dOPT: DIFFERENTIATING CONIC OPTIMIZATION VIA GEOMETRIC REDUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimization layers enable the incorporation of structured constraints and decision problems into learning systems. Training such systems requires differentiating through the embedded optimization problem, which can be challenging for general conic programs. We introduce **dOPT**, a solver-agnostic framework that, rather than differentiating the full conic formulation, reduces it at a computed primal-dual solution to an equality-constrained quadratic program that preserves the reference solution and its first-order sensitivity. The reduction captures the local first- and second-order conic geometry relevant to differentiation and remains well defined at singular configurations. Computing solution derivatives then requires a single symmetric linear solve, independently of the forward solver. We derive explicit reductions for convex NLPs, QPs, SOCPs, and SDPs. Numerical experiments validate the computed gradients and show favorable backward-pass scalability, with substantial speedups over existing differentiable conic optimization methods as problem size increases.

1 INTRODUCTION

Optimization layers incorporate mathematical structure into learning models by defining their outputs through parameterized optimization problems (Amos & Kolter, 2017; Agrawal et al., 2019a). They have been used across control, decision-making, robotics, imaging, and other structured learning tasks. Training such models requires differentiating the optimal solution with respect to problem parameters, *i.e.*, characterizing its local sensitivity to parameter perturbations, making the backward pass central to the practical use of optimization layers.

A standard approach differentiates the optimality conditions of the original problem. OptNet differentiates the KKT system for quadratic programs (QPs) (Amos & Kolter, 2017), while DiffCP and CVXPYLayers extend implicit differentiation to conic programs (Agrawal et al., 2019b;a). However, the structure needed to *solve* a problem often exceeds what is needed to characterize the *local sensitivity* of its solution. Exploiting this distinction, dQP (Magoon et al., 2025) constructs locally first-order-equivalent equality-constrained problems for QPs, BPQP (Pan et al., 2024) formulates an equality-constrained backward QP for smooth convex programs, including certain SOCPs, and FFO-Layer (Zhao et al., 2025) approximates gradients via finite differences on an equality-constrained surrogate. These methods use a solution computed by an arbitrary forward solver to construct a simpler proxy problem from which derivatives are computed.

This suggests a broader reduction principle: retain only the local constraint geometry that affects first-order sensitivity at the solution. Extending this principle beyond linearly constrained problems, however, is nontrivial. For linear inequality constraints, the relevant local geometry can be captured by a carefully selected subset of active constraints. More general, nonpolyhedral cones have curved boundaries whose normals vary under local perturbations of the solution. Simply replacing the cone locally by its tangent space can therefore be insufficient: changes in the active normal direction introduce a curvature term that is essential for recovering the correct solution derivative. The challenge extends to singular boundary configurations where a smooth local description may fail, including second-order cone apexes and PSD matrices with multiple zero eigenvalues.

We introduce **dOPT**, a solver-agnostic geometric framework that constructs an explicit, first-order equivalent equality-constrained QP from a primal–dual solution and the local conic geometry. Building on classical sensitivity analysis (Fiacco, 1983; Robinson, 1980; Bonnans & Shapiro, 2000),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

dOPT identifies the perturbation directions relevant to the derivative of the solution map through the *critical cone*, while accounting for variation of the cone normal along those directions. Under standard regularity conditions, these effects are encoded by two matrices: C^*, whose nullspace is the critical cone, and H^*, which captures the corresponding normal variation. This construction naturally accommodates the singular configurations described above.

Together, C^* and H^* define the constraints and curvature correction of the locally equivalent reduced formulation, which preserves both the reference solution and its parameter derivative. Differentiating its KKT conditions yields a single symmetric linear system, and the construction requires only the problem data and a computed primal–dual solution. The derivative can therefore be efficiently evaluated independently of how the forward solution is obtained, allowing mature black-box solvers such as Gurobi or MOSEK to be used without differentiating through their algorithms.

Our main contributions are:

- We derive a first-order equivalent reduction for convex conic programs that preserves both the reference solution and its sensitivity, while naturally accommodating nonsmooth conic geometry.
- We provide explicit critical-cone and curvature constructions for NLPs, QPs, SOCPs, and SDPs, yielding reduced equality-constrained QPs from which solution derivatives can be efficiently computed with a single symmetric linear solve. In particular, the derivations accommodate singular configurations such as SOC apexes and PSD solutions with multiple zero eigenvalues.
- We provide a straightforward, solver-agnostic implementation of dOPT, available at `https://github.com/dopt-anon/dOPT`, with full flexibility in the choice of forward solver.
- We evaluate our implementation on synthetic SOCPs and SDPs and an SDP-based learning task, validating gradient accuracy on the synthetic problems and demonstrating favorable backward-pass scalability and substantial speedups over existing differentiable conic optimization methods.

2 Related Work

Differentiable optimization has diverse applications, including control (Amos et al., 2018), differentiable physics (de Avila Belbute-Peres et al., 2018), and certifiable robotics (Holmes et al., 2024). A similarly broad range of differentiation methods has been developed, including modular implicit differentiation (Blondel et al., 2022; Besançon et al., 2024), conic differentiation (Agrawal et al., 2019b;a; Healey et al., 2026), alternating and augmented-Lagrangian methods (Sun et al., 2022; Butler & Kwon, 2023; Butler, 2023; Bambade et al., 2024), solver-iteration differentiation (Oshin et al., 2026; Liu & Hu, 2026), and penalty or perturbation methods (Linghu et al., 2026; Paulus et al., 2024). Extensions address nonconvexity (Rosemberg et al., 2025), GPU computation (Barratt et al., 2026), and code generation (Schaller & Boyd, 2025). Our focus is sensitivity analysis and local reductions of the backward problem, which are most closely related to our approach.

