-right multiplications of unitary matrices: that is, whether there exist special unitary matrices $U, V \in \mathrm{SU}_n(\mathbb{C})$ s.t. $UC_iV = D_i$ for all $i \in [m]$.

As argued in Section 1.2.1, we cannot hope for calculating $C$ or $D$ exactly since they do not even have finite bit length. However, we can run our operator scaling algorithm (on the capacity optimization problem) to find tuples $A' = (A'_1, \ldots, A'_m)$ and $B' = (B'_1, \ldots, B'_m)$, in the orbits of $A$ and $B$ respectively, that are exponentially close to $C$ and $D$ respectively. We describe how to do this (this is a standard argument and is explained in the full paper). Suppose $X_\varepsilon$ is s.t. $\log\mathrm{cap}(X_\varepsilon) \le \log\mathrm{cap}(T_A) + \varepsilon$. Then one defines $A'_i = c\,T_A(X_\varepsilon)^{-1/2} A_i X_\varepsilon^{1/2}$, and similarly for $B'$. Here $c$ is a normalization constant to ensure that we remain in the $\mathrm{SL}_n(\mathbb{C}) \times \mathrm{SL}_n(\mathbb{C})$ orbits. Now, if the orbit-closures of $A$ and $B$ intersect, not only $C$ and $D$ are related by unitary matrices, we also know $A'_i \approx_{\delta_1} UB'_iV$ up to some exponentially small error $\delta_1 > 0$. Note that due to our

[15] This $\widetilde{\mathrm{cap}}(T)$ is the same as the minimum $\ell_2$-norm optimization, described in Section 1.2.1, for the left-right action.
[16] This is a special case of Kirwan's gradient flow for general group actions [52, 74], and this particular gradient flow and its properties have also been studied in [55].

new operator-scaling algorithm, we can make the running time polylogarithmic in $1/\delta_1$.

We will prove (see below) that if the orbit-closures of $A$ and $B$ do not intersect, then the tuples $UA'V$ and $B'$ must be $\delta_2$ (in $\ell_2$ distance) far apart for every pair of unitary matrices $U, V \in \mathrm{U}_n(\mathbb{C})$ (with $\det(UV) \approx 1$). Here $\delta_2$ is some fixed exponentially small parameter, and we shall choose $\delta_1 \ll \delta_2$. In other words, the orbit-closure intersection problem now reduces to checking if there exist unitary matrices s.t. $UA'V$ is close to $B'$. We provide an efficient algorithm for this problem too, and overview of the techniques will be presented in Section 2.5.

**Distance between non-intersecting orbit-closures.** We now explain, how to prove that $UA'V$ and $B'$ must be $\delta_2$-apart if orbit-closures of $A$ and $B$ do not intersect. By Corollary 1.9, there is an invariant polynomial $p$ of degree at most $n^6$ such that $p(A) \neq p(B)$. We can arrange $p$ to have "small" integer coefficients (using the Schwarz-Zippel lemma). Since $p(A) \neq p(B)$ and $A$ and $B$ have integral entries, $p(A)$ and $p(B)$ are both integer valued and must satisfy $|p(A) - p(B)| \geq 1$. Now, since $UA'V$ and $B'$ lie in the orbits of $A$ and $B$ respectively, we have $p(UA'V) = p(A)$ and $p(B') = p(B)$ (by the definition of invariant polynomials), and hence $|p(UA'V) - p(B')| \geq 1$. Since $p$ has polynomial degree and has small integral coefficients, this implies that $UA'V$ and $B'$ have to far apart by a fixed (exponentially small) value $\delta_2$.

We provide a simple example to show that the orbit-closures can be exponentially close. In this example, $m = 1$, so the tuple has only one matrix. Let $A$ be the $(n \times n)$ diagonal matrix whose entries are all 2. Let $B$ be an arbitrary $(n \times n)$ matrix with entries 1's and 2's s.t. $\det(B) = 2^n + 1$. Since the determinants are different, the orbit-closures of $A$ and $B$ do not intersect. The matrix $2I_n$ lies in the orbit of $A$ and $(2^n + 1)^{1/n} I_n$ lies in the orbit of $B$. The $\ell_2$ distance between these is

$$\sqrt{n}\left((2^n+1)^{1/n} - 2\right) = 2\sqrt{n}\left((1 + 1/2^n)^{1/n} - 2\right) \approx \frac{2}{\sqrt{n}2^n}$$

which is exponentially small in the dimension $n$.

