LP-based Construction of DC Decompositions for Efficient Inference of Markov Random Fields

Chaitanya Murti
Indian Institute of Science

Dhruva Kashyap
Indian Institute of Science

Chiranjib Bhattacharyya
Indian Institute of Science

Abstract

The success of the convex-concave procedure (CCCP), a widely used technique for non-convex optimization, crucially depends on finding a decomposition of the objective function as a difference of convex functions (dcds). Despite the widespread applicability of CCCP, finding such dcds has attracted little attention in machine learning. For graphical models with polynomial potentials, existing methods for finding dcds require solving a Sum-of-Squares (SOS) program, which is often prohibitively expensive. In this work, we leverage tools from algebraic geometry certifying the positivity of polynomials, to derive LP-based constructions of dcds of polynomials which are particularly suited for graphical model inference. Our experiments demonstrate that using our LP-based technique constructs dcds for polynomial potentials of Markov random fields significantly faster compared to SOS-based approaches used in previous works.

1 INTRODUCITON

Markov Random Fields (MRFs) provide a powerful framework for modeling complex probabilistic systems. These models find applications in numerous domains, particularly in image analysis and computer vision (Wang et al., 2014; Ecker and Jepson, 2010; Salzmann, 2013), where they are often modeled as *Continuous Markov Random Fields* (CMRFs), wherein the probability distribution being modeled is over a continuous set.

Difference of convex (DC) programming offers a computationally efficient method for solving CMRF inference problems (Wang et al., 2014). DC programming relies on decomposing a given cost function into the difference of two convex functions called a DC decomposition (dcd), which can then be used to construct a convex upper bound on the cost (Le Thi et al., 2014; Salakhutdinov et al., 2012). Of particular interest is the convex-concave procedure (Yuille and Rangarajan, 2003), an algorithm for solving differentiable DC problems that has been widely adopted for a variety of problems (Shen et al., 2016; Lipp and Boyd, 2016), and has well-established convergence guarantees (Salakhutdinov et al., 2012; Lanckriet and Sriperumbudur, 2009a; Khamaru and Wainwright, 2018). In Wang et al. (2014), it was shown that difference of convex (DC) programming was a highly effective tool for solving the problem of efficient inference in Continuous Markov Random Fields with polynomial potentials. Specifically, it was shown that the convex-concave procedure proposed in Yuille and Rangarajan (2003) outperformed a variety of other methods, such as Salzmann (2013). Other than graphical model inference, DC programming has been widely used in a variety of applications, including reinforcement learning (Piot et al., 2016, 2014), machine learning (Le Thi and Pham Dinh, 2018), and others (Le Thi et al., 2014; Shen et al., 2016; Lipp and Boyd, 2016).

However, a crucial drawback of DC programming is the requirement of finding a *DC decomposition (dcd)* of a given cost function. Algorithmic methods for constructing dcds are generally restricted to polynomials (Niu, 2018; Wang et al., 2014; Ahmadi and Hall, 2017); however, as shown in Ahmadi and Hall (2017), this problem is, in general, NP-hard.

In this work, we address the problem of efficient inference of CMRFs with polynomial potential functions that are constrained to convex polytopes. Such problems arise naturally in domains such as image processing and computer vision, particularly in applications such as shape-from-shading, for which box constraints can naturally be applied. Motivated by prior success at using DC programming (and the CCCP in partic-

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

ular) for this task, we investigate methods for constructing *local* dcds for polynomials over convex polytopes, particularly to reduce expensive computations required to construct dcds. To that end, we leverage Handelman's theorem (Handelman et al., 1988; Lê and Thá-Hòa-Bình, 2017), a powerful result in algebraic geometry certifying the positivity of polynomials over convex polytopes, in order to derive methods for the construction of such dcds. We state our contributions formally below.

1. We formulate problems of finding dcds over convex polytopes. In Lemma 3, we establish that any polynomial can be expressed as the difference of two locally diagonally dominant polynomials on a given convex polytope. Moreover, Theorem 5 demonstrates the existence of undominated local DC decompositions as a solution to an optimization problem.

2. The optimization problem proposed in Theorem 5 is known to be NP-hard. To address this problem, we employ tools from algebraic geometry to formulate tractable optimization problems whose solutions are these dcds. Our strategy is to enforce the positive-definiteness of the Hessian of the dcd, thereby ensuring convexity. We establish results, leveraging powerful tools from algebraic geometry, to show that verifying positive-definiteness of a matrix polynomial is equivalent to checking the feasibility an SDP. Using this result, we propose Theorem 6, called SDP-Local, which provides an SDP-based relaxation for constructing undominated dc decompositions.

3. However, SDP-based methods such as Wang et al. (2014) and SDP-Local are computationally expensive, and CMRFs are typically high-dimensional - for instance, SFS problems have 16384 variables. To develop methods for constructing polynomial dcds without relying on solving SDPs, we leverage our previous result guaranteeing the existence of a dcd with a diagonally-dominant Hessian. To that end, we propose new algebraic results to certify positive-definiteness of diagonally dominant matrix polynomials in Theorem 8. Using this result, we derive an LP-based relaxation for constructing dcds called DD-Local that can effectively be used for graphical model inference. Furthermore, in Theorem 9, we propose a technique for constructing dcds for a polynomial over a polytope called DD-Linear, that does not require solving an LP or an SDP, thus making it apt for graphical model inference.

4. To highlight the tradeoffs between the quality of the dcd obtained via the different methods proposed, and the cost of computing such decompositions, we compare the cost of computing degree-d' polynomial dcds for an n-dimensional, degree-d polynomial over a convex polytope with $2n$ facets, in Theorem 10. We show that the cost of computing SDP-Local is $\tilde{O}(n^{4d'})$, whereas the cost of computing a dcd with DD-Local is $\tilde{O}(n^{2d'+1})$ and that computing a dcd with DD-Linear can be achieved in $O(n^{d'-1})$.

5. We empirically evaluate our dcd constructions in two ways. First, on a synthetic polynomial, we illustrate the tradeoff between ease of construction of the dcd, and effectiveness with the CCCP on a battery of random polynomials, and show that DD-Local and SDP-Local cause the CCCP to converge in up to 2x fewer iterations. We then evaluate our dcd construction algorithms on standard shape-from-shading tasks, namely the Penny and Mozart datasets. We show that SDP-Local is impractical for such problems, whereas DD-Local and DD-Linear are not. We then show that using dcds produced by solving DD-Local causes the CCCP to converge in 2.28x fewer iterations than DD-Linear.

2 NOTATION

We introduce the notation used throughout this paper. Let $[n]$ denote the set $\{1, \cdots, n\}$. Let $\mathbb{S}^n$ denote the space of $n \times n$ symmetric matrices. For a vector $B \in \mathbb{R}^n$, $B \geq 0$ indicates that B is elementwise nonnegative. Let $\mathbb{R}[x]$ denote the ring of real-valued polynomials over the variable(s) x, and let $\mathbb{S}^m(\mathbb{R}[x])$ denote the set of polynomials with coefficients in $\mathbb{S}^m$. For $f : \mathbb{R}^n \to \mathbb{R}$, $f \in C^k$ indicates the function is k-times differentiable. Let $H_f(x)$ be the Hessian of f, and for $f(x) = f_1(x) + f_2(x)$, let $H_{f_1+f_2}(x) = H_{f_1}(x) + H_{f_2}(x)$. We define monomials using the multi-index notation as follows: for $x \in \mathbb{R}^n$, a monomial of degree $d = |\alpha|$, where $\alpha \in \mathbb{N}^n_{>0}$, is given by $x^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$. Let $Z(x, d)$ be the vector of all monomials in x with degree upto d. $Z(x, d)$ has $\binom{n+d}{d}$ elements, such that $Z(x, d)_\alpha = x^\alpha$ in the multi-index notation. Suppose $f \in \mathbb{R}[x]$ has degree d. We write $f(x) = C_f^\top Z(x, d)$, where C_f is the unique vector of coefficients, and $C_{f,\alpha}$ denotes the coefficient of x^α. We let $(Z(x, d))_1 = 1$, and let e_1 be the coordinate vector, where $(e_1)_1 = 1$. If $H_f(x)$ is the Hessian of f, and $H_{f,ij}$ is the i,jth element of H_f, we have $H_{f,ij}(x) = C_f^\top A_{d,ij}^\top Z(x, d-2)$, where $A_{d,ij}$ is a known constant matrix. Next, suppose $f \in \mathbb{S}^m(\mathbb{R}[x])$, $x \in \mathbb{R}^n$ has degree d. Then, we write $f(x) = \sum_{|\alpha| \leq d} Q_{f,\alpha} x^\alpha$, where $Q_{f,\alpha}$ are unique symmetric matrix coefficients.

For standard definitions and properties of convex functions and sets, refer to Boyd and Vandenberghe (2004). Of particular interest in this work is the fact the Hessians of convex functions are positive semidefinite.

3 BACKGROUND

In this section, we review continuous markov random fields, difference of convex programming, and Handelman's theorem, a powerful tool in algebraic geometry certifying positivity of polynomials over convex polytopes. A more detailed survey of relevant literature is provided in Appendix A.

3.1 Continuous Markov Random Fields with Polynomial Potential Functions

In this section, we discuss continuous Markov random fields (CMRFs) with polynomial potential functions. CMRFs with polynomial potentials provide a natural framework to address problems in computer vision, namely shape-from-shading (SFS). Continuous Markov Random Fields represent distributions over continuous spaces $\mathcal{X} \subset \mathbb{R}^n$. Each $x \in \mathcal{X}$ is an output configuration, and is a set of random variables $x = (x_1, \cdots, x_n)$. The probability of a particular configuration $x \in \mathcal{X}$ is given by $\mathbb{P}[x] \propto e^{-f(x)}$, and we refer to $f(x)$ as a *potential function.* Furthermore, we write

$$f(x) = \sum_{r \in \mathcal{R}} f_r(x_r)$$

where $x_r \in \mathcal{X}_r \subset \mathcal{X}$ is a finite subset of the random variables $x = (x_1, \cdots, x_n)$. In Maximum a Posteriori (MAP) inference, we aim to find the most likely configuration by minimizing $f(x)$, given knowledge of all $f_r(\cdot)$. As in Wang et al. (2014), we formulate this as the following optimization problem.

$$x^* = \arg\min_{x \in \mathcal{X}} \sum_{r \in \mathcal{R}} f_r(x_r) \tag{1}$$

3.1.1 Polynomial Shape-From-Shading

In this section, we discuss the polynomial shape-from-shading (SFS) problem as solving a graphical model inference problem, as shown in Ecker and Jepson (2010); Wang et al. (2014). We are given a 2D image of a 3D object and assume that both the image and object are supported on the same grid. We further assume that the image intensity at each point is proportional to the angle between surface normals of that point, and the light source direction.

In Ecker and Jepson (2010); Salzmann (2013); Wang et al. (2014); Khamaru and Wainwright (2018), this problem is formulated as finding the minimizer of the sum of 3-variable polynomials of degree 4. As noted in Khamaru and Wainwright (2018), these polynomials are non-convex, but bounded from below and coercive. In this work, we apply box constraints to the SFS problem - that is, the depth cannot be a negative number, and the depth must also be bounded from above. For a more detailed discussion on the SFS problem, we refer readers to Appendix E.

3.2 Difference of Convex Programming and the Convex-Concave Procedure

In this section, we discuss difference of convex programming and the convex-concave procedure. First, we define Difference of Convex (DC) functions, as first described in Hartman et al. (1959).

Definition 1. *A real-valued function $f(x)$ is said to be a difference of convex function if there exist real-valued convex functions $g(x)$ and $h(x)$ such that $f(x) = g(x) - h(x)$. We say a function h is a difference of convex decomposition (dcd) of f if both h and $f + h$ are convex.*

We define *local* dcds as follows.

Definition 2. *We say a function h is a local difference of convex decomposition (dcd) of f if both h and $f + h$ are convex on a given convex polytope Γ*

We now state the following properties of difference of convex decompositions of twice differentiable functions, and polynomials in particular, stated first in Wang et al. (2014).

Theorem 1. *Consider a twice-differentiable function f. There exist convex functions g and h such that $f(x) = g(x) - h(x)$. Moreover, the set of g, h that satisfy $f(x) = g(x) - h(x)$ is convex.*

This theorem is a restatement of the claim made in (Hartman et al., 1959), Section II part (ii), and is similarly attributed in works such as Ahmadi and Hall (2017). Since convexity is a local property, Theorem 1 holds over compact sets as well.

Difference of convex programs where $f(x)$ and $h(x)$ are differentiable functions can be solved efficiently using the Convex-Concave Procedure (Yuille and Rangarajan, 2003). Suppose we aim to minimize a DC function $f(x) = g(x) - h(x)$ with dcd $h(x)$ over a convex set $\mathcal{X}$. The CCCP solves the following problem at each iteration.

$$x_{k+1} = \arg\min_{x \in \mathcal{X}} \; g(x) - \nabla h(x_k)^\top (x - x_k) \tag{2}$$

Thus, at each step, the CCCP minimizes a convex over-approximation of the cost function f. Next, when $\mathcal{X}$ can be easily projected upon (i.e. $\mathcal{X}$ is a hypercube), then we can use the projected gradient descent algorithm to solve (2). We state the CCCP formally in algorithm 1. The Convex-concave procedure has been

Algorithm 1: Convex Concave Procedure

Initialize: Cost function f with DC decomposition (g, h), convex constraints $c(x) \leq 0$.
Inputs: Tolerance ϵ, initial point x_0.
while $||\nabla(g(x) - h(x))|| \geq \epsilon$ **do**

$$x_{k+1} = \arg\min_{x \in \Gamma} \quad g(x) - \nabla h(x_k)^\top (x - x_k)$$

Outputs:
$x_* \in \Gamma$

adapted to solving constrained DC programs as well (Shen et al., 2016; Lipp and Boyd, 2016; Khamaru and Wainwright, 2018). Moreover, recent works such as Khamaru and Wainwright (2018); Abbaszadehpeivasti et al. (2023) have stated explicit convergence rates of the CCCP and the DCA respectively, in terms of the Lipschitz constants of g and h, where $f = g - h$. We state the result proposed in Khamaru and Wainwright (2018) below.