Classical sensitivity analysis characterizes solution derivatives through constraint qualifications, critical cones, and second-order geometry (Fiacco, 1983; Robinson, 1980; Bonnans & Shapiro, 2000; Bonnans et al., 1999; Mohammadi et al., 2021), including nonpolyhedral curvature (Bonnans et al., 1998) and SOCP/SDP sensitivity (Bonnans & Ramírez C, 2005; Shapiro & Fan, 1997; Alizadeh et al., 1997; Sun, 2006). Related work addresses partial smoothness and stratification (Lewis, 2002; Bao et al., 2026), weaker regularity (Andreani et al., 2023; 2024; Fukuda et al., 2023), nonsmooth differentiation and nonunique solutions (Bolte et al., 2021; 2026), composite bilevel problems (Solla & Royset, 2026), and directional influence (Wang et al., 2026). KKT-based implicit differentiation connects this theory to optimization layers (Barratt, 2018); see Pacaud (2025) for a recent survey.

Several methods simplify the backward problem through local reductions. BPQP (Pan et al., 2024) formulates an equality-constrained backward QP for smooth convex programs, including certain SOCPs, while dQP (Magoon et al., 2025) constructs a first-order-equivalent equality-constrained QP through active-set reduction. FFOLayer (Zhao et al., 2025) constructs an equality-constrained surrogate with linearized active constraints and a Lagrangian objective, then approximates hypergradients via finite differences using perturbed surrogate solves. Related reductions appear in SOCP trajectory sensitivity (Xu et al., 2025) and tangent-space methods for decision-focused learning (Lee et al., 2026). dOPT extends this viewpoint to general convex conic programs, using critical-cone geometry and curvature to construct a reduced QP whose KKT system yields solution derivatives through a single symmetric linear solve.

3 PRELIMINARIES

In this paper, we consider the *parametric convex conic program*

$$\begin{aligned} z^*(\theta) = \arg\min_{z} \quad & f_\theta(z) \\ \text{s.t.} \quad & G_\theta(z) \in \mathcal{K} \\ & h_\theta(z) = 0, \end{aligned} \tag{P1}$$

where $z \in \mathbb{R}^n$ is the optimization variable and $\theta \in \mathbb{R}^s$ is the problem parameter. The *solution map* $z^*(\theta) : \mathbb{R}^s \to \mathbb{R}^n$ maps parameters to optimal solutions. The parametric objective and constraint functions $f_\theta(z) := f(z, \theta) : \mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}$, $G_\theta(z) := G(z, \theta) : \mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}^m$ and $h_\theta(z) := h(z, \theta) : \mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}^l$ and their Jacobians with respect to z are continuously differentiable jointly in (z, θ). For fixed θ, $f_\theta(z)$ is convex, $h_\theta(z)$ is affine and the set $\{z : G_\theta(z) \in \mathcal{K}\}$ is convex, where $\mathcal{K}$ is a closed convex cone. The parameter θ encodes perturbable problem data, and we study the response (sensitivity) of $z^*(\theta)$ to variations in this data. For example, in a quadratic program with $f_\theta(z) = \frac{1}{2} z^T P z + q^T z$, θ may encode (P, q), and the corresponding sensitivities include derivatives such as dz^*/dP and dz^*/dq.

3.1 THE KKT OPTIMALITY CONDITIONS AND THE LAGRANGIAN

The first-order Karush–Kuhn–Tucker (KKT) conditions (Karush, 1939; Kuhn & Tucker, 1951; Boyd & Vandenberghe, 2004; Nocedal & Wright, 2006) characterize the optimal points of the convex constrained optimization problems, which are the key to analyzing the sensitivity of the optimal solution with respect to the parameters. The KKT conditions for the parametric convex conic program P1 are

$$\nabla_z \mathcal{L}_\theta(z^*(\theta), \mu^*(\theta), \nu^*(\theta)) = 0 \tag{2a}$$
$$h_\theta(z^*(\theta)) = 0 \tag{2b}$$
$$y^*(\theta) := G_\theta(z^*(\theta)) \in \mathcal{K} \tag{2c}$$
$$\mu^*(\theta) \in \mathcal{K}^* \tag{2d}$$
$$\langle \mu^*(\theta), y^*(\theta) \rangle = 0 \tag{2e}$$

where $\mu^*(\theta)$ and $\nu^*(\theta)$ are the *optimal dual variables* associated with the conic and affine constraints, respectively, stationarity Equation 2a is expressed in terms of the *Lagrangian* $\mathcal{L}_\theta(z, \mu, \nu) = f_\theta(z) - \mu^T G_\theta(z) - \nu^T h_\theta(z)$, and $\mathcal{K}^* = \{\mu : \langle \mu, y \rangle \geq 0, \forall y \in \mathcal{K}\}$ is the *dual cone* of $\mathcal{K}$.