**Comparison of null-cone membership with orbit-closure intersection.** We highlight differences of our result from the work of Garg et al. [32] (which solves a simpler null-cone membership problem). Garg et al. [32] used invariant theory and degree bounds to analyze the convergence of Gurvits' algorithm from [37] [17]. For the simpler null cone problem, it sufficed for them to have an algorithm with inverse polynomial dependence on the approximation parameter.In this paper, we need significant more work (on designing operator-scaling algorithms) to achieve a polylogarithmic time dependency on the error as can be seen from the previous example where non-intersecting orbit-closures can be inverse exponentially close (in terms of the input size).

## 2.5 Algorithm for Checking Unitary Equivalence

In Section 2.4, we have essentially reduced the orbit-closure intersection problem to the following unitary equivalence problem. Given two tuples of matrices $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$, decide:[18]

- if there exist unitary matrices $U, V \in \mathrm{U}_n(\mathbb{C})$ s.t. the tuples $UAV$ and $B$ are $\varepsilon$ close; or
- for all unitary matrices $U, V \in \mathrm{U}_n(\mathbb{C})$, the tuples $UAV$ and $B$ are $\varepsilon'$ far apart.

Here, $\varepsilon \ll \varepsilon'$ and both are exponentially small in the input-size.

What does the left-right action by unitary matrices preserve?

The (real) singular values of individual matrices $A_i$ and $B_i$ are preserved. Therefore, we look for an $i \in [m]$ s.t. the singular values of $A_i$ form at least two distinct clusters. Since singular values in different clusters must be matched differently, we can reduce problem into smaller dimensions each corresponding to one cluster of singular values. However, what if all singular values for $A_i$ are close to each other? This means each $A_i$ must be close to being (a scaling of) a unitary matrix.

Next, let us assume for simplicity that all matrices $A_i$ and $B_i$ are exactly unitary. Since $UA_1V \approx B_1$ if and only if $V \approx A_1^{-1}U^\dagger B_1$, this restricts the search to just $U$ because $V$ can be explicitly written as a function of $U$. Therefore, the new problem we need to solve is the following: does there exist a unitary $U$ s.t. $UA_iA_1^{-1}U^\dagger \approx B_iB_1^{-1}$ for $i \in \{2, \ldots, m\}$.

What does conjugation by a unitary matrix (i.e., left multiplication by $U$ and right by $U^\dagger$) preserve? The eigenvalues! Therefore, similar to the previous step, we can compute the eigenvalues of our new matrices $A_iA_1^{-1}$ and $B_iB_1^{-1}$ for all $i \in \{2, \ldots, m\}$, and look for clusters of eigenvalues to reduce dimensions. If all the eigenvalues are close to each other for every unitary matrices $A_iA_1^{-1}$ and $B_iB_1^{-1}$, then they must both be close to scalings of the identity matrix so all we are left to do is to compare scalars.

Unfortunately, after reducing the dimensions using eigenvalues, we may come back to matrices with different singular values. Therefore, we need to alternatively apply singular-value and eigenvalue decomposition routines, until we are left with identity matrices. It is in fact tricky, but anyways possible, to ensure that the error does not blow up too much in this decomposition process.

## Acknowledgements

We would like to thank Peter Bürgisser, Yash Deshpande, Partha Mukhopadhyay, Suvrit Sra, KV Subrahmanyam and Michael Walter for helpful discussions.

This material is based upon work supported by the National Science Foundation under agreement No. CCF-1412958.

## REFERENCES

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. 2009. *Optimization algorithms on matrix manifolds.* Princeton University Press.
[2] Bharat Adsul, Suresh Nayak, and K. V. Subrahmanyam. 2010. A geometric approach to the Kronecker problem II : rectangular shapes, invariants of n*n matrices, and a generalization of the Artin-Procesi theorem. *Manuscript, available at http://www.cmi.ac.in/ kv/ANS10.pdf* (2010).
[3] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. 2017. Finding Approximate Local Minima for Nonconvex Optimization in Linear Time. In *STOC*. Full version available at http://arxiv.org/abs/1611.01146.