Theorem 2 (Khamaru 2018)**.** *Suppose $f(x) = g(x) - h(x)$, where $g, h \in C^2$, and where $g(x)$ is M_g-smooth. Then, the sequence of iterates generated by* (2)*, starting at x_0, satisfies*

$$\text{Avg}_k \|\nabla f(x_k)\| \leq \frac{2M_g(f(x_0) - f(x^*))}{k+1},$$

where x^* is a critical point of f in $\mathcal{X}$.

Crucial to DC programming is the construction of DC decompositions. From Theorem 2, we see that we require dcds where M_g is minimized. Note that this analysis only assumes that the dcd is Lipschitz smooth, and does not make use of second order derivatives at all. Inspired by Ahmadi and Hall (2017), we seek to minimize the trace of the Hessian of $g(x)$ to find *undominated* dcds, defined and characterized in Section 4, by finding a dcd h that minimizes $\max_{x \in \Gamma} \ \text{Tr}(H_h(x))$; this also minimizes the pointwise maximum of the trace of the Hessian of g, since $g = f + h$. Since the Lipschitz constant of twice-differentiable convex functions is given by the largest eigenvalue, we use the trace of the Hessian as a proxy for the Lipschitz constant M_g. However, characterizing the convergence rate of the CCCP when the objective function and dcds are polynomials remains an open problem, as noted in Abbaszadehpeivasti et al. (2023).

Methods for constructing DC decomposition for quadratic polynomials were first proposed in Bomze and Locatelli (2004), and extended to general polynomials, in Wang et al. (2014); Ahmadi and Hall (2017). However, these results have not been extended to the case where local dcds are required. This motivates us to use Handelman's theorem, a powerful tool from algebraic geometry, to construct *local* DC decompositions of polynomials over convex polytopes.

3.3 Positive Polynomials over Polytopes

Our strategy for constructing dcds over convex polytopes requires enforcing the positive-semidefiniteness of the Hessian, and minimizing it's trace. Polynomial optimization offers a wide array of tools to address this problem. In this section, we review prior work in algebraic geometry, and state Handelman's theorem, a powerful result in algebraic geometry that provides a means to certify strict polynomial positivity over closed convex polytopes. In the sequel, we use Handelman's theorem to guide principled approaches for constructing local DC decompositions over closed convex polytopes.

First, we define a convex polytope $\Gamma \subset \mathbb{R}^n$ as the closed, bounded intersection of K half-spaces (referred to as facets) as the set

$$\Gamma : \{x : \lambda_i(x) = a_i^\top x + b_u \geq 0, \ i = 1, \cdots K\}. \quad (3)$$

We denote $\Lambda = (\lambda_1, \cdots, \lambda_K)$ and $\Lambda(x) = (\lambda_1(x), \cdots, \lambda_K(x))$ throughout this work. We state Handelman's theorem as follows.

Theorem 3 (Handelman (Handelman et al., 1988))**.** *Consider a convex polytope Γ as defined in* (3)*, and $f \in \mathbb{R}[x]$ of degree d. $f(x) > 0$ for all $x \in \Gamma$ if and only if, for some $d' \geq d$, there exists a polynomial $q_f \in \mathbb{R}[\Lambda]$ where*

$$q_f(\Lambda) = \sum_{|\alpha| \leq d'} b_\alpha \Lambda^\alpha$$

with nonnegative coefficients, and such that $q_f(\Lambda(x)) = f(x)$.

This theorem was originally stated in Handelman et al. (1988), and a proof with a degree bound was presented in Powers and Reznick (2001).

We now state an extension of Theorem 3 to symmetric polynomial matrices.

Theorem 4 (Le and Binh (Lê and Thá-Hòa-Bình, 2017))**.** *Consider a convex polytope Γ as defined in* (3)*, and suppose $F \in \mathbb{S}^m(\mathbb{R}[x])$ is of degree d. $F(x) \succ 0$ for all $x \in \Gamma$, if and only if, for some $d' \geq d$, there exists a polynomial $Q_f \in \mathbb{S}^m(\mathbb{R}[\Lambda])$ where*

$$Q_f(\Lambda) = \sum_{|\alpha| \leq d'} B_\alpha \Lambda^\alpha$$

with where $B_\alpha \succeq 0$, and such that $Q_f(\Lambda(x)) = F(x)$.

In subsequent sections, we will use this theorem to enforce the positive-definiteness of the Hessian of the dcd.

4 CONSTRUCTION OF LOCAL UNDOMINATED DC DECOMPOSITIONS

In this section, we define undominated dcds of polynomials defined over compact convex sets, and provide Theorem 5, inspired by Ahmadi and Hall (2017), that guarantees the existence of an undominated dcd, and show that it is a solution to an optimization problem. We propose Theorem 6, which provides an SDP-based relaxation for constructing undominated dcds. All proofs are relegated to Appendix B.

4.1 The Existence of Undominated DC Decompositions

We now define undominated DC decompositions. As shown in Ahmadi and Hall (2017); Bomze and Locatelli (2004), undominated DC decompositions yield significant improvements in the convergence rate of the convex-concave procedure, and are thus of significant interest.

Definition 3. *Suppose h is a dcd of $f \in C^2$, where $f : \mathbb{R}^n \to \mathbb{R}$. If, for any dcd h' of f where $h' \neq h$, we have $h - h'$ is nonconvex, then we say h is an undominated dcd of f.*

We now provide a characterization of *local* undominated dc decompositions defined over compact convex sets, such as polytopes. We show that we are guaranteed to find undominated dcds for polynomials defined over polytopes, and that they are the solution(s) to a convex optimization problem.

Theorem 5. *Let Γ be a polytope as defined in (3), and suppose $f \in \mathbb{R}[x]$ has degree at least 3. Consider the optimization problem*

$$\min_{h \in \mathbb{R}[x]} \max_{x \in \Gamma} \ \mathrm{Tr}(H_h(x)) \quad \text{s.t.} \ H_h(x),\ H_{f+h}(x) \succeq 0 \ \forall\, x \in \Gamma \tag{4}$$

where the degree of h is at least 3. Then, a solution h^* to (4) always exists, and is an undominated dcd of f.

Thus, we can find undominated DC decompositions as follows. Given $f \in \mathbb{R}[x]$ of degree d and a polytope Γ as defined in (3), we find $h \in \mathbb{R}[x]$ of degree $d' \geq d$ that solves

$$\begin{aligned} &\min \ \rho \\ &\text{s.t.} \ \rho - \mathrm{Tr}(H_h(x)) \geq 0, \\ &\qquad H_h(x) \succeq 0,\ H_{f+h}(x) \succeq 0 \ \forall\, x \in \Gamma. \end{aligned} \tag{5}$$

As noted in Ahmadi and Hall (2017), constructing DC decompositions is, in general, NP-hard. However, for any polynomial $f \in \mathbb{R}[x]$ of degree d, we can optimize over a subset of valid dcds of $f(x)$. We provide such constructions in the sequel.

4.2 Checking Polynomial Positivity over Convex Polytopes

In order to develop tractable algorithms for constructing local DC decompositions, it is necessary to develop tractable tests of the positive-semidefiniteness of matrix polynomials. First, we establish connections between Handelman's theorem and convex optimization. In this section, we elucidate the connection between the positivity certificates for polynomials supported on convex polytopes provided in Handelman et al. (1988); Lê and Thá-Hòa-Bình (2017), and the feasibility of convex problems. Much like checking non-negativity of polynomials using SOS relaxations is equivalent to checking the feasibility of an SDP (Parrilo, 2000, 2003), we show that checking the positivity of a polynomial over a convex polytope using Handelman's theorem is equivalent to checking the feasibility of a sufficiently large LP.

We now introduce a lemma relating the coefficients of a polynomial and its Handelman representations defined over a convex polytope.

Lemma 1. *Let Γ be a polytope defined as in (3), and suppose $f \in \mathbb{R}[x]$ of degree d is written as $C^\top Z(x, d)$. Then, there exists a polynomial $q \in \mathbb{R}[\Lambda]$, where $q_f(\Lambda) = B^\top Z(\Lambda, d')$ for some $d' > d$, such that $q_f(\Lambda(x)) = q_f(a_1^\top x + b_1, \cdots, a_K^\top x + b_K) = f(x)$, and a real-valued matrix $L(\Gamma, d') \in \mathbb{R}^{\binom{K+d'}{d'} \times \binom{n+d'}{d'}}$ such that*

$$C = L(\Gamma, d')^\top B. \tag{6}$$

Remark: $q(\Lambda)$ is referred to as a *Handelman representation* (Sankaranarayanan et al., 2013). Furthermore, the above lemma conveys the fact that given a polynomial $f(x)$ of degree d with coefficients C, and for any polytope Γ, a Handelman representation $q_f(\Lambda)$ exists and can be found by solving a linear equation (though the solution is not necessarily unique). The matrix $L(\Gamma, d)$ and the vectors $l_\alpha(\Gamma)$ are fixed and unique, and constant for each polytope Γ and each degree $d > 0$; we will use this notation everywhere hereafter.

Corollary 1. *Suppose $f \in \mathbb{R}[x]$ of degree d, where $f = C^\top Z(x, d)$ is positive on a polytope Γ defined as in (3). Then, $f(x) > 0 \ \forall\, x \in \Gamma$ if and only if, for some $d' \geq d$, there exists a vector $B \in \mathbb{R}^{\binom{K+d'}{d'}}$ such that*

$$C = L(\Gamma, d')^\top B, \ \ B \geq 0, \tag{7}$$

where $L(\Gamma, d')$ is a known matrix as given in (6).

Proof Sketch. We apply Theorem 3 and Lemma 1, and collect terms. ☐

4.2.1 Positive-semidefineteness of Matrix Polynomials over Polytopes

In this section, we extend the result of Corollary 1 to matrix polynomials by showing that checking the positive-semidefiniteness of a matrix polynomial is equivalent to checking the feasibility of a linear matrix inequality (LMI). This result enables us to use Handelman's theorem to construct undominated dcds of polynomials using semidefinite programming.

First, we establish the relationship between the coefficients of a polynomial and it's Handelman representation.

Lemma 2. *Let* Γ *be a polytope as defined in* (3)*, and suppose* $F \in \mathbb{S}^m(\mathbb{R}[x])$ *is of degree* d*, where* $f(x) = \sum_{|\alpha|\leq d'} C_\alpha x^\alpha$*, where* $d' \geq d$*. Then, there exists a matrix-valued polynomial* $Q_F \in \mathbb{S}^m(\mathbb{R}[\Lambda])$ *of degree* d'*, where* $Q_F(\Lambda(x)) = \sum_{|\alpha|\leq d'} B_\alpha x^\alpha$*, and a vector of real coefficients* $l_\alpha(\Gamma)$*, such that* $Q_F(\Lambda(x)) = F(x)$ *and*

$$C_\alpha = \sum_{|\alpha'|\leq d} (l_\alpha(\Gamma, d))_{\alpha'} B_{\alpha'} \text{ for all } \alpha. \quad (8)$$

where $l_\alpha(\Gamma, d)$ *is the row of* $L(\Gamma, d)$ *corresponding to the exponent* $\alpha \in \mathbb{N}^n$*.*

Using this result, we show that checking the positive-semidefiniteness of a matrix polynomial over a polytope is equivalent to checking the feasibility of a set of LMIs.

Corollary 2. *Suppose* $F \in \mathbb{S}^m(\mathbb{R}[x])$ *where* $F(x) = \sum_{|\alpha|\leq d} C_\alpha x^\alpha$*, where* $F(x)$ *is positive definite on a polytope* Γ *defined as in* (3)*. Then, there exists a matrix-valued polynomial* $Q_F \in \mathbb{S}^m(\mathbb{R}[\Lambda])$ *of degree* d'*, where* $Q_F(\Lambda(x)) = \sum_{|\alpha|\leq d'} B_\alpha x^\alpha$*, and a vector of real coefficients* $l_\alpha(\Gamma)$*, such that* $Q_F(\Lambda(x)) = F(x)$*, such that*

$$C_\alpha = \sum_{|\alpha|'\leq d'} (l_\alpha(\Gamma, d))_{\alpha'} B_{\alpha'}, \;\; B_{\alpha'} \succeq 0 \text{ for all } \alpha', \quad (9)$$

where $l_\alpha(\Gamma, d)$ *is the row of* $L(\Gamma, d)$ *corresponding to the exponent* $\alpha \in \mathbb{N}^n$*.*

Proof Sketch. We apply Theorem 4 and Lemma 2 to F, and collect terms appropriately. ☐

This condition can also be checked using linear programming by assuming that the coefficients must be diagonally dominant. However, in general, the existence of a Handelman representation with diagonally dominant coefficients is not guaranteed. We discuss this further in Appendix C.

4.3 SDP-based construction of Undominated DC Decompositions

We begin with solving (5). While this problem is NP-hard, we can optimize over subsets of a fixed degree using Corollary 2.

First, let $h(x) = C_h^\top Z(x)$, and recall that $C_{h,\alpha}$ is the element of C_h corresponding to the monomial x^α in the multi-index notation. Then, the Hessian of $h(x)$ can be written as

$$H_h(x) = \sum_{|\alpha|\leq d'-2} D_\alpha x^\alpha, \; D_\alpha = \sum_{2\leq|\alpha'|\leq d'} C_{h,\alpha'} M_{\alpha,\alpha'} \;\forall\; \alpha, \quad (10)$$

where each $M_{\alpha,\alpha'}$ is a known constant matrix. With this, we derive an SDP relaxation for finding undominated dcds.

Theorem 6 (SDP-Local)**.** *Suppose* $f \in \mathbb{R}[x]$*,* $f(x) = C_f^\top Z(x, d)$ *is of degree* d*, and suppose* Γ *is a polytope as defined in* (3)*. Then, for some* $d' \geq d$*, the SDP*

$$\begin{aligned} \min_{\rho, C_h, B} \;\; & \rho \\ \text{s.t. } & (9) \text{ is satisfied for } H_h(x),\, H_{f+h}(x) \\ & \rho e_i + \sum_{2\leq|\alpha'|\leq d'} C_{h,\alpha'} \mathrm{Tr}(M_{\alpha,\alpha'}) = L(\Gamma, d)B \end{aligned} \quad \text{(SDP)}$$

is feasible with $B \geq 0$ *being an elementwise nonnegative vector, and* $h(x) = C_h^\top Z(x, d')$ *is a dcd of* f*.*

Proof Sketch. We directly apply corollary 2 to (5), and collect terms, following which, we get the result. We provide the full proof in Appendix B. ☐

For the sake of brevity, we state the constraints by referring to (9). We state the full SDP in Appendix C.