3.2 CONE GEOMETRY

To analyze how the solution of Problem P1 responds to changes in the problem parameters, we characterize the local cone geometry near a primal–dual solution. The *tangent cone* describes feasible first-order directions at a point $y \in \mathcal{K}$, while the *normal cone* describes the corresponding supporting directions:

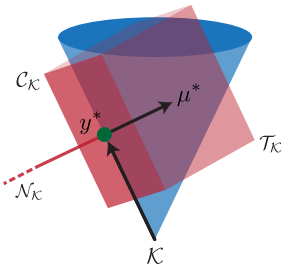
$$\mathcal{T}_\mathcal{K}(y) := \text{cl}\{\alpha(u - y) : \alpha \geq 0, u \in \mathcal{K}\}, \qquad \mathcal{N}_\mathcal{K}(y) := \{w : \langle w, u - y \rangle \leq 0, \forall u \in \mathcal{K}\}.$$

The same notions apply to the dual cone $\mathcal{K}^*$, giving $\mathcal{T}_{\mathcal{K}^*}(\mu)$ and $\mathcal{N}_{\mathcal{K}^*}(\mu)$ for $\mu \in \mathcal{K}^*$. Together, these cones characterize the feasible first-order primal–dual perturbation directions (Rockafellar, 1970; Bonnans & Shapiro, 2000; Boyd & Vandenberghe, 2004).

The key geometric object governing solution sensitivity is the *critical cone* associated with a complementary primal–dual pair $y^* \in \mathcal{K}$ and $\mu^* \in \mathcal{K}^*$,

$$\mathcal{C}_\mathcal{K}(y^*) = \mathcal{T}_\mathcal{K}(y^*) \cap \mu^{*\perp}, \qquad \mathcal{C}_{\mathcal{K}^*}(\mu^*) = \mathcal{T}_{\mathcal{K}^*}(\mu^*) \cap y^{*\perp},$$

which is determined entirely by the local first-order cone geometry (Bonnans & Shapiro, 2000, Proposition 3.10, p. 152). The inset illustrates a complementary pair and its associated cones. While the tangent cones encode first-order feasibility, the critical cones further restrict perturbations through the linearized complementarity condition Equation 2e. Intuitively, this restricts attention to the perturbation directions relevant to sensitivity analysis and along which cone curvature must be accounted for. These are cone-space objects; linearized equality constraints are imposed separately for the full problem.



3.3 Regularity Assumptions and Notation

Our analysis relies on standard assumptions from sensitivity analysis. Strict complementarity holds at a complementary primal–dual pair (y^*, μ^*) if $\mu^* \in \operatorname{relint}(-\mathcal{N}_{\mathcal{K}}(y^*))$; geometrically, it provides a clear separation between active and inactive cone geometry and supports its stable identification under small perturbations. We additionally assume non-degeneracy, the second-order sufficient condition and $\mathcal{C}^2$-cone reducibility whose definitions are deferred to Appendix A. Under non-degeneracy and strict complementarity, the critical cone is in fact a linear subspace, $\mathcal{C}_{\mathcal{K}}(y^*) = \operatorname{aff}(\mathcal{N}_{\mathcal{K}}(y^*))^{\perp}$ (Bonnans & Shapiro, 2000, Proposition 4.73, p. 317). This property is central to constructing a reduced problem that preserves the first-order behavior of the original problem.

Throughout, (θ) denotes dependence of a solution map on the parameter θ, whereas the subscript θ denotes a function parameterized by θ. Let $(z^*(\theta), \mu^*(\theta), \nu^*(\theta))$ denote the primal–dual solution map. Fix a reference parameter θ_0 and define the reference solution $(z^*, \mu^*, \nu^*) := (z^*(\theta_0), \mu^*(\theta_0), \nu^*(\theta_0))$. For gradients, Jacobians and Hessians evaluated at the reference solution, we suppress the solution arguments while retaining the parameter subscript; for example, the Hessian of the Lagrangian $\nabla^2_{zz}\mathcal{L}_\theta := \nabla^2_{zz}\mathcal{L}_\theta(z^*, \mu^*, \nu^*)$. Evaluation is indicated explicitly by the subscript; for example, $\nabla_z G_\theta := \nabla_z G_\theta(z^*)$, and $\nabla_z G_{\theta_0} := \nabla_z G_\theta(z^*)|_{\theta=\theta_0}$. For a direction $\mathrm{d}\theta \in \mathbb{R}^s$, denote the directional differentials at θ_0 by

$$\mathrm{d}z^* := Dz^*(\theta_0)[\mathrm{d}\theta], \qquad \mathrm{d}\mu^* := D\mu^*(\theta_0)[\mathrm{d}\theta], \qquad \mathrm{d}\nu^* := D\nu^*(\theta_0)[\mathrm{d}\theta].$$

Similarly, define

$$y^*(\theta) := G_\theta(z^*(\theta)), \qquad y^* := G_{\theta_0}(z^*), \qquad \mathrm{d}y^* := Dy^*(\theta_0)[\mathrm{d}\theta].$$

4 dOPT: First-Order Equivalent Reduction

This section presents our main contribution: a reduced problem that is first-order equivalent to the conic problem P1 at the optimal solution (z^*, μ^*, ν^*), yielding the same local solution and the same derivative with respect to the parameters. The central observation is that only the active cone geometries at the solution influence its first-order sensitivity. By restricting attention to these active directions and eliminating those that do not contribute to the gradient, we obtain a simplified problem. The resulting reduced formulation is a quadratic program with only equality constraints, capturing precisely the degrees of freedom relevant for differentiation. As a result, the gradient can be computed efficiently by solving a single linear system. This framework fully decouples optimization and differentiation: one may use any black-box solver to compute the solution of the original problem, and subsequently differentiate it solely through the reduced problem.