[17] Indeed, the most recent version of their paper does not use any degree bounds

[18] Recall that, testing exact equivalence (i.e., for $\varepsilon = 0$) is a much simpler problem.

[4] Zeyuan Allen-Zhu. 2017. Natasha 2: Faster Non-Convex Optimization Than SGD. *ArXiv e-prints* abs/1708.08694 (Aug. 2017). http://arxiv.org/abs/1708.08694
[5] Zeyuan Allen-Zhu. 2017. Natasha: Faster Non-Convex Stochastic Optimization via Strongly Non-Convex Parameter. In *ICML*. Full version available at http://arxiv.org/abs/1702.00763.
[6] Zeyuan Allen-Zhu and Elad Hazan. 2016. Variance Reduction for Faster Non-Convex Optimization. In *ICML*. Full version available at http://arxiv.org/abs/1603.05643.
[7] Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. 2017. Much Faster Algorithms for Matrix Scaling. In *FOCS*. Full version available at http://arxiv.org/abs/1704.02315.
[8] M. D. Atkinson. 1980. Spaces of matrices with several zero eigenvalues. *Bulletin of the London Mathematical Society* 12, 89-95 (1980).
[9] M. D. Atkinson and S. Lloyd. 1980. Large spaces of matrices of bounded rank. *Quarterly Journal of Math. Oxford* 31 (1980), 253–262.
[10] L. B. Beasley. 1987. Nullspaces of spaces of matrices of bounded rank. *Current trends in matrix theory* (1987).
[11] Nicolas Boumal, P.-A. Absil, and Coralia Cartis. 2016. Global rates of convergence for nonconvex optimization on manifolds. *arXiv preprint arXiv:1605.08101* (2016).
[12] Peter A. Brooksbank and Eugene M. Luks. 2008. Testing isomorphism of modules. *Manuscript, available at http://ix.cs.uoregon.edu/ luks/ModIso.pdf* (2008).
[13] Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. 2018. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. *ITCS* (2018).
[14] H. Busemann. 1955. *The geometry of geodesics*. Academic Press, New York.
[15] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. 2016. Accelerated Methods for Non-Convex Optimization. *ArXiv e-prints* abs/1611.00756 (Nov. 2016).
[16] Arthur Cayley. 1846. On linear transformations. *Cambridge and Dublin Mathematical Journal* 1 (1846), 104–122.
[17] Alexander Chistov, Gábor Ivanyos, and Marek Karpinski. 1997. Polynomial Time Algorithms for Modules Over Finite Dimensional Algebras. *ISSAC* (1997).
[18] Michael B. Cohen, Aleksander Mądry, Dimitris Tsipras, and Adrian Vladu. 2017. Matrix Scaling and Balancing via Box Constrained Newton's Method and Interior Point Methods. In *FOCS*. Full version available at http://arxiv.org/abs/1704.02310.
[19] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. 2000. *Trust Region Methods*. Society for Industrial and Applied Mathematics.
[20] Harm Derksen. 2001. Polynomial bounds for rings of invariants. *Proc. Amer. Math. Soc.* 129, 4 (2001), 955–963.
[21] H. Derksen and G. Kemper. 2002. *Computational Invariant Theory*. Vol. 130. Springer-Verlag, Berlin.
[22] Harm Derksen and Visu Makam. 2017. Polynomial degree bounds for matrix semi-invariants. *Advances in Mathematics* 310 (2017), 44–63. arXiv:1512.03393
[23] Harm Derksen and Visu Makam. 2018. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *arXiv preprint arXiv:1801.02043* (2018).
[24] Harm Derksen and Jerzy Weyman. 2000. Semi-invariants of quivers and saturation for Littlewood-Richardson coefficients. *Journal of the American Mathematical Society* 13, 3 (2000), 467–479.
[25] Mátyás Domokos and A. N. Zubkov. 2001. Semi-invariants of quivers as determinants. *Transformation Groups* 6, 1 (2001), 9–24.
[26] Z. Dvir and A. Shpilka. 2006. Locally Decodable Codes With 2 Queries and Polynomial Identity Testing for Depth 3 Circuits. *SIAM J. Comput* (2006).
[27] Jack Edmonds. 1967. Systems of distinct representatives and linear algebra. *Journal of research of the National Bureau of Standards* 71, 241-245 (1967).