Solving the SDP associated with these constraints is computationally expensive, even for relatively small problems. This motivates us to derive methods for constructing dcds using linear programming, by assuming the Hessian of h is diagonally dominant.

5 LP-BASED METHODS FOR CONSTRUCTING DC DECOMPOSITIONS

While Theorem 6 offers a powerful tool for constructing local dcds of polynomials over convex polytopes, the cost of solving SDPs makes them impractical for use with high-dimensional graphical models. To address this issue, we use the fact that the polynomial potential functions are a sum of a large number of low-dimensional, low-degree polynomials. Then, to escape the cost of solving SDPs, even for low-dimensional problems, we develop an LP-based approach for constructing local dcds over convex polytopes. All proofs are relegated to Appendix B.

First, we show that any polynomial can be written as the difference of two convex polynomials with diagonally dominant Hessians. This result is crucial as it facilitates the use of the conditions stated in Corollary 3 to use linear programming to construct dcds, as well as a construction of dcds that only requires solving a linear system of equations.

Lemma 3. *Suppose* Γ *is defined as in* (3), *and* $f \in \mathbb{R}[x]$. *Then, for any* $d' \geq d$, *there exists* $h \in \mathbb{R}[x]$ *of degree* d' *such that* $H_h(x)$ *and* $H_{f+h}(x)$ *are diagonally dominant and PSD for all* $x \in \Gamma$.

5.1 Positive-Definite Diagonally Dominant Matrix Polynomials

Next, we extend Theorem 3 to certify the positive-definiteness of diagonally-dominant matrix polynomials.

Theorem 7. $F \in \mathbb{S}^m(\mathbb{R}[x])$ *of degree* d *is strictly diagonally dominant and positive definite for all* $x \in \text{int}(\Gamma)$ *for* Γ *as defined in* (3), *if and only if, for some* $d' \geq d$, *there exist* B_i, $C_{ij,+}$, $C_{ij,-} \in \mathbb{R}^{\binom{K+d'}{d'}}$, *and polynomials* Q_{ij} *for each* i, j *such that for each* $i \in [m]$,

$$F_{(ii)}(x) - \sum_{j \neq i} Q_{(ij)}(x) = Z(\Lambda(x), d')^\top B_i, \tag{11}$$

$$F_{(ij)}(x) - Q_{(ij)}(x) = Z(\Lambda(x), d)^\top C_{ij,+} \quad \forall\, i \neq j \tag{12}$$

$$F_{(ij)}(x) + Q_{(ij)}(x) = Z(\Lambda(x), d)^\top C_{ij,-} \quad \forall\, i \neq j \tag{13}$$

where each element of each $B_{i,+}, B_{i,-}, C_{ij,+}, C_{ij,-}$ *is nonnegative.*

Next, as with Corollaries 1 and 2, we show that checking whether a matrix polynomial is diagonally dominant and PSD is equivalent to checking the feasibility of a set of linear constraints. If a matrix polynomial is diagonally dominant, we show that the certificate provided in Theorem 7 is equivalent to checking a set of linear feasibility constraints.

Corollary 3. $F \in \mathbb{S}^m(\mathbb{R}[x])$, *where* $F_{ij}(x) = C_{F,ij}^\top Z(x,d)$, *of degree* d *is strictly diagonally dominant and positive definite for all* $x \in \text{int}(\Gamma)$ *for* Γ *as defined in* (3), *if and only if, for some* $d' \geq d$, *there exist vectors* D_{ij}, B_{ii}, $B_{ij,+}$ *and* $B_{ij,-}$ *for all* i, j *such that, for each* i, j,

$$C_{F,ii} - \sum_{i \neq j} D_{ij} = L(\Gamma, d) B_{ii} \tag{14}$$

$$C_{F,ij} + D_{ij} = L(\Gamma, d) B_{ij,+} \tag{15}$$

$$D_{ij} - C_{F,ij} = L(\Gamma, d) B_{ij,-} \tag{16}$$

and where each B_{ii}, $B_{ij,+}$ *and* $B_{ij,-}$ *is element-wise nonnegative, and where* $L(\Gamma, d) \in \mathbb{R}^{\binom{K+d'}{d'} \times \binom{n+d'}{d'}}$ *is a known matrix.*

5.2 Linear Programming for Constructing Undominated DC Decompositions

Using Theorem 7 and Corollary 3, we derive LP-based techniques for the construction of dcds of polynomials over polytopes, that are apt for use in graphical moedls. Specifically, we optimize over those dcds $h \in \mathbb{R}[x]$ that are of degree d' and have diagonally dominant Hessians; this can be achieved by solving an LP. We formally propose this construction below.

Theorem 8 (DD-Local)**.** *Suppose* $f \in \mathbb{R}[x]$, $f(x) = C_f^\top Z(x,d)$ *is of degree* d, *and suppose* Γ *is a polytope as defined in* (3). *Then, for some* $d' \geq d$, *the linear program*

$$\begin{aligned} &\min_{\rho, C_h, B} \ \rho \\ &\text{s.t. } \rho e_1 - \sum_i A_{d',ii} C_h = L(\Gamma, d) B, \ \ B \geq 0 \\ &\text{(14)-(16) are satisfied for } H_h(x),\ H_{f+h}(x) \end{aligned} \tag{LP}$$

is feasible, and $h(x) = C_h^\top Z(x, d')$ *is a dcd of* f.

Proof Sketch. By Lemma 3, we are guaranteed the existence of a dcd with a diagonally dominant Hessian. We then apply the conditions of Corollaries 1 and 3 to the conditions of problem (5). □

5.3 Constructing dcds by solving Linear Equations

In this section, we propose a method to construct a dcd with a diagonally dominant Hessian that only requires solving a system of linear equations.

Next, we propose a method for constructing DC decompositions of polynomials over convex polytopes, by ensuring that the coefficients of the dcd in terms of Λ (for a given polytope Γ) are diagonally dominant. We construct a convex function h with a diagonal Hessian, with the additional property that every monomial of the polynomial in indeterminates Λ should have strictly positive coefficients, allowing us to ensure that the coefficients of the Hessian of $f + h$ are positive (where f is the polynomial we seek to find a dc decomposition for). We state this as the theorem below.

Theorem 9 (DD-Linear)**.** *Suppose* $f \in \mathbb{R}[x]$ *is of degree* d, *and suppose* Γ *is a convex polytope defined as in* (3). *Let* $H_f(x)$ *be the Hessian of* f, *and let* $Q_{H_f} \in \mathbb{R}[\Lambda]$, *where* $Q_{H_f}(\Lambda) = \sum_{|\alpha| \leq d'} B_\alpha \Lambda^\alpha$, *satisfy*

$H_f(x) = Q_{H_f}(\Lambda(x))$. Then, the polynomial $h \in \mathbb{R}[x]$ of even degree $d' \geq d$, where $h(x) = \sum_j c_j (x_j + 1)^{d'}$,

$$c_j = \max_\alpha \frac{\max\left\{0, 1 - B_{\alpha,ii} + \sum_j |B_{\alpha,ij}|\right\}}{(d)(d-1)\bar{G}_j}, \quad \text{(Lin)}$$

and where $\bar{G}_j$ is the largest coefficient of $(x_j + 1)^{d'}$ in the basis of monomials in $(\lambda_1, \cdots, \lambda_K)$, is a dcd of f.

While this construction of dcds is the least expensive computationally, the resultant dcds are not undominated, and thus yield slower convergence rates when employed with the CCCP. Furthermore, we propose additional dcd constructions that assumes that the Hessian of h is diagonally dominant in Appendix C.

5.4 Complexity of constructing dcds

In this section, we compare the cost of computing dcds with SDP-Local, DD-Local, and DD-Linear for a polynomial over a convex polytope $\Gamma \subset \mathbb{R}^n$ with $2n$ facets. For the sake of brevity, we state the computational complexities of constructing dcds with our proposed methods as a single result below.

Theorem 10. *Suppose $f \in \mathbb{R}[x]$, $f : \mathbb{R}^n \to \mathbb{R}$, is of degree d, and we wish to find a local dcd of f over a polytope $\Gamma \subset \mathbb{R}^n$ with $2n$ facets. Then, assuming d' is fixed, the complexity of constructing a polynomial dcd of degree $d' \geq d$ using* SDP-Local *(Theorem 6), by solving* (SDP) *to an accuracy $\varepsilon > 0$ is $\tilde{O}(n^{4d'})$;* DD-Local *(Theorem 8) by solving* (22) *to an accuracy $\varepsilon > 0$ is $\tilde{O}(n^{2d'+1})$; and* DD-Linear*, by solving* (Lin)*, assuming suitable preprocessing, is $O(n^{d'-1})$, where $\tilde{O}(\cdot)$ suppresses the* poly$(1\backslash\varepsilon)$ *terms.*

The proof for this result is provided in Appendix F.

This result highlights the tradeoffs between using different constructions of local dcds. Using SDP-Local or DD-Local would yield dcds that improve the performance of the CCCP, as we highlight in Section 6. However, using DD-Linear is significantly cheaper to compute, and can be used to obtain local dcds even for very high dimensional polynomials.

6 NUMERICAL EXPERIMENTS

In this section, we examine the efficacy of our proposed constructions for DC decompositions, both on synthetic examples, and for inference in CMRFS with polynomial potentials. For the sake of brevity, we refer readers to Appendix D for experimental details as well as additional experiments.

6.1 DC Decompositions for Synthetic Polynomials

In this section, we investigate the efficacy of SDP-Local, DD-Local, and DD-Linear for the construction of dcds. We consider two key metrics - the time taken to construct the dcd, and the number of iterations of the CCCP required to reach a stationary point.

6.1.1 Time Efficiency of Methods for DCD Construction

We consider the problem of constructing a dcd for the degree 4 polynomial $f(x) = \sum_{|\alpha| \leq 4} x^\alpha$, over the polytope $\Gamma := \{x \in \mathbb{R}^n : \sum_i x_i \geq -1,\ x_i \leq 1\}$.

We compare the time taken by DD-Local, DD-Linear nad SDP-Local for constructing dcds. For DD-Linear, we chose $h(x) = \sum_j c_j (x_j + 1)^4$. We consider $n = 2, 5, 20$. We present the experimental details in Table 1. We see that constructing dcds for a polynomial

Table 1: Comparison of times used to construct dcds with the DD-Linear, DD-Local, and the SDP-Local. Each entry contains the time taken to compute the coefficients of the dcd of a polynomial in n variables.

Algorithm	$n=2$	$n=5$	$n=20$
DD-Linear	**.062s**	**.118s**	216s
DD-Local	.656s	3.93s	1674s
SDP-Local	1.52s	61.2s	>4hrs

using DD-Linear is dramatically faster than using DD-Local and SDP-Local. Even at $n = 50$, constructing a dcd of a nonconvex polynomial is accomplished in a few hours, whereas we were unable to complete experiments using SeDuMi. Thus, we clearly establish the efficacy of DD-Linear in general, and DD-Local when the dimension of the problem is relatively small.

6.1.2 Comparison of CCCP convergence for different DC Decompositions

In this section, we empirically investigate the convergence of the CCCP when different dcd constructions are used. For our experiments, we consider polynomials $f(x) = C^\top Z(x)$ of degree 4 in 2 variables. We then randomly select coefficient vectors C, with each element $C_\alpha \in [0, 2]$. We run this experiment for 40 such cases, and apply the CCCP for each case. We report the average number of iterations for dcds obtained by each method in 2. We see that using SDP-local re-

Table 2: Comparison of the number of iterations of the CCCP required to reach a stationary point using dcds obtained via DD-Linear, DD-Local, and the SDP-Local.

Algorithm	SDP-Local	DD-Local	DD-Linear
CCCP iterations	31	42	89

sults in the fewest iterations of the CCCP, whereas DD-Linear requires the most.

6.2 MAP Inference for CMRFs

We construct DC decompositions for inference of graphical models used in polynomial SFS. First, we study the efficiency of our methods, proposed in Theorems 8 and 9 in the construction of dcds for the polynomial potential function. Second, we verify that our optimization-based methods effectively construct dcds of the potential function that achieves improved convergence rates and solution times of the convex-concave procedure.

In this experiment, we use the potential function for polynomial SFS proposed in Wang et al. (2014); Khamaru and Wainwright (2018). This is a function of the form (1), where each

$$f(z) = \sum_{r \in \mathcal{R}} \Big((l_1 p_r(z) + l_2 q_r(z) + l_3)^2 + \\ + I_r^2 \left((p_r(z)^2 + q_r(z)^2 + 1) \right)^2, \qquad (17)$$

where z is the vector of z-coordinates for the object that generated the image. For the light vector, we use $l = (0, 0, 1)$, as in (Wang et al., 2014).

6.2.1 Construction of Local DC Decompositions for Graphical Model Inference

In this section, we compute dcds for the potential function (33). We compute a dcd for each f_r using SDP-Local, DD-Local, and DD-Linear. Each $f_r(z)$ is of degree 4, and is in three unknowns. We measure the time taken to compute 1,100, 1000, and 16129 dcds (which then completes the dcd for the entire polynomial). For the Mozart example, we choose $\Gamma_r = \{z \in \mathbb{R}^3 : \ 0 \leq z \leq 600\}$ and for the Penny example. we choose $\Gamma_r = \{z \in \mathbb{R}^3 : \ 0 \leq z \leq 300\}$. As in previous sections, we compute and store $L(\Gamma_r, d)$ once, since each Γ_r is the same. We present our experimental results in Table 3 We clearly see that DD-Linear and DD-Local are able to compute dcds for each $f_r(z)$ highly efficiently. The CMRF structure enables us to optimize over dcds even in the extraordinarily high number of variables.

Table 3: Comparison of times used to compute dcds for the Mozart shape-from-shading problem. Each entry contains the time taken to compute the dcds of N polynomials $f_r(z)$ using our proposed constructions.