Under the nondegeneracy, strict complementarity, second-order sufficient conditions and $\mathcal{C}^2$-cone reducibility at the optimal solution (z^*, μ^*, ν^*) of Problem P1, the following theorem constructs a reduced problem that is first-order equivalent to the original conic problem. A proof is provided in Appendix B.

Theorem 4.1. *The conic Problem P1 is first-order equivalent at θ_0 to*

$$\begin{aligned} \hat{z}(\theta) = \arg\min_{z} \quad & \nabla_z f_\theta^T \bar{z} + \frac{1}{2}\bar{z}^T \left(\nabla^2_{zz}\mathcal{L}_\theta - \nabla_z G_\theta^T H^* \nabla_z G_\theta\right) \bar{z} \\ \text{s.t.} \quad & C_\theta^* \left(G_\theta(z^*) - G_{\theta_0}(z^*) + \nabla_z G_\theta \bar{z}\right) = 0 \\ & h_\theta(z^*) + \nabla_z h_\theta \bar{z} = 0 \end{aligned} \tag{P2}$$

where $\bar{z} := z - z^*$. The matrices C_θ^* and H^* form a compatible pair: C_θ^* varies smoothly, has full row rank, and satisfies $\ker C^* = \mathcal{C}_{\mathcal{K}}(y^*)$, where $C^* := C^*_{\theta_0}$. The symmetric curvature matrix H^* determines the dual variation up to a normal residual: $\mathrm{d}\mu^* - H^*\mathrm{d}y^* \in \mathcal{C}_{\mathcal{K}}(y^*)^{\perp}$ for every solution differential $(\mathrm{d}y^*, \mathrm{d}\mu^*)$.

At θ_0, the primal–dual solution map $(\hat{z}(\theta), \hat{\mu}(\theta), \hat{\nu}(\theta))$ of the reduced Problem P2 is related to the primal–dual solution map of the original problem by

$$z^* = \hat{z}(\theta_0), \qquad \mu^* = C^{*T}\hat{\mu}(\theta_0), \qquad \nu^* = \hat{\nu}(\theta_0), \tag{4}$$

and their directional differentials satisfy, for every parameter direction $\mathrm{d}\theta$*,*

$$\mathrm{d}z^* = \mathrm{d}\hat{z}, \qquad \mathrm{d}\mu^* = H^*\mathrm{d}y^* + C^{*T}\mathrm{d}\hat{\mu}, \qquad \mathrm{d}\nu^* = \mathrm{d}\hat{\nu}. \tag{5}$$

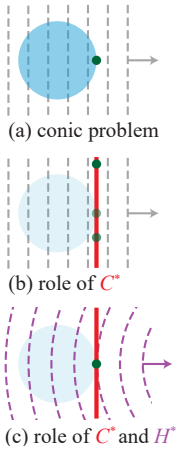
Consequently, the differential $(\mathrm{d}\hat{z}, \mathrm{d}\hat{\mu}, \mathrm{d}\hat{\nu})$ *is obtained by solving the linear system*

$$\begin{bmatrix} Q_{\theta_0} & -\nabla_z G_{\theta_0}^T C^{*T} & -\nabla_z h_{\theta_0}^T \\ -C^*\nabla_z G_{\theta_0} & 0 & 0 \\ -\nabla_z h_{\theta_0} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}\hat{z} \\ \mathrm{d}\hat{\mu} \\ \mathrm{d}\hat{\nu} \end{bmatrix} = \begin{bmatrix} -\rho_{\theta_0} \\ C^*\nabla_\theta G_{\theta_0} \\ \nabla_\theta h_{\theta_0} \end{bmatrix} \mathrm{d}\theta, \tag{6}$$

where

$$Q_{\theta_0} := \nabla_{zz}^2 \mathcal{L}_{\theta_0} - \nabla_z G_{\theta_0}^T H^* \nabla_z G_{\theta_0},$$
$$\rho_{\theta_0} := \nabla_{z\theta}^2 \mathcal{L}_{\theta_0} - \nabla_z G_{\theta_0}^T H^* \nabla_\theta G_{\theta_0}.$$

Theorem 4.1 yields a solver-agnostic procedure for computing solution derivatives. Given a primal–dual solution (z^*, μ^*, ν^*) obtained from any forward solver, the matrices C^* and H^* are constructed from the local cone geometry at y^*. These characterize the critical cone $\mathcal{C}_\mathcal{K}(y^*)$ and the variation of the dual solution along it, as in Equation 5. They define the reduced problem P2, from which $\mathrm{d}z^*$, $\mathrm{d}\mu^*$, and $\mathrm{d}\nu^*$ are recovered by solving the linear system Equation 6.



Remarks. (*i*) C^* and H^* capture complementary first- and second-order components of the local cone geometry, as illustrated in the inset. Panel (a) depicts the original conic problem. In (b), C^* restricts perturbations to the admissible directions in the critical cone, but this first-order linearization alone may admit extraneous solutions that violate the original conic constraint. In (c), the quadratic correction induced by H^* accounts for the variation of the cone normal along these directions, removing this ambiguity. Crucially, this correction preserves the parameter sensitivity of the original problem, making the reduced problem first-order equivalent as stated in Theorem 4.1.