[28] David Eisenbud and Joe Harris. 1988. Vector spaces of matrices of low rank. *Advances in Math* 70 (1988), 135–155.
[29] Michael Forbes and Amir Shpilka. 2013. Explicit Noether normalization for simultaneous conjugation via polynomial identity testing. *RANDOM* (2013).
[30] Michael Forbes and Amir Shpilka. 2013. Quasipolynomial-time Identity Testing of Non-Commutative and Read-Once Oblivious Algebraic Branching Programs. *FOCS* (2013), 243–252.
[31] Vyacheslav Futorny, Roger A. Horn, and Vladimir V. Sergeichuk. 2017. Specht's criterion for systems of linear mappings. *arXiv preprint arXiv:1701.08826* (2017).
[32] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. 2016. Operator scaling: theory and applications. *FOCS* arXiv:1511.03730 (2016).
[33] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. 2017. Algorithmic and optimization aspects of Brascamp-Lieb inequalities, via Operator Scaling. *STOC* (2017).
[34] Boaz Gelbord and Roy Meshulam. 2002. Spaces of Singular Matrices and Matroid Parity. *European Journal of Combinatorics* 23, 4 (2002), 389–397.
[35] Valentina Georgoulas, Joel W Robbin, and Dietmar A Salamon. 2013. The moment-weight inequality and the Hilbert-Mumford criterion. *arXiv preprint arXiv:1311.0410* (2013).
[36] Mikhail Gromov. 1978. Manifolds of negative curvature. *Journal of Differential Geometry* 13, 2 (1978), 223–230.
[37] Leonid Gurvits. 2004. Classical complexity and quantum entanglement. *J. Comput. System Sci.* 69, 3 (2004), 448–484.
[38] Leonid Gurvits and Peter N. Yianilos. 1998. The Deflation-Inflation Method for Certain Semidefinite Programming and Maximum Determinant Completion Problems. *Technical Report, NECI* (1998).
[39] Fumio Hiai and Dénes Petz. 2012. Riemannian metrics on positive definite matrices related to means. *Linear Algebra Appl.* 436, 7 (2012), 2117–2136.
[40] David Hilbert. 1890. Ueber die Theorie der algebraischen Formen. *Math. Ann.* 36, 4 (1890), 473–534.
[41] David Hilbert. 1893. Uber die vollen Invariantensysteme. *Math. Ann.* 42 (1893), 313–370.
[42] Gabor Ivanyos, Marek Karpinski, Youming Qiao, and Miklos Santha. 2015. Generalized Wong sequences and their applications to Edmonds' problems. *J. Comput. System Sci.* 81, 7 (2015), 1373–1386.
[43] Gábor Ivanyos, Marek Karpinski, and Nitin Saxena. 2010. Deterministic Polynomial Time Algorithms for Matrix Completion Problems. *SIAM J. Comput* (2010).
[44] Gábor Ivanyos and Youming Qiao. 2018. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SODA* (2018).
[45] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. 2016. Non-commutative Edmonds' problem and matrix semi-invariants. *Computational Complexity* (2016).
[46] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. 2017. Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics. *ITCS* (December 2017).
[47] N. Jing. 2015. Unitary and orthogonal equivalence of sets of matrices. *Linear Algebra and Its Applications* 481 (2015), 235–242.
[48] Valentine Kabanets and Russell Impagliazzo. 2004. Derandomizing Polynomial Identity Tests means proving circuit lower bounds. *Computational Complexity* 13 (2004), 1–46.
[49] Bahman Kalantari and Leonid Khachiyan. 1996. On the complexity of nonnegative-matrix scaling. *Linear Algebra Appl.* 240 (1996), 87–103.
[50] N. Kayal and N. Saxena. 2007. Polynomial Identity Testing for Depth 3 Circuits. *Computational Complexity* (2007).
[51] George Kempf and Linda Ness. 1979. The length of vectors in representation spaces. *Algebraic geometry (Proc. Summer Meeting, Univ. Copenhagen, Copenhagen, 1978), Lecture Notes in Math.* (1979), 233–243.