Algorithm	$N = 1$	$N = 100$	$N = 1000$	$N = 16128$
DD-Linear	**.097s**	**13.4s**	103.4s	1827s
DD-Local	.876s	102.9s	1081s	≈4.8 hrs
SDP-Local	2.06s	342.9s	>6hrs	n/a

6.2.2 Solving the Shape-from-Shading problem with the Convex-Concave Procedure

In this section, we examine the efficacy of different dcd constructions for solving SFS problems. We consider the Mozart and Penny datasets. For the inner loop of Algorithm 1, we use projected gradient descent, as the projections onto the hypercube can easily be computed. We detail this further in Appendix D, including stepsizes and tolerances used. We detail our results in Table 4.

Table 4: Comparison of times used to compute dcds for the Mozart shape-from-shading problem. Each entry contains the time taken to compute the dcds of N polynomials $f_r(z)$ using our proposed constructions. 'Iters' refers to iterations and 'F. value' refers to objective function value.

Dataset	**Algorithm**	**Iters.**	**F. value**	**Time**
Mozart	DD-Linear	182	1499	312s
	DD-Local	73	1531	125s
Penny	DD-Linear	139	82.9	244s
	DD-Local	68	80.7	103s

Our experiments clearly show that using DD-Local clearly outperforms DD-Linear in terms of the iterations of the CCP required. In each case, the number of CCCP iterations when dcds produced by DD-Linear is more than twice that required by DD-Local.

7 CONCLUSIONS

In this work, we study the construction of local dc decompositions of polynomials constrained to convex polytopes, with a view toward MAP inference of CMRFs. We provide characterizations for locally undominated dcds - shown to give the greatest stepwise decrease in cost function value in Ahmadi and Hall (2017) - as well as guaranteeing the existence of local dcds with diagonally dominant structure. Using Handelman's theorem, we obtain tractable methods for constructing such dcds, namely SDP-Local, DD-Local, and DD-Linear. We measure the complexity of constructing dcds with each of our proposed methods, and show that while SDP-Local obtains the best dcds, the cost of constructing dcds using DD-Local or DD-Linear is significantly cheaper. We then validate our methods empirically, and see that our method outperforms the nearest baselines. However, there are still various unaddressed problems. First, the convergence rate of the CCCP for polynomial or smooth dcds is still not well understood, and second, systematically constructing dcds for non-polynomial cost functions remains an open problem.

ACKNOWLEDGEMENTS

The authors Chaitanya Murti and Dhruva Kashyap are supported by a generous grant from Shell India.

References

H. Abbaszadehpeivasti, E. de Klerk, and M. Zamani. On the rate of convergence of the difference-of-convex algorithm (dca). *Journal of Optimization Theory and Applications*, pages 1–22, 2023.

A. A. Ahmadi and G. Hall. Dc decomposition of nonconvex polynomials with algebraic techniques. *Mathematical Programming*, Apr 2017.

A. A. Ahmadi and A. Majumdar. Dsos and sdsos optimization: Lp and socp-based alternatives to sum of squares optimization. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–5. IEEE, 2014.

A. Asadi, K. Chatterjee, H. Fu, A. K. Goharshady, and M. Mahdavi. Polynomial reachability witnesses via stellensätze. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 772–787, 2021.

S. Bach, M. Broecheler, L. Getoor, and D. O'leary. Scaling mpe inference for constrained continuous markov random fields with consensus optimization. *Advances in Neural Information Processing Systems*, 25, 2012.

S. H. Bach, M. Broecheler, B. Huang, and L. Getoor. Hinge-loss markov random fields and probabilistic soft logic. *Journal of Machine Learning Research*, 18(109):1–67, 2017.

H. Bauermeister, E. Laude, T. Mollenhoff, M. Moeller, and D. Cremers. Lifting the convex conjugate in lagrangian relaxations: A tractable approach for continuous markov random fields. *SIAM Journal on Imaging Sciences*, 15(3):1253–1281, 2022.

J. Bednarik, P. Fua, and M. Salzmann. Learning to reconstruct texture-less deformable surfaces from a single view. In *2018 international conference on 3d vision (3DV)*, pages 606–615. IEEE, 2018.

D. Boland and G. A. Constantinides. Automated precision analysis: A polynomial algebraic approach. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, pages 157–164. IEEE, 2010.

I. M. Bomze and C. Lemaréchal. Necessary conditions for local optimality in difference-of-convex programming. *J. Convex Anal*, 17(2):673–680, 2010.

I. M. Bomze and M. Locatelli. Undominated d.c. decompositions of quadratic functions and applications to branch-and-bound approaches. *Computational Optimization and Applications*, 28(2):227–245, Jul 2004. ISSN 1573-2894. doi: 10.1023/B:COAP.0000026886.61324.e4. URL `https://doi.org/10.1023/B:COAP.0000026886.61324.e4`.

S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

C. Briat. Dwell-time stability and stabilization conditions for linear positive impulsive and switched systems. *Nonlinear Analysis: Hybrid Systems*, 24:198–226, 2017.

M. Chiang. Nonconvex optimization for communication networks. *Advances in Applied Mathematics and Global Optimization: In Honor of Gilbert Strang*, pages 137–196, 2009.

E. De Klerk and M. Laurent. Error bounds for some semidefinite programming approaches to polynomial minimization on the hypercube. *SIAM Journal on Optimization*, 20(6):3104–3120, 2010.

E. De Klerk, J. B. Lasserre, M. Laurent, and Z. Sun. Bound-constrained polynomial optimization using only elementary calculations. *Mathematics of Operations Research*, 42(3):834–853, 2017.

B. Deng, Y. Yao, R. M. Dyke, and J. Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, volume 41, pages 559–589. Wiley Online Library, 2022.

M. Dür. Conditions characterizing minima of the difference of functions. *Monatshefte für Mathematik*, 134:295–303, 2002.

A. Ecker and A. D. Jepson. Polynomial shape from shading. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 145–152. IEEE, 2010.

O. Faust, H. Fawzi, and J. Saunderson. A bregman divergence view on the difference-of-convex algorithm. In *International Conference on Artificial Intelligence and Statistics*, pages 3427–3439. PMLR, 2023.

D. Handelman et al. Representing polynomials by positive linear functions on compact convex polyhedra. *Pac. J. Math*, 132(1):35–62, 1988.

P. Hartman et al. On functions representable as a difference of convex functions. *Pacific J. Math*, 9 (3):707–713, 1959.

A. Ihler and D. McAllester. Particle belief propagation. In *Artificial intelligence and statistics*, pages 256–263. PMLR, 2009.

M. I. Jordan. Graphical models. 2004.

R. Kamyar and M. M. Peet. Polynomial optimization with applications to stability analysis and control - alternatives to sum of squares. *Discrete & Continuous Dynamical Systems - B*, 20, 2015. ISSN 1531-3492. doi: 10.3934/dcdsb.2015.20.2383. URL http://aimsciences.org//article/id/f4f6bc09-8cd5-4a41-b131-b63ee5a37a2d.

R. Kamyar, C. Murti, and M. M. Peet. Constructing piecewise-polynomial lyapunov functions for local stability of nonlinear systems using handelman's theorem. In *53rd IEEE Conference on Decision and Control*, pages 5481–5487. IEEE, 2014.

K. Khamaru and M. Wainwright. Convergence guarantees for a class of non-convex and non-smooth optimization problems. In *International Conference on Machine Learning*, pages 2601–2610. PMLR, 2018.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

G. Lanckriet and B. K. Sriperumbudur. On the convergence of the concave-convex procedure. *Advances in neural information processing systems*, 22, 2009a.

G. R. Lanckriet and B. K. Sriperumbudur. On the convergence of the concave-convex procedure. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1759–1767. 2009b.

J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. doi: 10.1137/S1052623400366802. URL https://doi.org/10.1137/S1052623400366802.

J. B. Lasserre. Polynomial programming: Lp-relaxations also converge. *SIAM Journal on Optimization*, 15(2):383–393, 2005.

F. Latorre, P. T. Y. Rolland, and V. Cevher. Lipschitz constant estimation for neural networks via sparse polynomial optimization. In *8th International Conference on Learning Representations*, number CONF, 2020.

C.-T. Lê and Thá-Hòa-Bình. Handelman's positivstellensatz for polynomial matrices positive definite on polyhedra. *Positivity*, Aug 2017.

H. A. Le Thi and T. Pham Dinh. Dc programming and dca: thirty years of developments. *Mathematical Programming*, 169(1):5–68, 2018.

H. A. Le Thi, M. C. Nguyen, and T. P. Dinh. A dc programming approach for finding communities in networks. *Neural computation*, 2014.

T. Lipp and S. Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287, Jun 2016. ISSN 1573-2924. doi: 10.1007/s11081-015-9294-x. URL https://doi.org/10.1007/s11081-015-9294-x.

S. Marx, T. Weisser, D. Henrion, and J. Lasserre. A moment approach for entropy solutions to nonlinear hyperbolic pdes. *arXiv preprint arXiv:1807.02306*, 2018.

A. Nemirovski. Interior point polynomial time methods in convex programming. *Lecture notes*, 42(16):3215–3224, 2004.

Y.-S. Niu. On difference-of-sos and difference-of-convex-sos decompositions for polynomials. *arXiv preprint arXiv:1803.09900*, 2018.

P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, California Institute of Technology, 2000.

P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, May 2003. ISSN 1436-4646. doi: 10.1007/s10107-003-0387-5. URL https://doi.org/10.1007/s10107-003-0387-5.

B. Piot, M. Geist, and O. Pietquin. Difference of convex functions programming for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2519–2527, 2014.

B. Piot, M. Geist, and O. Pietquin. Difference of convex functions programming applied to control with expert data. *arXiv preprint arXiv:1606.01128*, 2016.

V. Powers and B. Reznick. A new bound for pólya's theorem with applications to polynomials positive on polyhedra. *Journal of pure and applied algebra*, 164(1-2):221–229, 2001.

M. Putinar. Positive polynomials on compact semialgebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC, 2005.

R. R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani. On the convergence of bound optimization algorithms. *arXiv preprint arXiv:1212.2490*, 2012.

M. Salzmann. Continuous inference in graphical models with polynomial energies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1744–1751, 2013.

S. Sankaranarayanan, X. Chen, et al. Lyapunov function synthesis using handelman representations. *IFAC Proceedings Volumes*, 46(23):576–581, 2013.

N. R. Shah, A. Misra, A. Miné, R. Venkat, and R. Upadrasta. Bullseye: Scalable and accurate approximation framework for cache miss calculation.

ACM Transactions on Architecture and Code Optimization, 20(1):1–28, 2022.

X. Shen, S. Diamond, Y. Gu, and S. Boyd. Disciplined convex-concave programming. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1009–1014, Dec 2016. doi: 10.1109/CDC.2016.7798400.

B. K. Sriperumbudur and G. R. Lanckriet. A proof of convergence of the concave-convex procedure using zangwill's theory. *Neural computation*, 24(6):1391–1407, 2012.

P. D. Tao and L. T. H. An. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

C. Wang, N. Komodakis, and N. Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117 (11):1610–1627, 2013.

S. Wang, A. Schwing, and R. Urtasun. Efficient inference of continuous markov random fields with polynomial potentials. *Advances in neural information processing systems*, 27, 2014.

S. Wang, S. Fidler, and R. Urtasun. Proximal deep structured models. *Advances in Neural Information Processing Systems*, 29, 2016.

T. Winkler, S. Junges, G. A. Pérez, and J.-P. Katoen. On the complexity of reachability in parametric markov decision processes. *arXiv preprint arXiv:1904.01503*, 2019.

Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. Zickler. From shading to local shape. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):67–79, 2014.

K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 45–58. Springer, 2012.

A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.

A. Yurtsever and S. Sra. Cccp is frank-wolfe in disguise. *Advances in Neural Information Processing Systems*, 35:35352–35364, 2022.

L. F. Zuluaga, J. Vera, and J. Pena. Lmi approximations for cones of positive semidefinite forms. *SIAM Journal on Optimization*, 16(4):1076–1091, 2006.

APPENDIX

In this Appendix, we present the following additional information and context.

1. In Appendix A, we provide a detailed discussion on relevant related literature. In particular, we discuss inferencing continuous Markov Random Fields, fundamental work on the construction of dc decompositions, and additional analysis and applications Handelman's theorem.
2. In Appendix B, we provide the proofs missing in the main paper. We also state the optimization problems involved in Theorems 6 and 8.
3. In Appendix C, we propose additional constructions of local dcds of polynomials. In particular, we detail LP- and coefficient-based approaches for constructing dcds whose Hessians have Handelman representations with diagonally dominant coefficients. We also propose an SOS-based construction of local dcds.
4. In Appendix D, we detail additional experiments we conducted that highlight the efficacy of using our local construction of dcds. We also provide additional detail and context to experiments described in section 6.
5. In Appendix E, we describe the polynomial shape-from shading problem in greater detail.
6. In Appendix F, we provide an analysis of the complexity of solving the optimization problems involved in SDP-Local, DD-Local, and DD-Linear.
7. Lastly, our code is available at: `https://github.com/chaimurti/DC-Handelman`

A RELATED WORK

In this section, we provide a more detailed survey of relevant literature. We separate this section into three parts - first, we discuss recent work on continuous markov random fields. Next, we discuss recent work on difference of convex programming. Last, we provide an overview of Handelman's theorem and some prior applications of it.

A.1 Continuous Markov Random Fields

Continuous Markov Random Fields, also known as undirected graphical models (UGMs), have been used for a variety of tasks in machine learning (Koller and Friedman, 2009; Jordan, 2004). Typically, Markov Random Fields, model collections of discrete random variables Koller and Friedman (2009). However, in many applications, the random variables may not be discrete. In Gaussian Markov Random Fields, these are considered to be Gaussian (Wang et al., 2013; Rue and Held, 2005), and have a variety of applications such as image denoising and segmentation. Continuous Markov Random Fields dispense with the Gaussianity assumption on the random variables. Such models have a variety of applications, including stereo estimation (Yamaguchi et al., 2012), collaborative filtering Koren et al. (2009), and various other image processing tasks such as shape-from-shading (Ecker and Jepson, 2010; Bednarik et al., 2018; Xiong et al., 2014), and non-rigid 3D reconstruction (see Deng et al. (2022) and the references therein). However, learning and inference on such models is a challenging tasks, as noted in Wang et al. (2014). Popular methods include using discrete particle approximations (Ihler and McAllester, 2009), specialized models that are amenable to convex optimization (Bach et al., 2012, 2017), proximal methods (Wang et al., 2016), nonlinear optimization techniques such as dual-decompositions (Bauermeister et al., 2022), or the convex-concave procedure (Wang et al., 2014). In this work, we address the problem of CMRF inference using the CCCP.