(*ii*) Global smoothness of $\mathcal{K}$ is not required. When the active boundary is locally smooth, the restriction of H^* acts, up to sign convention, as the scaled shape operator, and $-\mathrm{d}y^{*T}H^*\mathrm{d}y^*$ is the associated curvature correction, closely related to the second fundamental form (do Carmo, 1976). At conic singularities, such as an SOC apex or PSD matrices with multiple zero eigenvalues, a smooth shape operator is undefined (Fukuda et al., 2023). Our framework instead adopts the variational perspective of Bonnans & Shapiro (2000), where the solution differential is restricted to the critical cone, $\mathrm{d}y^* \in \mathcal{C}_\mathcal{K}(y^*)$. Curvature therefore need only be characterized along these admissible directions, where it can be expressed through the support function of the corresponding second-order tangent set. This directional curvature admits the quadratic representation induced by H^*, while C^* excludes singular directions along which the curvature is not well-defined (see Sections 5.3 and 5.4).

(*iii*) In practical applications, *e.g.*, neural-network training, one need not solve Equation 6 explicitly to backpropagate gradients; instead, the required vector–Jacobian product can be computed by solving the adjoint system with the same symmetric coefficient matrix. The framework may also extend to nonconvex problems with locally convex geometry under suitable regularity conditions, although this extension is beyond the scope of this paper.

5 Applications to NLP, QP, SOCP, and SDP

To demonstrate the utility of Theorem 4.1, we explicitly instantiate the framework to derive reduced formulations for nonlinear programs (NLPs), quadratic programs (QPs) as a special case of NLPs, second-order cone programs (SOCPs), and semidefinite programs (SDPs). Theorem 5.1 below summarizes the resulting formulations for all these classes. Sections 5.1–5.4 then provide the corresponding derivations, constructing the matrices H^* and C^* and capturing the relevant local cone geometry, including nonsmooth configurations at SOC apex blocks and SDP solutions with multiple zero eigenvalues.

Theorem 5.1. *Under the standing regularity assumptions of Section 3.3, consider any of the NLP, QP, SOCP, or SDP problems in the table below. The reduced problem in each corresponding row is first-order equivalent to the original problem at the reference parameter θ_0.*

	Original Problem	**Reduced Problem**
NLP [5.1]	$\min_z \ f(z)$ *s.t.* $\ g_j(z) \le 0$	$\min_z \ \nabla_z f^T z + \frac{1}{2}\bar{z}^T Q_{\mathrm{NLP}} \bar{z}$ *s.t.* $\ g_j(z^*) + \nabla_z g_j^T \bar{z} = 0, \quad j \in J$ *with* $\ Q_{\mathrm{NLP}} = \nabla^2_{zz} f - \sum_{j \in J} \mu_j^* \nabla^2_{zz} g_j$
QP [5.2]	$\min_z \ q^T z + \frac{1}{2} z^T P z$ *s.t.* $\ c_j^T z \le d_j$	$\min_z \ q^T z + \frac{1}{2} z^T P z$ *s.t.* $\ c_j^T z = d_j, \quad j \in J$
SOCP [5.3]	$\min_z \ q^T z$ *s.t.* $\ \|A_j z + b_j\| \le c_j^T z + d_j$	$\min_z \ q^T z + \frac{1}{2}\bar{z}^T Q_{\mathrm{SOCP}} \bar{z}$ *s.t.* $\ \nabla\phi(G_j(z^*))^T (G_j(z) - y_j^*) = 0, \quad j \in J_{\mathrm{na}},$ $G_j(z) = 0, \qquad j \in J_{\mathrm{ap}}$ *with* $\ Q_{\mathrm{SOCP}} = \sum_{j \in J_{\mathrm{na}}} s_j^* \nabla G_j^T \nabla^2 \phi(y_j^*) \nabla G_j$ $G_j(z) := \begin{bmatrix} A_j \\ c_j^T \end{bmatrix} z + \begin{bmatrix} b_j \\ d_j \end{bmatrix}, \quad \phi(x,t) := \|x\|_2 - t$
SDP [5.4]	$\min_{Z = Z^T} \ \langle C, Z \rangle$ *s.t.* $\ \langle A_i, Z \rangle = B_i,$ $Z \succeq 0$	$\min_{Z = Z^T} \ \langle C, Z \rangle + \frac{1}{2}\mathrm{vec}(\bar{Z})^T Q_{\mathrm{SDP}} \mathrm{vec}(\bar{Z})$ *s.t.* $\ \langle A_i, Z \rangle = B_i$ $\langle u_j^* u_{j'}^{*T}, Z \rangle = 0, \quad j, j' \in J,\ j' \le j$ *with* $\ Q_{\mathrm{SDP}} = Z^{*\dagger} \otimes S^* + S^* \otimes Z^{*\dagger}$

where $\bar{z} := z - z^*$ and $\bar{Z} := Z - Z^*$. For NLP and QP, j indexes the inequality constraints and J is the active subset. For SOCP, j indexes the SOC blocks, while J_{na} and J_{ap} denote the non-apex and apex active subsets, respectively. For SDP, i indexes the affine constraints, while $\{u_j^*\}_{j \in J}$ is an orthonormal basis for $\ker(Z^*)$. For compactness, we suppress the dependence of data on θ.