[52] Frances Kirwan. 1984. *Cohomology of quotients in symplectic and algebraic geometry*. Mathematical Notes, Vol. 31. Princeton University Press.
[53] A. Klivans and D. Spielman. 2001. Randomness Efficient Identity Testing of Multivariate Polynomials. In *Proceedings of the 33rd Annual STOC*.
[54] Fumio Kubo and Tsuyoshi Ando. 1980. Means of positive linear operators. *Math. Ann.* 246, 3 (1980), 205–224.
[55] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, and Akshay Ramachandran. 2017. The Paulsen Problem, Continuous Operator Scaling, and Smoothed Analysis. *arXiv preprint arXiv:1710.02587* (2017).
[56] Nati Linial, Alex Samorodnitsky, and Avi Wigderson. 1998. A Deterministic Strongly Polynomial Algorithm for Matrix Scaling and Approximate Permanents. *STOC* (1998), 644–652.
[57] Laszlo Lovasz. 1979. On determinants, matchings, and random algorithms. *Fundamentals of Computation Theory* (1979), 565–574.
[58] M. Moakher. 2005. A differential geometric approach to the geometric mean of symmetric positive definite matrices. *SIAM J. Matrix Anal. Appl.* 26 (2005).
[59] Ketan D. Mulmuley. 2017. Geometric complexity theory V: Efficient algorithms for Noether normalization. *J. Amer. Math. Soc.* 30, 1 (2017), 225–309.
[60] Ketan D. Mulmuley and Milind Sohoni. 2001. Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems. *SIAM J. Comput* 31, 2 (2001), 496–526.
[61] David Mumford. 1965. *Geometric invariant theory*. Springer-Verlag, Berlin-New York. vi+145 pages.
[62] Yurii Nesterov and Boris T. Polyak. 2006. Cubic regularization of Newton method and its global performance. *Mathematical Programming* 108, 1 (2006), 177–205.
[63] Tamás Rapcsák. 1997. *Smooth nonlinear optimization in Rn.*
[64] Ran Raz and Amir Shpilka. 2005. Deterministic polynomial identity testing in non commutative models. *Computational Complexity* 14 (2005), 1–19.
[65] Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. 2016. Stochastic Variance Reduction for Nonconvex Optimization. In *ICML*.
[66] S. Saraf and I. Volkovich. 2011. Black-box Identity Testing of Depth 4 Multilinear Circuits. In *Proceedings of the 43rd annual STOC*.
[67] Aidan Schofield and Michel Van den Bergh. 2001. Semi-invariants of quivers for arbitrary dimension vectors. *Indagationes Mathematicae* 12, 1 (2001), 125–138.
[68] H. Shapiro. 1991. A survey of canonical forms and invariants for unitary similarity. *Linear Algebra and Its Applications* 147 (1991), 101–167.
[69] R. Sinkhorn. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics* 35 (1964), 876–879.
[70] Suvrit Sra and Reshad Hosseini. 2013. Conic geometric optimisation on the manifold of positive definite matrices. *arXiv preprint arXiv:1312.1039* (2013).
[71] Bernd Sturmfels. 2008. *Algorithms in Invariant Theory* (2nd ed.). Springer.
[72] Constantin Udriste. 1994. *Convex functions and optimization methods on Riemannian manifolds*. Springer Science & Business Media.

[73] Leslie Valiant. 1979. The complexity of computing the permanent. *Theoretical Computer Science* 8 (1979), 189–201.

[74] Michael Walter. 2014. *Multipartite Quantum States and their Marginals*. Ph.D. Dissertation. ETH Zurich.

[75] N.A. Wiegmann. 1961. Necessary and sufficient conditions for unitary similarity. *Journal of Australian Math. Society* 2 (1961), 122–126.

[76] Ami Wiesel. 2012. Geodesic Convexity and Covariance Estimation. *IEEE transactions on signal processing* (2012).

[77] Chris Woodward. 2011. MOMENT MAPS AND GEOMETRIC INVARIANT THEORY. *arXiv preprint arXiv:0912.1132* (2011).

[78] Hongyi Zhang, Sashank J. Reddi, and Suvrit Sra. 2016. Riemannian SVRG: Fast Stochastic Optimization on Riemannian Manifolds. *NIPS* (2016).

[79] Hongyi Zhang and Suvrit Sra. 2016. First-order Methods for Geodesically Convex Optimization. *COLT* (2016).