A.2 Difference of Convex Programming and the Convex-Concave Procedure

Difference-of-convex programming is a well-studied field, and has been applied to a wide variety of applications (Le Thi et al., 2014; Le Thi and Pham Dinh, 2018). Difference-of-convex programs are solved using the DC Algorithm (Tao and An, 1997), when the dcd is nonsmooth, and the convex-concave procedure for differentiable dcds (Yuille and Rangarajan, 2003). Variants of this algorithm, including for DC programming with smooth DC constraints, were proposed, such as Lipp and Boyd (2016); Shen et al. (2016). However, several open problems remain. Most importantly, understanding the convergence properties of these algorithms remains an open problem, however. Initial convergence results for the CCCP were stated in Lanckriet and Sriperumbudur (2009b); Sriperumbudur and Lanckriet (2012). In Salakhutdinov et al. (2012), the CCCP is described as a 'bound optimization algorithm', and connected the convergence rate of the CCCP to the curvature of the objective function, thus providing a connection to works such as Bomze and Locatelli (2004); Dür (2002); Bomze and Lemaréchal (2010). More recently, convergence results for the CCCP were given for the case when the cost function and the dcd are Lipschitz smooth (Khamaru and Wainwright, 2018). Similarly, in Abbaszadehpeivasti et al. (2023), convergence results for the DCA are proposed for the case where the Lipshitz constants of the dcd are available. Other works investigate connections between the CCCP and Frank-Wolfe algorithms (Yurtsever and Sra, 2022), and Bregman divergences (Faust et al., 2023). However, as noted in Abbaszadehpeivasti et al. (2023), convergence analysis of the CCCP when the objective function and the dcds are polynomials, or even twice differentiable, remain open problems.

A.3 Handelman's Theorem and Applications

Handelman's theorem, first introduced in Handelman et al. (1988), is a significant result in algebraic geometry, with particular importance to polynomial optimization. For polynomial optimization problems where the feasible set is a convex polytope, Handelman's theorem offers a more efficient alternative to SOS programming, as it allows for the derivation of LP-based relaxations for such problems (Kamyar and Peet, 2015; Zuluaga et al., 2006). Moreover, it has been shown that the hierarchy of LP-relaxations derived from Handelman's theorem converges (Lasserre, 2005). Other works, such as (De Klerk and Laurent, 2010; De Klerk et al., 2017) investigate the error bounds of the relaxations to polynomial optimization problems obtained by Handelman's theorem. Handelman's theorem has also been successfully applied to a variety of real-world problems. In control engineering, it has been applied to Lyapunov function synthesis (Kamyar et al., 2014; Sankaranarayanan et al., 2013; Briat, 2017) and reachability analysis (Asadi et al., 2021; Winkler et al., 2019). Other interesting applications of Handelman's theorem include network utility maximization (Chiang, 2009), solving polynomial hyperbolic partial differential equations (Marx et al., 2018), cache miss calculations in program verification (Shah et al., 2022), automated precision analysis for programs deployed on specialized hardware (Boland and Constantinides, 2010), and Lipschitz constant estimation for neural networks (Latorre et al., 2020).

B MISSING PROOFS OF MAIN RESULTS

In this section, we provide proofs for the main results. Proofs are presented in the order in which the original results appeared in the main paper.

Proof of Theorem 5

We prove Theorem 5, a key result used throughout this work to motivate the construction of local dcds by solving convex optimization problems.

Proof. We prove the theorem in 4 steps.

Step 1: First, we show that (4) is a convex optimization problem.We know the feasible set (set of dcds for a polynomial) is convex. Next, we show that $\max_{x\in\Gamma} \operatorname{Tr}(H_h(x))$ is convex in the set of feasible h. Consider two feasible functions $h_1,\ h_2$. Then,

$$\begin{aligned}&\max_{x\in\Gamma} \operatorname{Tr}\left(\alpha H_{h_1}(x) + (1-\alpha) H_{h_1}(x)\right)\\ \leq&\alpha \max_{x\in\Gamma} \operatorname{Tr}(H_{h_1}(x)) + (1-\alpha) \max_{x\in\Gamma} \operatorname{Tr}(H_{h_2}(x))\end{aligned}$$

since $\operatorname{Tr}(H_{h_1}(x)),\ \operatorname{Tr}(H_{h_2}(x)) \geq 0$ by the convexity of $h_1,\ h_2$.

Step 2: Let h' be a feasible solution to (4) such that $\max_{x\in\Gamma} \mathrm{Tr}(H_{h'}(x)) = \gamma$. Consider the following optimization problem:

$$\min_{h\in\mathbb{R}[x]} \quad \max_{x\in\Gamma} \; \mathrm{Tr}\left(H_h(x)\right)$$
$$h,\; f+h \text{ are convex on } \Gamma,\; \max_{x\in\Gamma} \mathrm{Tr}(H_h(x)) \leq \gamma \tag{P2}$$

Clearly, any optimal solution to (P2) is a solution to (4) and vice versa. Let U be the feasible set of (P2). Similar to Ahmadi and Hall (2017), we see that U is closed. Next, we show boundedness. Suppose some $h \in U$ has a monomial that is at least quadratic in one variable that coefficient c_β for each $\beta > 0$ such that $c_\beta > 0$. Thus, under this assumption, there is a diagonal entry of H_h that can get arbitrarily large. However, by the definition of U, we have that $\max_{x\in\Gamma} \mathrm{Tr}(H_h(x)) = \max \|H_h(x)\|_* \leq \gamma$, where $\|\cdot\|_*$ denotes the nuclear norm. This is a contradiction, and thus, we show that the set is bounded as well. Since the feasible set is compact, and the cost function of (P2) is convex, we see that (P2) is guaranteed to have a solution, thereby guaranteeing that (4) has a solution.

Step 3: We now show that optimal solutions are undominated by suboptimal dcds. Suppose ρ^* is the optimal value of (4). Then, by convexity, any optimal solution to (4) yields the objective function value ρ^*. Next, let h^* be an optimal solution to (4), and let h be a suboptimal feasible solution to (4). Suppose that h^* is not undominated, and that $h^* - h$ is convex. Then, $\min_{x\in\Gamma}\ \mathrm{Tr}(H_{h^*-h}(x)) \geq 0$. However, we see that

$$\mathrm{Tr}(H_{h^*-h}(x)) = \mathrm{Tr}(H_{h^*}(x)) - \mathrm{Tr}(H_h(x)) \leq \rho^* - \mathrm{Tr}(H_h(x)).$$

Thus, we have

$$\min_{x\in P} \mathrm{Tr}(H_{h^*-h}(x)) \leq \min_{x\in P} \rho^* - \mathrm{Tr}(H_h(x)) = \rho^* - \max_{x\in P} \mathrm{Tr}(H_h(x)).$$

However, $\max_{x\in P} \mathrm{Tr}(H_h(x)) > \rho^*$. Thus, we obtain a contradiction, and we show that optimal solutions to (4) are undominated by suboptimal feasible points.

Step 4: Last, we show that distinct optimal solutions are undominated by each other. Let $h_1,\ h_2$ be distinct optimal solutions of (4). Then, by the feasibility of (4), $\max_{x\in P} \mathrm{Tr}(H_{h_1}(x) + H_{h_2}(x)) \geq \rho^*$. However,

$$\begin{aligned}\max_{x\in P} \mathrm{Tr}(H_{h_1-h_2}(x)) &= \max_{x\in P} \mathrm{Tr}(H_{h_1}(x) - (H_{h_2}(x)) \\ &\leq \max_{x\in P} \mathrm{Tr}(H_{h_1}(x)) - \min_{x\in P} \mathrm{Tr}(H_{h_2}(x)) \leq \rho^*\end{aligned}$$

Therefore, $\max_{x\in P} \mathrm{Tr}(H_{h_1-h_2}(x)) = \rho^*$. Since the degree of h is at least 3, the Hessian is non-constant. Furthermore, suppose $\max_{x\in P} \mathrm{Tr}(H_{h_1-h_2}(x)) = \rho^*$. This is achieved when $x = \arg\max\ \mathrm{Tr}(H_{h_1}(x))$, which must coincide with $\mathrm{Tr}(H_{h_2}(x)) = 0$. Since
$\max \mathrm{Tr}(H_{h_1}(x)) = \rho*$, there exists an x where $\mathrm{Tr}(H_{h_1}(x)) < \mathrm{Tr}(H_{h_2}(x))$, thereby implying that $h_1 - h_2$ is nonconvex. Thus, we obtain a contradiction and prove the theorem.

□

Proof of Lemma 1

We present the proof for Lemma 1.

Proof of Lemma 1. Ensuring that $q_f(a_1^\top x + b_1, \cdots, a_L^\top x + b_K) = f(x_1, \cdots, x_n)$ is equivalent to matching the coefficients. Observe that we can expand each monomial $\lambda_1(x)^{\alpha_1} \cdots \lambda_K(x)^{\alpha_K}$ as

$$\lambda_1(x)^{\alpha_1} \cdots \lambda_K(x)^{\alpha_K} = \sum_{|\alpha|\leq d} c_{\alpha'} x_1^{\alpha'_1} \cdots x_K^{\alpha'_K}$$

by substituting $\lambda_i(x) = a_i^\top x + b_i$. From this, it follows that $Z_\alpha(\Lambda(x), d') = l_\alpha(\Gamma)^\top Z(x, d')$ From this, we can write $q_f(\Lambda(x), d') = B^\top L Z(x, d')$, where L is a matrix with rows l_α. If $f(x) = q_f(\Lambda(x))$, then $C = L(\Gamma, d')^\top B$. We can solve for B by solving an underdetermined set of linear equations of full rank. Thus, a solution always exists. □

Proof of Corollary 1

Here, we present the proof for Corollary 1.

Proof of Corollary 1. We are given that $f(x) > 0$ for $x \in \Gamma$. Thus, by Handelman's theorem (Theorem 3, for some $d' \geq d$, there must exist a polynomial $q(\Lambda) = B^\top Z(\Lambda, d')$ with nonnegative coefficients such that $q(\Lambda(x)) = f(x)$. Thus, $C^\top Z(x, d) = B^\top Z(\Lambda(x), d')$. But by Lemma 1, we have that $C = L(\Gamma, d')B$. Thus, the statement is proved. □

Proof of Lemma 2

We present the proof for Lemma 2, which extends Lemma 1 to matrix polynomials.

Proof of Lemma 2. From Lemma 1, we know that for each i, j, $F_{ij}(x) = Z(x, d')^\top C^{ij}$, and $Q_{F,ij}(\Lambda) = Z(\Lambda, d')^\top B^{ij}$, where $C_{\alpha,ij} = C^{ij}_\alpha$, $B_{\alpha,ij} = B^{ij}_\alpha$, and where $C^{ij} = L(\Gamma, d')^\top B^{ij}$, with $L(\Gamma, d')$ defined as in Lemma 1. Thus, $C_{\alpha,ij} = C^{ij}_\alpha = l_\alpha(\Gamma)^\top B^{ij}$.Thus, we have $w_{\alpha,\alpha'} = (l_\alpha(\Gamma))_{\alpha'}$, which is the entry of $l_\alpha(\Gamma)$ corresponding to the monomial $x^{\alpha'}$ (in the multi-index notation). Since this holds for all i, j and since $l_\alpha(\Gamma)$ is fixed, the statement is proved. □

Proof of Corollary 2

Following the proof of Lemma 2, we present the proof for Corollary 2.

Proof of Corollary 2. We are given that $F(x) \succ 0$ for $x \in \Gamma$. Thus, by Theorem 4, for some $d' \geq d$, there must exist a polynomial $Q(\Lambda) = \sum_\alpha B_\alpha \Lambda^\alpha$ with PSD coefficients such that $Q(\Lambda(x)) = F(x)$. Thus, $C^\top Z(x, d) = B^\top Z(\Lambda(x), d')$. By Lemma 2, the proof follows in the same fashion as that of Corollary 1. □

Proof of Theorem 6

In this section, we provide a proof for Theorem 6, which guarantees the feasibility of the SDP-Local method. We first restate the Theorem and include the SDP in full.

Theorem (SDP-Local)**.** *Suppose $f \in \mathbb{R}[x]$, $f(x) = C_f^\top Z(x, d)$ is of degree d, and suppose Γ is a polytope as defined in* (3)*. Then, for some $d' \geq d$, the SDP*

$$\min_{\mathcal{A}} \ \rho$$

$$\text{s.t.} \sum_{2 \leq |\alpha'| \leq d'} C_{h,\alpha'} M_{\alpha,\alpha'} = \sum_{\alpha'' \leq |d'|} (l_\alpha(\Gamma))_{\alpha''} B_{h,\alpha''} \tag{18}$$

$$\sum_{2 \leq |\alpha'| \leq d'} (C_{h,\alpha'} + C_{f,\alpha'}) M_{\alpha,\alpha'} = \sum_{\alpha'' \leq |d'|} (l_\alpha(\Gamma))_{\alpha''} B_{f+h,\alpha''} \tag{19}$$

$$\rho e_i + \sum_{2 \leq |\alpha'| \leq d'} C_{h,\alpha'} \mathrm{Tr}(M_{\alpha,\alpha'}) = L(\Gamma, d) B \tag{20}$$

$$B \geq 0, \ B_{h,\alpha''} \succ 0, \ B_{f+h,\alpha''} \succ 0 \quad \text{for all } \alpha'' \tag{21}$$

is feasible, where

$$\mathcal{A} := \left\{\rho, \ B, \ C_h, \ \{B_{h,\alpha}\}_{|\alpha| \leq d'}, \{B_{f+h,\alpha}\}_{|\alpha| \leq d'}\right\}$$

with B being an elementwise nonnegative vector, and $h(x) = C_h^\top Z(x, d')$ is a dcd of f.