5.1 Nonlinear Programs

Consider the NLP in Theorem 5.1, where $f_\theta(z), g_{j,\theta}(z) : \mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}$ and their Jacobians are jointly $\mathcal{C}^1$ in (z, θ). The inequality constraints can be represented by the non-positive orthant $\mathcal{K} = \mathbb{R}^m_-$. Denote by $J := \{j : y_j^* = g_{j,\theta_0}(z^*) = 0\}$ the active set at the reference solution. Its critical cone is given by $\mathcal{C}_{\mathcal{K}}(y^*) = \{\mathrm{d}y \in \mathbb{R}^m : \mathrm{d}y_j = 0,\ j \in J\}$, implying that perturbations must keep active constraints tight to zero. Consequently, $C^* = C_\theta^*$ is the coordinate-selection matrix associated with J. Since the boundary of the nonpositive orthant is locally polyhedral and hence has no curvature, $H^* = 0$. Substituting these expressions into Theorem 4.1 yields the reduced NLP in Theorem 5.1, recovering the classical active-set reduction for nonlinear programs (Fiacco, 1983).

5.2 Quadratic Programming

The case of NLP with $f_\theta(z) = q_\theta^T z + \frac{1}{2} z^T P_\theta z$ and $g_{j,\theta}(z) = c_{j,\theta}^T z - d_{j,\theta}$ readily recovers the reduced problem for the case of QP derived in Magoon et al. (2025).

5.3 Second-order Cone Programs

The j-th second-order cone (SOC) constraint in Theorem 5.1 takes the form

$$y_j \in \mathcal{K}_{\mathrm{SOC}_j} := \{(x,t) : \phi(x,t) \le 0\},$$

where $y_j = G_j(z) := (A_j z + b_j, c_j^T z + d_j)$ and $\phi(x,t) := \|x\|_2 - t$. Recall that y_j^* is the fixed value of $G_j(z^*)$ at the reference problem data. Let $\mu_j = (u_j, s_j)$ denote the corresponding dual variable. Under our assumptions, each SOC block at the solution is either inactive, with $y_j^* \in \operatorname{int} \mathcal{K}_{\mathrm{SOC}_j}$, or active, with non-apex and apex index sets

$$J_{\mathrm{na}} := \{j : y_j^* \in \partial\mathcal{K}_{\mathrm{SOC}_j} \setminus \{0\},\ \mu_j^* \in \partial\mathcal{K}^*_{\mathrm{SOC}_j} \setminus \{0\}\},$$
$$J_{\mathrm{ap}} := \{j : y_j^* = 0,\ \mu_j^* \in \operatorname{int}(\mathcal{K}^*_{\mathrm{SOC}_j})\}.$$

The critical cones for active blocks are

$$\mathcal{C}_{\mathcal{K}_{\mathrm{SOC}_j}}(y_j^*) = \begin{cases} \{\mathrm{d}y : \nabla\phi(y_j^*)^T \mathrm{d}y = 0\}, & j \in J_{\mathrm{na}}, \\ \{0\}, & j \in J_{\mathrm{ap}}. \end{cases} \tag{7}$$

Thus, first-order solution perturbations lie in the tangent hyperplane for non-apex blocks and vanish for apex blocks. These restrictions and the cone's local curvature determine $C^*_{j,\theta}$ and H_j^*:

Lemma 5.1. *For each active SOC block,*

$$(H_j^*, C^*_{j,\theta}) = \begin{cases} \left(-s_j^* \nabla^2\phi(y_j^*), \nabla\phi(G_j(z^*))^T\right), & j \in J_{\mathrm{na}}, \\ (0, I), & j \in J_{\mathrm{ap}}. \end{cases}$$

The derivation appears in Appendix C.1. Substituting these expressions into Theorem 4.1 yields the reduced SOCP in Theorem 5.1.

Remark. Beyond giving a closed-form reduction, this specialization illustrates the local smoothness requirement underlying Theorem 4.1. The cone boundary need not admit a smooth representation in every ambient direction; its curvature is required only along admissible directions in the critical cone. Although an SOC constraint can be written as the scalar inequality $\phi(y_j) \le 0$, this representation is nonsmooth at the apex $y_j^* = 0$, so direct differentiation as a smooth NLP is not applicable at such configurations. By Equation 7, however, the critical cone at the apex is $\{0\}$, so there is no nonzero admissible direction along which curvature must be evaluated, resulting in a well-defined reduction.

5.4 Semidefinite Programs

Let $Z \in \mathbb{S}^n$ and $S \in \mathbb{S}^n$ be the primal and dual matrices associated with the constraint $Z \succeq 0$ of the SDP in Theorem 5.1. To describe the local geometry of the PSD cone, we write the compact eigendecomposition $Z^* = U_1^* \Lambda_1^* U_1^{*T}$ with $\Lambda_1^* \succ 0$, and let U_0^* be an orthonormal basis for $\ker(Z^*)$ with columns u_j^* indexed by $j \in J$. Under our assumptions, the critical cone is

$$\mathcal{C}_{\mathbb{S}^n_+}(Z^*) = \{\mathrm{d}Z \in \mathbb{S}^n : U_0^{*T} \mathrm{d}Z\, U_0^* = 0\} = \{\mathrm{d}Z : u_j^{*T} \mathrm{d}Z\, u_{j'}^* = 0,\ j, j' \in J\}. \tag{8}$$

Thus, admissible first-order perturbations cannot modify the block of Z^* supported on its zero-eigenspace. This restriction determines $C^* = C^*_\theta$, while the curvature of the PSD cone along the remaining admissible directions determines H^* as follows:

Lemma 5.2. *The differential of the dual matrix admits the vector-form decomposition*

$$\mathrm{vec}(\mathrm{d}S^*) = H^* \mathrm{vec}(\mathrm{d}Z^*) + C^{*T} \mathrm{d}\hat{\mu}, \qquad H^* = -\left(Z^{*\dagger} \otimes S^* + S^* \otimes Z^{*\dagger}\right),$$

where the rows of C^* are $\mathrm{vec}(u_j^* u_j^{*T})^T$ and $\mathrm{vec}(u_j^* u_{j'}^{*T} + u_{j'}^* u_j^{*T})^T$ for $j, j' \in J, j' < j$.