Proof. Since f is a polynomial, we are guaranteed to find a polynomial dcd h of f. Since $H_h(x) \succeq 0$ and $H_{f+h}(x) \succeq 0$ for all $x \in \Gamma$, (18)-(21) follow from Theorem 4 and Corollary 2. Last, we write $\rho - \mathrm{Tr}(H_h(x)) = \rho - \sum_\alpha x^\alpha \left(\sum_{2 \leq |\alpha'| \leq d'} C_{h,\alpha'} \mathrm{Tr}(M_{\alpha,\alpha'})\right)$. Then, (20) follows from Theorem 3 and Corollary 1. □

Proof of Theorem 7

Proof of Theorem 7. First, we show necessity. Assuming $F(x) \succ 0$ for all $x \in \Gamma$, it follows that $F_{(ii)}(x) > 0$ for all $x \in \Gamma$ and each $i \in [m]$. Furthermore, since $F(x)$ is strictly diagonally dominant, we have

$$F_{(ii)}(x) - \sum_{j \neq i} |F_{(ij)}(x)| > 0$$

Let $Q_{ij}(x) \leq |F_{(ij)}(x)|$. Thus, it follows that $F_{ij}(x) + Q_{ij}(x) \geq 0$ and $F_{ij}(x) - Q_{ij}(x) \geq 0$ Thus, it follows that

$$F_{(ii)}(x) > \sum_{j \neq i} Q_{(ij)}(x) \text{ and } F_{(ii)}(x) > -\sum_{j \neq i} Q_{(ij)}(x)$$

From Theorem 3, for each i, j, there must exist elementwise nonnegative vectors B_i, $C_{ij,+}$, $C_{ij,-}$ such that

$$\begin{aligned} F_{(ii)}(x) - \sum_{j \neq i} F_{(ij)}(x) &= Z(\Lambda(x), d)^\top B_i \\ F_{ij}(x) + Q_{ij}(x) &= Z(\Lambda(x), d)^\top C_{ij,+} \\ F_{ij}(x) - Q_{ij}(x) &= Z(\Lambda(x), d)^\top C_{ij,-}. \end{aligned}$$

Thus, we show necessity. To show sufficiency, we recall from Ahmadi and Majumdar (2014) that, given $P \in \mathbb{R}^{n \times n}$, if for each $i \in [n]$, there exist real numbers R_{ij}, $j \in [n]$ such that $P_{ii} - \sum_{j \neq i} R_{ij} \geq 0$ and $-R_{ij} \leq P_{ij} \leq R_{ij}$, then P is diagonally dominant and PSD. We replace P with $F(x)$, R_{ij} with $Q_{ij}(x)$ and apply Theorem 3 to the conditions listed above, obtaining equations (11)-(13). □

Proof of Lemma 3

In this section, we prove Lemma 3, which guarantees the existence of a local dcd with a diagonally dominant Hessian. To prove this, we first show that we are guaranteed to find a local dcd with a *diagonal* Hessian.

Lemma 4. *Suppose* Γ *is defined as in* (3)*, and* $f \in \mathbb{R}[x]$*. Then, there exists* $h \in \mathbb{R}[x]$ *such that* $H_h(x)$ *is PSD and diagonal and* $H_{f+h}(x)$ *is PSD and diagonally dominant for all* $x \in \Gamma$*.*

Proof. Let H_f be the Hessian of f. Let

$$z_i = \max_{x \in \Gamma} \sum_{j \neq j} |H_{f,ij}(x)| - H_{f,ii}(x).$$

Since Γ is compact, and $H_{f,ij}(x)$ is smooth for each i, j, $z_i < \infty$. Next, for each $i \in [n]$, let $h_i(x_i)$ be a convex polynomial in x_i such that $w_i = \min_{x \in \Gamma} h_i''(x_i) > 0$, where $h_i''(x) = \frac{\partial^2}{\partial x_i^2} h(x)$. Let $h(x) = \sum_{i \in [n]} c_i h_i(x)$. Then, for each i, $H_{f+h,ii}(x) = c h_i''(x) + H_{f,ii}(x)$, and

$$\begin{aligned} H_{f+h,ii}(x) - \sum_{i \neq j} |H_{f+h,ij}(x)| &= c h_i''(x) + H_{f,ii}(x) - \sum_{i \neq j} |H_{f,ij}(x)| \\ &= c h_i''(x) - \left(\sum_{i \neq j} |H_{f,ij}(x)| - H_{f,ii}(x) \right) \geq c w_i - z_i \ \forall, \ x \in \Gamma. \end{aligned}$$

Thus, we can choose $c_i = 1 + z_i / w_i$, ensuring that $H_{f+h}(x)$ is diagonally dominant for all $x \in \Gamma$. □

The construction used in the proof of Lemma 4 is used again in the proof of Lemma 3, which we now present.

Proof of Lemma 3. We prove this Lemma by construction. Let $\bar{h}$ be a polynomial dcd of polynomial f, which is guaranteed to exist. Define

$$\begin{aligned} y_i &= \max_{x \in \Gamma} \sum_{j \neq j} |H_{\bar{h},ij}(x)| - H_{\bar{h},ii}(x) \\ z_i &= \max_{x \in \Gamma} \sum_{j \neq j} |H_{f+\bar{h},ij}(x)| - H_{f+\bar{h},ii}(x) \end{aligned}$$

Similar to the proof of Lemma 4, let $h_i(x_i)$ be a convex polynomial in x_i such that $w_i = \min_{x\in\Gamma} h_i''(x_i) > 0$, where $h_i''(x) = \frac{\partial^2}{\partial x_i^2} h(x)$. Let $h(x) = \sum_i c_i h_i(x_i)$, where each $c_i > 0$. Using a similar construction used in the proof of Lemma 4, if we require $c_i > \max\{y_i, z_i\}/w_i$. Thus, we can choose $c_i = 1 + \max\{y_i, z_i\}/w_i$, which guarantees that $\bar{h} + \sum_i c_i h_i$ is a diagonally dominant dcd of f. □

Proof of Theorem 8

In this section, we prove Theorem 8, which guarantees the feasibility of the DD-Local method. We first restate Theorem 8 with the LP involved in DD-Local stated in full.

Theorem (DD-Local)**.** *Suppose $f \in \mathbb{R}[x]$, $f(x) = C_f^\top Z(x, d)$ is of degree d, and suppose Γ is a polytope as defined in* (3)*. Then, for some $d' \geq d$, the linear program*

$$\begin{aligned}
\min_{\mathcal{A}} \quad & \rho \\
\text{s.t.} \quad & \rho e_1 - \sum_i A_{d',ii} C_h = L(\Gamma, d)B, \quad B \geq 0 \\
& A_{d',ii} C_h - \sum_{i\neq j} D_{h,ij} = L(\Gamma, d) B_{h,ii} \quad \textit{for all } i \\
& A_{d',ij} C_h + D_{h,ij} = L(\Gamma, d) B_{h,ij,+} \quad \textit{for all } i, j \\
& D_{h,ij} - A_{d',ii} C_h = L(\Gamma, d) B_{h,ij,-} \quad \textit{for all } i, j \\
& A_{d',ii}(C_h + C_f) - \sum_{i\neq j} D_{f+h,ij} = L(\Gamma, d) B_{f+h,ii} \quad \textit{for all } i, j \\
& A_{d',ij}(C_h + C_f) + D_{f+h,ij} = L(\Gamma, d) B_{f+h,ij,+} \quad \textit{for all } i, j \\
& D_{f+h,ij} - A_{d',ii}(C_h + C_f) = L(\Gamma, d) B_{f+h,ij,-} \quad \textit{for all } i, j \\
& B_{h,ii} \geq 0, \ B_{h,ij,+} \geq 0, \ B_{h,ij,-} \geq 0 \quad \textit{for all } i, j \\
& B_{f+h,ii}, \geq 0, \ B_{f+h,ij,+} \geq 0, \ B_{f+h,ij,-} \geq 0 \quad \textit{for all } i, j
\end{aligned} \tag{22}$$

where

$$\mathcal{A} := \{\rho, B, C_h, \{D_{h.ij}\}_{i\neq j}, \{B_{h,ii}\}_{i=1}^n, \{D_{f+h.ij}\}_{i\neq j}, \{B_{f+h,ii}\}_{i=1}^n\}$$

is feasible, and $h(x) = C_h^\top Z(x, d')$ is a dcd of f.

Proof. By Lemma 3, any polynomial f is guaranteed to have a dcd h with a Hessian that is diagonally dominant on a given polytope Γ. The constraints are directly derived from Theorem 7 and Corollary 3. □

Proof of Theorem 9

In this section, we prove Theorem 9, which provides a simple construction of local dcds that only requires solving a linear equation. To begin, we first state the following technical Lemma.

Lemma 5. *Suppose Γ is a convex polytope defined as in* (3)*. Then, for each j and each $\varepsilon \geq 0$, there exist scalars $k_{ji} > \varepsilon$ such that*

$$x_j = \sum_j k_{ji} a_i. \tag{23}$$

Proof. A polyhedron $P := \{x : Ax \geq b\}$, $A = [a_1^\top, \cdots, a_K^\top]^\top$ is unbounded if there exists a point z and a vector d such that $x + \eta d \in \Gamma$ for all $\eta \geq 0$. This implies the set $\{d \in \mathbb{R}^n : Ad \geq 0\}$ is nonempty iff P is unbounded. However, we assume that Γ is bounded, and hence, the set $\{x : Ax \geq 0\}$ is empty. Next, we see that by matching coefficients, finding $k_{ji} > \epsilon$ such that $\sum_i k_{ji} a_i = x_j$ where $k_{ji} \geq \varepsilon$ is equivalent to solving

$$A^\top k_j = e_j - \varepsilon A^\top \mathbf{1}, \ k_j \geq 0 \text{ entrywise} \tag{24}$$

where e_j is the j-th coordinate vector and $\mathbf{1}$ is the vector of ones. Then, by Farkas' Lemma, (24) is feasible iff there exists no vector y s.t. $Ay \geq 0$ and $(e_j - \varepsilon A^\top \mathbf{1})^\top y < 0$. Since the only feasible $y = 0$, $(e_j - \varepsilon A^\top \mathbf{1})^\top y = 0$, thus proving the statement. □

Proof. Let $d' \geq d$ be an even integer. By Lemma 4, we are guaranteed that f can be written as the difference of a diagonal and a diagonally dominant matrix. Next, we can choose

$$h(x) = \sum_j c_j (x_j + \eta_j + 1)^{d'},$$

where $x_j + \eta_j + 1 = 1 + \sum_i k_{ji}\lambda_i(x)$. Thus,

$$q_h(\Lambda) = \sum_j c_j (1 + k_{j1}\lambda_1 + \cdots + k_{jK}\lambda_K)^{d'} = \sum_{|\alpha| \leq d'} g_{i,\alpha}\Lambda^\alpha,$$

where crucially, each $g_{i,\alpha} > 0$. Similarly, we see that

$$H_{h,jj}(x) = c_j d'(d'-1)(x_j + \eta_j + 1)^{d'-2}$$

has polytopic representation

$$\begin{aligned} Q_{H_h,jj}(\Lambda) &= c_j d'(d'-1)(1 + k_{j1}\lambda_1 + \cdots + k_{jK}\lambda_K)^{d'-2} \\ &= \sum_{|\alpha| \leq d'} g'_{j,\alpha}\Lambda^\alpha, \end{aligned}$$

again, with all $g'_{j,\alpha} > 0$. Thus, the coefficients G_α of $Q_{H_h}(\Lambda)$ are diagonal and positive definite. Next, by Lemma 2, we can write the polytopic form of H_f, as $Q_{H_f} = \sum_{|\alpha| \leq d'} B_\alpha \Lambda^\alpha$. Next,let $g = f + h$, and let it's Hessian be H_g. We write the polytopic form of H_g as

$$Q_{H_g}(\Lambda) = Q_{H_h}(\Lambda) + Q_{H_f}(\Lambda) = \sum_{|\alpha| \leq d-2} (d'(d'-1)CG_\alpha + B_\alpha)\Lambda^\alpha, \tag{25}$$

where C is a diagonal matrix with $C_{jj} = c_j$.
By theorem 4, we require each $P_\alpha = d'(d'-1)CG_\alpha + B_\alpha \succ 0$. If we require C_α to be diagonally dominant as well, we require

$$c_j d'(d'-1)g'_{\alpha,j} + B_{\alpha,jj} > \sum_i |B_{\alpha,ji}|.$$

From this, we can solve for c_j. For each α, we require

$$c_j \geq \frac{1 - B_{\alpha,ii} + \sum_j |B_{\alpha,ij}|}{(d)(d-1)G_{\alpha,i}}$$

and thus, can choose

$$c_j = \max_\alpha \frac{\max\left\{0, 1 - B_{\alpha,ii} + \sum_j |B_{\alpha,ij}|\right\}}{(d)(d-1)G_{\alpha,i}} \tag{26}$$

□

C ADDITIONAL RESULTS

In this section, we detail additional results that may be of interest to readers, but not included in the main paper. In particular, we focus on additional constructions of dcds.

C.1 SOS-based Construction of Local dcds

In this section, we propose a Sum-of-Squares-based relaxation for the construction of undominated local dcds. In subsequent sections, we will use this result to illustrate the cost of solving the SDPs required is significantly higher than that required for DD-Local.

We first define Sum-of-Squares (SOS) polynomials.

Definition 4. *A polynomial $f \in \mathbb{R}[x]$ is said to be Sum of Squares (SOS) if there exist finitely many polynomials $p_i \in \mathbb{R}[x]$ such that*

$$f(x) = \sum_i (p_i(x))^2$$

We use $\Sigma_{n,d} \subset \mathbf{R}[x]$ to denote the convex cone of polynomials of degree d in n variables which are SOS.

Note that the cone of SOS polynomials is, in general, a strict subset of the cone of nonnegative polynomials, as noted in Parrilo (2000); Lasserre (2001).

Checking whether a polynomial is nonnegative is, in general, strongly NP-hard. However, checking whether a polynomial is SOS can be achieved using semidefinite programming Parrilo (2000). We will leverage this property to derive an SOS based construction of local dcds.