A detailed derivation is deferred to Appendix C.2. Substituting these expressions into Theorem 4.1 yields the reduced SDP in Theorem 5.1.

Remark. This instance further illustrates that differentiability does not require cone curvature to be well defined in all directions, but only along admissible directions in the critical cone. In the SDP case, nonsmoothness arises when perturbations change the zero-eigenspace block of Z^*. The critical-cone condition $U_0^{*T} \mathrm{d}Z^* U_0^* = 0$ removes exactly those directions; see Appendix C.2 for an illustrative example. Therefore, the reduced problem remains well defined even when Z^* has multiple zero eigenvalues.

Table 6.1: Synthetic SOCP and SDP performance and scalability.

Method	Metric	SOCP				SDP			
	Problem size	20	100	500	700	20×20	50×50	100×100	150×150
dOPT (MOSEK)	Accuracy	$\mathbf{2.14 \times 10^{-8}}$	1.28×10^{-7}	3.06×10^{-6}	$\mathbf{2.26 \times 10^{-5}}$	$\mathbf{1.52 \times 10^{-8}}$	$\mathbf{5.16 \times 10^{-9}}$	$\mathbf{2.27 \times 10^{-9}}$	$\mathbf{7.32 \times 10^{-9}}$
	Forward [ms]	14.3	260.1	64055.1	**346639.9**	**24.5**	**615.4**	15040.9	**142613.1**
	Backward [ms]	**1.6**	**7.2**	**184.0**	**1442.8**	**3.4**	**193.2**	**6946.4**	**82890.7**
	Total [ms]	15.9	**267.3**	**64239.1**	**348082.7**	**27.9**	**808.6**	**21987.3**	**225503.8**
CVXPYLayers (SCS)	Accuracy	2.15×10^{-8}	$\mathbf{3.04 \times 10^{-8}}$	$\mathbf{1.15 \times 10^{-6}}$	Failed	1.80×10^{-6}	2.57×10^{-6}	2.58×10^{-6}	Failed
	Forward [ms]	**2.0**	**152.7**	160960.3	–	29.4	934.6	39438.4	–
	Backward [ms]	1.9	773.0	83147.4	–	69.2	6761.2	115012.5	–
	Total [ms]	**3.9**	925.7	244107.7	–	98.6	7695.8	154450.9	–
FFOLayer (MOSEK)	Accuracy	$\mathbf{2.14 \times 10^{-8}}$	$1.22 \times 10^{-7*}$	$2.00 \times 10^{-6*}$	Failed	$\mathbf{1.52 \times 10^{-8}}$	$\mathbf{5.16 \times 10^{-9}}$	$\mathbf{2.27 \times 10^{-9}}$	$\mathbf{7.32 \times 10^{-9}}$
	Forward [ms]	16.4	268.6	**63283.2**	–	25.4	638.9	**14551.6**	148793.2
	Backward [ms]	31.3	334.8	46807.9	–	25.6	689.4	15968.4	199709.1
	Total [ms]	47.7	603.5	110091.2	–	51.1	1328.4	30520.0	348502.3

* Some FFOLayer backward passes errored; results are averaged over successful runs: 9/10 instances at $n = 100$ and 6/10 at $n = 500$.

6 Experiments

We evaluate dOPT through two experiments. First, we assess gradient accuracy and computational scalability on synthetic SOCP and SDP instances. Second, we demonstrate dOPT in an end-to-end learning benchmark with a differentiable SDP layer, focusing on backward computational cost during training. As baselines, we use CVXPYLayers (with diffCP as its backend) (Agrawal et al., 2019a;b) and FFOLayer (Zhao et al., 2025). To our knowledge, these are the closest publicly available differentiation frameworks applicable to general conic programs and supporting evaluation on both SOCP and SDP instances[1]. In our experiments, both dOPT and FFOLayer solve the forward problems using MOSEK (Andersen & Andersen, 2000), while CVXPYLayers uses SCS (O'Donoghue et al., 2026). Our comparisons focus primarily on backward computational cost.

6.1 Synthetic Evaluation

We evaluate dOPT on randomly generated SOCP and SDP instances of increasing size, averaging over ten instances per setting. We report forward and backward runtimes and forward-solution accuracy, measured by the duality gap. Problem-generation and implementation details are provided in Appendix D. Gradient validation in Appendix E shows that optimal-value gradients computed by dOPT agree with analytic gradients from the envelope theorem (Afriat, 1971) to relative errors of approximately 10^{-9} for SOCPs and 10^{-7} for SDPs.