Next, we define semialgebraic sets, which are sets defined by polynomial inequalities and equalities. Note that convex polytopes as defined in (3) are examples of semialgebraic sets.

Definition 5. *A semialgebraic set is a set of the form*

$$S := \{x \in \mathbb{R}^n : f_i(x) \geq 0, i = 1, ..., n_1,\ h_i(x) = 0, i = 1, ..., n_3\}$$

where each $f_i,\ h_i \in \mathbb{R}[x]$.

Next, we state an SOS-based relaxation for constructing local dcds of a polynomial. Our construction relies on Putinar's positivstellensatz Putinar (1993), which certifies nonnegativity of polynomials over compact semialgebraic sets.

Theorem 11. *Suppose $f \in \mathbb{R}[x]$ is of degree d, and let Γ be a polytope defined as in (3), and let $\Sigma_{n,d}$ denote the cone of SOS polynomials in n indeterminates and of degree d. Then, for some $d' \geq d$, there exists $h \in \mathbb{R}[x],\ \sigma^\rho, \sigma_i^\rho \in \Sigma_{n,d-2},\ \sigma_i^h,\ \sigma_i^{f+h} \in \Sigma_{2n,d}$. that solves*

$$\begin{aligned}
\min\ & \rho \\
s.t.\ & \rho - \mathrm{Tr}(H_h(x)) - \sigma_i^\rho(x)\lambda_i(x) = \sigma_0^\rho(x) \\
& y^\top H_h(x) y - \sigma_i^h(x,y)\lambda_i(x,y) = \sigma_0^h(x,y) \\
& y^\top H_{f+h}(x) y - \sigma_i^{f+h}(x,y)\lambda_i(x) = \sigma_0^{f+h}(x,y)
\end{aligned} \tag{SOS}$$

Proof. By Ahmadi and Hall (2017), the existence of an SOS convex dcd is guaranteed. The constraints are guaranteed due to Putinar's positivstellensatz Putinar (1993); Parrilo (2003). □

C.2 Construction of Local dcds with Linear Programming - An Alternative approach

In this section, we state a variant of the optimization problem stated in Theorem 6, which assumes that the coefficients of the Handelman representation of the Hessian of the dcd are diagonally dominant. Note that since we cannot guarantee a representation with diagonally dominant coefficients, we do not state this problem as a theorem.

Suppose $f \in \mathbb{R}[x],\ f(x) = C_f^\top Z(x,d)$ is of degree d, and suppose Γ is a polytope as defined in (3). Then, for some $d' \geq d$, we wish to solve the LP

$$\begin{aligned}
\min_{\mathcal{A}} \ & \rho \\
\text{s.t.} \ & \sum_{2 \leq |\alpha'| \leq d'} C_{h,\alpha'} M_{\alpha,\alpha'} = \sum_{\alpha'' \leq |d'|} (l_\alpha(\Gamma))_{\alpha''} B_{h,\alpha''} \\
& \sum_{2 \leq |\alpha'| \leq d'} (C_{h,\alpha'} + C_{f,\alpha'}) M_{\alpha,\alpha'} = \sum_{\alpha'' \leq |d'|} (l_\alpha(\Gamma))_{\alpha''} B_{f+h,\alpha''} \\
& \rho e_i + \sum_{2 \leq |\alpha'| \leq d'} C_{h,\alpha'} \mathrm{Tr}(M_{\alpha,\alpha'}) = L(\Gamma, d) B \\
& (B_{h,\alpha})_{ii} - \sum_{i \neq j} |(B_{h,\alpha})_{ij}| \geq 0 \qquad (27) \\
& (B_{f+h,\alpha})_{ii} - \sum_{i \neq j} |(B_{f+h,\alpha})_{ij}| \geq 0 \qquad (28) \\
& B \geq 0,
\end{aligned}$$

where

$$\mathcal{A} := \left\{ \rho,\ B,\ C_h,\ \{B_{h,\alpha}\}_{|\alpha| \leq d'}, \{B_{f+h,\alpha}\}_{|\alpha| \leq d'} \right\}$$

with B being an elementwise nonnegative vector. If this problem is feasible, $h(x) = C_h^\top Z(x, d')$ is a dcd of f.

To derive this result, we directly use the construction used in SDP-Local (Theorem 6). The only difference is, we assume that the PSD matrices $B_{h,\alpha}$ and $B_{f+h,\alpha}$ are diagonally dominant, thereby allowing us to enforce the positive-semidefiniteness using (27)-(28). This, therefore, reduces the problem to an LP.

C.3 Construction of local dcds with Diagonally Dominant Hessians without Linear Programming

Previously, we proposed a method to compute a dcd h of f such that $\hat{H}_{f+h}$ has diagonally dominant coefficients. ‘However, it is also possible to construct dcds with diagonally dominant Hessians without needing to solve a conic optimization problem.

We begin by choosing $h(x)$ in the same manner as previously; that is,

$$h(x) = \sum_i c_i (x_i + c + 1)^d$$

for some even d greater than or equal to the degree of f. Similarly, the Hessian of h is given by

$$H_{h,ii}(x) = C_i d'(d' - 1)(x_i + c_i + 1)^{d-2}.$$

Next, we compute the polytopic representation of each element of H_f.

$$q_{f,ij}(\Lambda) = \sum_{|\alpha| \leq d} b_{ij,\alpha} \Lambda^\alpha$$

for each i, j. Next, we define

$$\begin{aligned}
q^+_{f,ij}(\Lambda) &= \sum_{|\alpha| \leq d, b_{ij,\alpha} > 0} b_{ij,\alpha} \Lambda^\alpha = \sum_{|\alpha| \leq d} b^+_{ij,\alpha} \Lambda^\alpha \\
q^-_{f,ij}(\Lambda) &= \sum_{|\alpha| \leq d, b_{ij,\alpha} < 0} b_{ij,\alpha} = - \sum_{|\alpha| \leq d} b^-_{ij,\alpha} \Lambda^\alpha
\end{aligned}$$

We have $b^+_{ij,\alpha} = 0$ when $b_{ij,\alpha} < 0$ and $b^-_{ij,\alpha} = 0$ when $b_{ij,\alpha} > 0$ By theorem 3, $q^+_{f,ij}(\Lambda(x)) > 0$ and $q^-_{f,ij}(\Lambda(x)) < 0$ for all $x \in \Gamma$. Next, note that

$$|H_{f,ij}(x)| = |q^+_{f,ij}(\Lambda(x)) + q^-_{f,ij}(\Lambda(x))| \leq q^+_{f,ij}(\Lambda(x)) - q^-_{f,ij}(\Lambda(x)) \qquad (29)$$

From (29), for each i, we have

$$\sum_{j\neq i} |H_{f,ij}| \leq q^{+}_{f,ij}(\Lambda(x)) - q^{-}_{f,ij}(\Lambda(x)).$$

Next, we consider the Hessian of $f+h$, which is given by

$$H_{f+h,ij}(x) = H_{h,ij}(x) + H_{f,ij}(x).$$

To certify that $H_{f+h}(x)$ is diagonally dominant, for each i, we need

$$P_i(x) = H_{h,ii}(x) + H_{f,ii}(x) - \sum_{j\neq i} |H_{f,ij}(x)| \geq 0 \;\; \forall\, x \in \Gamma$$

However, from (29), it follows that $H_{f+h}(x)$ is diagonally dominant if for each i and all $x \in \Gamma$

$$P_i(x) = H_{h,ii}(x) + H_{f,ii}(x) - \sum_{j\neq i} q^{+}_{f,ij}(\Lambda(x)) - q^{-}_{f,ij}(\Lambda(x)) \geq 0.$$

From Handelman's theorem, this means that the polynomial

$$Q_{P_i}(\Lambda) = \sum_{|\alpha|\leq d} C_\alpha \Lambda^\alpha$$

must have nonnegative coefficients. Recall that the polytopic representation of H_h can be written as

$$Q_{H_h,ii}(\Lambda) = c_i d'(d'-1) \sum_{|\alpha|\leq d-2} G_{i,\alpha}\Lambda,$$

where $G_{i,\alpha} > 0$ for each i, α. Thus, we can choose

$$Q_{P_i}(\Lambda) = Q_{H_h,ii}(\Lambda) + Q_{H_f,ii}(\Lambda) - \left(\sum_{j\neq i} q^{+}_{f,ij}(\Lambda) - q^{-}_{f,ij}(\Lambda)\right) \tag{30}$$

Thus, for each α, we have

$$C_{i,\alpha} = c_i d'(d'-1) G_{i,\alpha} + b_{ii,\alpha} - \left(\sum_{j\neq i} b^{+}_{ij,\alpha} + b^{-}_{ij,\alpha}\right) \geq 0$$

If $H_{f+h}(x)$ is diagonally dominant, we require each $C_{i,\alpha} \geq 0$. Thus,

$$c_i \geq \frac{1}{d'(d'-1)G_{i,\alpha}} \left(-b_{ii,\alpha} + \sum_{j\neq i} b^{+}_{ij,\alpha} + b^{-}_{ij,\alpha}\right) \text{ for all } \alpha.$$

Thus, we can find

$$c_i = \max_{\alpha} \frac{\max\{-b_{ii,\alpha} + \sum_{j\neq i} b^{+}_{ij,\alpha} + b^{-}_{ij,\alpha} b_{i,\alpha}, 0\}}{d'(d'-1)G_{i,\alpha}} \tag{31}$$

C.4 Computation of k_{ji} Coefficients

In this section, we discuss the construction of k_{ji} coefficients as described in Lemma 5.

Suppose we have a polytope $\Gamma := \{x \in \mathbb{R}^n : a_i^\top x + b_i \geq 0\}$. Recall that the k_{ji} coefficients introduced in 5 satisfy $k_{ji} a_i = e_j$ where e_j is the jth coordinate vector. We can find k_{ji} for a given polytope by solving the following linear program:

$$k_j = \arg\min_{k} \{1 | A^\top k \geq e_j - \varepsilon A^\top \mathbf{1}\}$$

where ε is any positive constant, and $\mathbf{1}$ is a vector of ones. For certain sets, however, k_{ji} can be found algorithmically. For a simplex

$$\Gamma_{\text{simplex}} := \{x \in \mathbb{R}^n : l_0(x) \geq 0, l_j(x) \geq 0, j \in [n]\},$$

where $l_0(x) = 1 - \sum_i x_i$ and $l_i(x) = x_i$, we can find $k_{ji} = 1$ for $i \neq j$, $i = 0, 1, \cdots, n$, and $k_{ji} = 2$ otherwise. Similarly, for a hypercube

$$\Gamma_{\text{cube}} := \{x \in \mathbb{R}^n : l_j^+(x) \geq 0, l_j^-(x) \geq 0, j \in [n]\},$$

where $l_i^+(x) = x_i + u_i$ and $l_i^- = v_i - x_i$, where $u_i < v_i$ are real numbers, we can find $k_{ji} = 1$ for $i \neq j$ and $k_{ji} = 2$ for $i = j$.

D ADDITIONAL IMPLEMENTATION DETAILS

In this section, we provide provide additional experimental details from section 6, as well as a simple example highlighting the construction of dcds.

D.1 Experimental Setup

We conduct our experiments on a system with the following specifications:

- 32GB RAM
- intel17-11700
- MATLAB2020b

We use the Mozart and Penny datasets as used in Wang et al. (2014); Ecker and Jepson (2010) for shape-from-shading experiments.

D.2 Optimization Details

We use projected gradient descent to solve the inner loop of the CCCP. Given a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we aim to solve

$$x^* = \arg\min_{X \in \Gamma} f(x).$$

Here, we consider Γ to be a polytope as defined in (3).

We use the following update rule:

$$\begin{aligned} y_{k+1} &= x_k - \eta \nabla f(x_k) \\ x_{k+1} &= P_\Gamma(y_{k+1}), \end{aligned} \tag{32}$$

where $P_\Gamma(y)$ is the projection of y onto Γ. In particular, for the set $\Gamma := \{x \in \mathbb{R}^n : \ a \leq x_i \leq b\}$, if $x = P_\Gamma(y)$, then

$$x_i = \min\{b, \max\{a, x_i\}\}.$$

We typically use $\eta = 0.05$ as a stepsize.

D.3 Undominated vs Suboptimal dcds

In this section, we provide a simple experiment that provides intuition into the difference between undominated and suboptimal dcds. We also highlight the utility of undominated dcds when it comes to the CCCP.

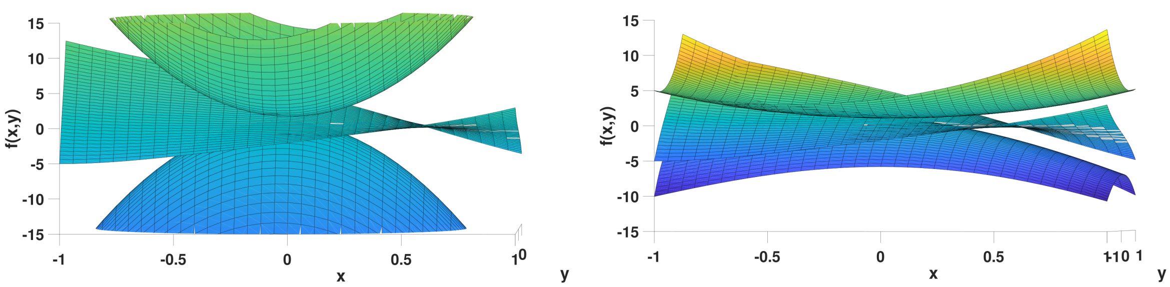
We consider the problem of finding dcds of

$$f(x, y) = x^4 - xy^3 + x^2y - 5xy - x^2 - 2x + 3y + 1$$

over the polytope $\Gamma := \{x \in \mathbb{R}^2 : -1 \leq x_i \leq 1, \ i = 1, 2\}$. We apply a variety of methods to construct dcds, and compare the efficacy of the convex-concave procedure at solving $x^* = \arg\min\{f(x) : x \in \Gamma\}$ having used the different dcds we obtained, and using a projected gradient descent algorithm to solve the convex subproblems. We obtain dcds by (a) solving Problem 1 with the relaxation given by Theorem 8, (b) solving the feasibility problem stated in Theorem 6 giving us, (c) solving (26). We compare the values of $\max_{x \in \Gamma} \operatorname{Tr}(H_h(x))$, as well as the number of iterations for Convex-Concave procedure. We state the experimental details in section D in the Appendix. We present our results in Table 5, and plot the dcds obtained by solving optimization problems in Figure 1.

Method	$\max_{x \in \Gamma} \operatorname{Tr}(H_h(x))$	CCCP Iterations
Thm. 8 Opt.	**26.7**	**13**
Thm. 6 Feas.	70	31
Thm. 9	< 81.3	89

Table 5: Comparison of efficacy of methods for construction of dcds for a polynomial.Thm. 8 Opt. refers to solving Problem 1 using the conditions in Theorem 8, and Thm 6 Feas. refers to finding $h(x)$ that satisfies (18)-(20).



(a) Plot of $f + h_s$, $f(x, y)$, and $-h_s$.

(b) Plot of $f + h_u$, f, and $-h_u$.

Figure 1: Plots of different dcds of $f(x, y)$; $f(x, y)$ is the surface "sandwiched" between the convex and concave functions. We find $h_s = 23x^2 + 12y^2 + 2x - 3y$ and $h_u(x, y) = .52x^4 - .28x^3y + .42x^2y^2 + .5xy^3 + .07y^4 + .06x^3 + .83x^2y + .17xy^2 - .06y^3 + 5.5x^2 + 1.67xy + 2.67y^2 + 2x - 3y$

D.4 Additional Details on Shape-from-Shading Experiments

In this section, we provide additional details for the shape-from-shading experiments given in Section 6. We choose a random initialization for the CCCP.

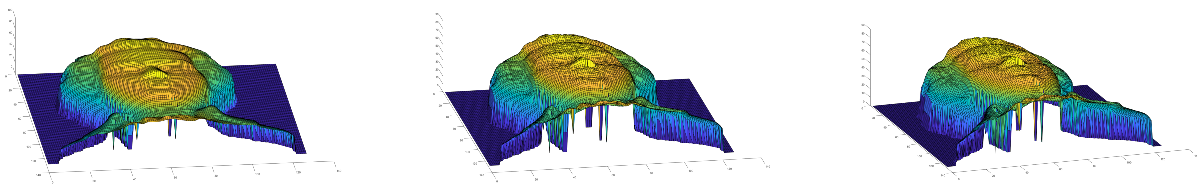
In Figure 2, we compare the 3D-structures with the ground truth structure obtained. We find that we obtain solutions that are close to the ground truth. We see that the quality of the depth maps is high, though imperfections can be noticed. The quality of the shape is roughly the same irrespective of the choice of dcd used.

E ADDITIONAL DETAILS ON POLYNOMIAL SHAPE FROM SHADING

In this section, we provide additional discussion on the polynomial shape-from-shading (SFS) problems. We focus on the fundamental computer-vision aspects, as how such problems can be solved using graphical model inference, as shown in Ecker and Jepson (2010); Wang et al. (2014).

In SFS, we are given a 2D image of a 3D object. We assume that both the image and object have the same support, and we assume knowledge of the camera and the light source. Under the Lambertian lighting assumption (used in this model), we further assume that the image intensity at each point is proportional to the angle between surface normals of that point, and the light source direction Wang et al. (2014); Ecker and Jepson (2010); Salzmann (2013).

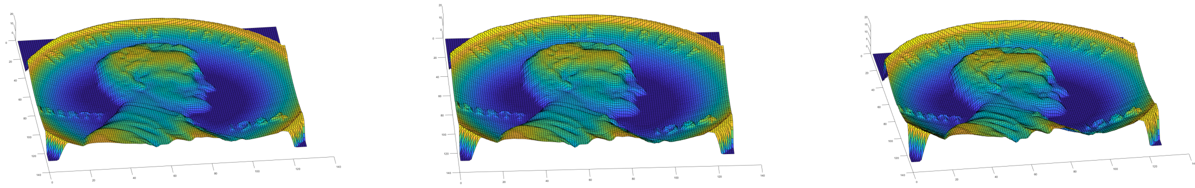
In Ecker and Jepson (2010); Salzmann (2013); Wang et al. (2014); Khamaru and Wainwright (2018), this problem is formulated as finding the minimizer of the sum of 3-variable polynomials of degree 4. As noted in Khamaru and Wainwright (2018), these polynomials are non-convex, but bounded from below and coercive. In this work, we apply box constraints to the SFS problem - that is, the depth cannot be a negative number, and the depth must also be bounded from above. More specifically, we let $V_{ij} = (x_{ij}, y_{ij}, z_{ij})$ be the 3D coordinates of the i, jth grid point. Our goal is to find z_{ij}. Next, let r be a clique of 3 neighboring points on the grid that form a triangular section of the mesh (i.e. $r_{ij} = (V_{ij}, V_{i+1,j}, V_{i,j+1})$). Under the Lambertian lighting model, I_r is the observed



(a) Ground Truth for Mozart

(b) Depth Map for Mozart after CCCP with DD-Local

(c) Depth Map for Mozart after CCCP with DD-Linear



(d) Ground Truth Depth Map for Penny

(e) Depth Map for Penny after CCCP with DD-Local

(f) Depth Map for Penny after CCCP with DD-Linear

Figure 2: Depth Maps for the Mozart/Penny datasets obtained via CCCP using DD-Local and DD-Linear.

intensity, and $L = (l_1, l_2, l_3)$ is the light direction. The surface normal is the vector $N_r = (p_r, q_r, 1)$, where p_r and q_r are functions of $V_{ij}, V_{i+1,j}, V_{i,j+1}$, and, in particular, are affine functions of the z-coordinates.Thus, we can write $N_r = N_r(z_r) = N_r(z) = (p_r(z), q_r(z), 1)$. Then, as shown in Salzmann (2013); Wang et al. (2014); Khamaru and Wainwright (2018), the SFS problem can be written as

$$z^* = \arg\min_{z} \sum_{r \in \mathcal{R}} \left((l_1 p_r(z) + l_2 q_r(z) + l_3)^2 + I_r^2 \left((p_r(z)^2 + q_r(z)^2 + 1 \right) \right)^2 .$$

Furthermore, we have that each $r \in \mathbb{R}$ is a tuple of the form $r = ((i,j), (i, j+1), (i+1, j))$. With this, we have we have

$$p_r(z) = \frac{(y_{i,j+1} - y_{i,j})(z_{i+1,j} - z_{i,j}) - (y_{i+1,j} - y_{i,j})(z_{i,j+1} - z_{i,j})}{(x_{i,j+1} - x_{i,j})(y_{i+1,j} - y_{i,j}) - (x_{i+1,j} - x_{i,j})(y_{i,j+1} - y_{i,j})}, \tag{33}$$

$$q_r(z) = \frac{(x_{i,j+1} - x_{i,j})(z_{i+1,j} - z_{i,j}) - (x_{i+1,j} - x_{i,j})(z_{i,j+1} - z_{i,j})}{(x_{i,j+1} - x_{i,j})(y_{i+1,j} - y_{i,j}) - (x_{i+1,j} - x_{i,j})(y_{i,j+1} - y_{i,j})}. \tag{34}$$

F COMPLEXITY ANALYSIS

The computational challenge faced when searching for DC Decompositions (or indeed, solving any polynomial optimization problem) is the size of the associated LP or SDP. In most relaxations for polynomial optimization problems (POPs), the decision variables in the optimization problem are the coefficients of the original polynomial cost function and constraints. In this section, we provide a proof for Theorem 10, and thus compare the computational complexities of obtaining dcds for some $f \in \mathbb{R}[x]$ over a polytope Γ as defined in (3). For clarity, we derive the complexity of each method separately, as well as that of the SOS-based method derived in Theorem 11 stated in Appendix C .

F.1 Complexity of Obtaining a DC Decomposition using Sum-of-Squares Programming

We consider the problem of finding a dcd of $f \in \mathbb{R}[x]$ using Theorem 11. We now state the complexity of solving (SOS).

Lemma 6. *Suppose $f \in \mathbb{R}[x]$ is of degree d, and we wish to find a local dcd of f over a polytope Γ as defined in* (3) *with K facets. The complexity of solving* (SOS) *up to accuracy ε with SOS multipliers σ_i^h and σ_i^{f+h}, for all i, are of degree $d-2$, and where σ_o^h and σ_o^{f+h} are of degree d, using interior point algorithms is $\tilde{O}(K^3 n^{7d})$, where $\tilde{O}$ is used to suppress* $\mathrm{poly}(1/\varepsilon)$ *factors.*

Proof. Solving an SDP with Θ_{SOS} variables and Ξ_{SOS} constraints with an interior point algorithm requires $\tilde{O}(\Xi_{SOS}^3 + \Theta_{SOS}^3\Xi_{SOS} + \Theta_{SOS}^2\Xi_{SOS}^2)$ Nemirovski (2004). We have

$$C_h \in \mathbb{R}^{\binom{n+d}{d}},\ \sigma_0^h \in \Sigma_{2n,d},\ \sigma_i^h \in \Sigma_{2n,d-2}.$$

From this, the number of elements of the Gram matrix associated with the SOS form is

$$O\binom{2n+d/2}{d/2}^2,$$

and the total number of SOS polynomials of the same size is $2K$, which are of size

$$O\binom{n+(d-2)/2}{(d-2)/2}^2.$$

Thus,

$$\Theta_{SOS} = 2K\binom{2n+(d-2)/2}{(d-2)/2}^2 + 2\binom{2n+(d/2)/2}{(d/2)}^2 = O(Kn^{2d}).$$

Similarly, we get $\Xi_{SOS} = O((2n)^d)$. Applying this to the complexity, we get

$$C = (\tilde{O}(2n^d))^3 + (\tilde{O}(2Kn^{2d})^3\tilde{O}(2n^d)) + (\tilde{O}(2Kn^{2d})^2\tilde{O}(2n^d))^2 = \tilde{O}(K^3n^{7d})$$

□

F.2 Complexity of solving SDP-Local

In this section, we analyze the time complexity of finding a local dcd using SDP-Local.

Lemma 7. *Suppose $f \in \mathbb{R}[x]$ is of degree d, where d is even, and suppose we want to find a local dcd $h \in \mathbb{R}[x]$ of degree d. Then, the complexity of solving the SDP in Theorem 6 to an accuracy ε is $\tilde{O}(K^{4d})$, where $\tilde{O}(\cdot)$ is used to suppress* poly$(1/\varepsilon)$ *factors.*

Proof. Solving an SDP with Θ variables and Ξ constraints with an interior point algorithm requires $\tilde{O}(\Xi^3 + \Theta^3\Xi + \Theta^2\Xi^2)$ Nemirovski (2004). Recall that $C_h \in \mathbb{R}^{\binom{n+d}{d}}$, and each $B_{h,\alpha},\ B_{f+h,\alpha} \in \mathbb{S}^n$. Thus, for each α, we have $O(2n^2)$ variables, and we have $2\binom{K+d-2}{d-2} = O(K^{d-2})$ such variables. Thus, we have

$$\Theta = 2O(n^2)O(K^{d-2}) + O(n^d) = O(n^2K^{d-2} + O(n^d)) = O(K^d)$$

where the last equality holds since $K > n, O(n^d) = O(K^d)$. Similarly, we $H_h(x)$ has $\binom{n+d-2}{d-2} = O(n^{d-2})$ coefficients, and each coefficients, and each coefficient has $O(n^2)$ elements. Thus,

$$\Xi = O(n^d).$$

From this, we get

$$C = \tilde{O}((O(n^3d) + O(K^{3d})O(n^d) + O(n^2d)O(K^{2d})) = \tilde{O}(K^{4d}).$$

□

F.3 Complexity of Solving DD-Local

Next, we analyze the complexity of constructing a local dcd using DD-Local. We state the result below.

Lemma 8. *Suppose $f \in \mathbb{R}[x]$ is of degree d, where d is even, and suppose we want to find a local dcd $h \in \mathbb{R}[x]$ of degree d. Then, the complexity of solving the LP in Theorem 8 to an accuracy ε is $\tilde{O}(K^{2d+2})$, where $\tilde{O}(\cdot)$ is used to suppress* poly$(1/\varepsilon)$ *factors.*

Proof. Recall that solving an LP with Θ variables and Ξ constraints with an interior point algorithm requires

$$\tilde{O}(\Theta^2\Xi)$$

operations Nemirovski (2004).

First, we count the number of variables. $C_h \in \mathbb{R}^{\binom{n+d}{d}} = O(n^d)$, $D_{h,ij} \in \mathbb{R}^{\binom{n+d-2}{d-2}} = O^(n^{d-2})$; there are $2n$ such variables. Next, each $B_{h,ij,+}$, $B_{h,ij,-}$ has $\binom{K+d-2}{d-2} = O(K^{d-2})$ variables, of which there are $O(n)$. Lastly, each $B_{h,ii}$ has $\binom{K+d-2}{d-2}$ elements, and there are $2n$ such variables. Thus,

$$\Theta = O(2nK^{d-2}) + O(2n^2K^{d-2}) + O(2nK^d - 2) = O(K^d).$$

Similarly, there are $O(n) = O(K)$ constraints. Thus, we have

$$C = \tilde{O}(O(K^{2d})O(K^2)) = \tilde{O}(K^{2d+1})$$

□

Lemma 8 clearly shows the improvement in complexity when using DD-Local as compared to SDP-Local and Theorem 11.

F.4 Complexity of Solving DD-Linear

Last, we analyze the complexity of constructing a local dcd using DD-Linear.

Lemma 9. *Suppose $f \in \mathbb{R}[x]$ is of degree d, where d is even, and suppose we want to find a local dcd $h \in \mathbb{R}[x]$ of degree d'. Then, the complexity of constructing a dcd using Theorem 9 to an accuracy ε is $\tilde{O}(nK^{d'-2}$.*

Proof. We assume we can preprocess the Hessian of f and the Handelman representation of h as given in Theorem 9. Then, for each of the n variables, we have $O(K^{d'-2})$ monomials. Finding the max of these values (to obtain the max required in (Lin)) costs $O(nK^{d'-2})$. □