Table 6.1 summarizes performance and scalability. dOPT has the lowest backward runtime at every tested size. For SOCPs at $n = 500$, it is over two orders of magnitude faster than both baselines. At $n = 700$, only dOPT completes the benchmark; both baselines terminate with errors on nearly every instance. FFOLayer also has backward-pass failures at smaller sizes, as noted in the table. For SDPs at $n = 100$, dOPT's backward pass is $2.3\times$ faster than FFOLayer and $16.6\times$ faster than CVXPYLayers; at $n = 150$, CVXPYLayers terminates with errors on every instance.

6.2 Learning Stable Switched Linear Systems

To demonstrate dOPT in an end-to-end learning benchmark with a differentiable SDP layer, we consider a discrete-time switched linear system:

$$x_{k+1} = A_{\sigma(k)} x_k, \tag{9}$$

where $A_1, \ldots, A_M \in \mathbb{R}^{d \times d}$ are mode matrices and $\sigma(k)$ selects the active mode. Such systems model dynamics that switch between operating regimes. Given observed trajectories, the goal is to learn both the mode matrices A_i and the switching signal $\sigma(k)$. Minimizing prediction error alone, however, does not ensure that the learned dynamics are stable. To promote stability, we regularize the training loss using a standard common quadratic Lyapunov certificate (Ahmadi et al., 2014). Specifically, a differentiable SDP layer computes

$$t^*\big((A_i)_{i=1}^M\big) = \max_{P,t} \; t \quad \text{s.t.} \quad P \succeq 0, \quad \operatorname{tr}(P) = 1, \quad P - A_i^\top P A_i \succeq tI, \quad i = 1, \ldots, M. \tag{10}$$

[1] We excluded Moreau (Barratt et al., 2026) due to SDP failure, DiffOpt.jl (Besançon et al., 2024) for lack of native PyTorch integration, and diffQCP (Healey et al., 2026) due to inconsistent gradients across solvers.

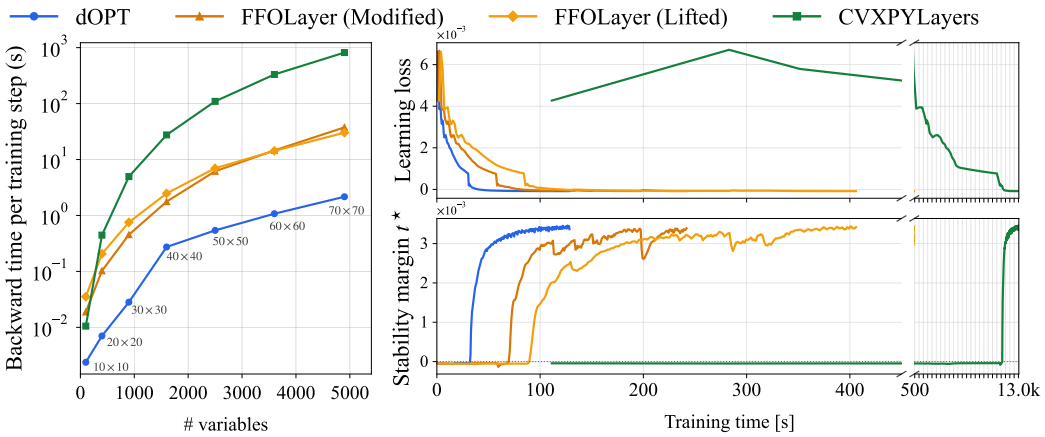


Figure 6.1: Learning stable switched linear systems. Left: backward time against number of optimization variables $O(d^2)$ for synthetic switched linear systems. Right: learning loss and stability margin t^* versus training time for the 10-vehicle platoon system ($N = 10$).

Table 6.2: Performance of the adapted vehicle-platoon benchmark for 3, 5, and 10 vehicles.

	9×9 ($N = 3$)			15×15 ($N = 5$)			30×30 ($N = 10$)		
Method	Fwd [ms]	Bwd [ms]	Train [s]	Fwd [ms]	Bwd [ms]	Train [s]	Fwd [ms]	Bwd [ms]	Train [s]
dOPT	**12.7**	**1.4**	**4.3**	41.4	**3.6**	**12.0**	468.4	**26.1**	**128.8**
FFOLayer (Modified)	14.3	16.2	8.0	**41.0**	43.6	21.4	**459.0**	472.7	241.9
FFOLayer (Lifted)	28.1	30.5	15.1	84.3	88.9	44.2	797.4	815.1	405.9
CVXPYLayers	93.3	6.6	28.6	345.3	84.3	115.2	5592.1	4948.6	13047.9

A positive margin $t^* > 0$ certifies stability under arbitrary switching. We use the scaled margin αt^* to regularize trajectory fitting, giving $\mathcal{L} = \mathcal{L}_{\text{traj}} - \alpha t^*$.

Figure 6.1(left) evaluates scalability on synthetic switched systems. Since the original FFOLayer implementation does not directly support parametric PSD constraints as in Equation 10, we compare against a minimally modified version, *FFOLayer (Modified)*, and an equivalent lifted formulation, *FFOLayer (Lifted)*. At $d = 40$, dOPT's backward pass is $6.4\times$ faster than the next-fastest FFOLayer variant and $97.9\times$ faster than CVXPYLayers; at $d = 70$, it remains over an order of magnitude faster than FFOLayer and over two orders of magnitude faster than CVXPYLayers.

We also adapt the networked vehicle-platoon benchmark of Makhlouf & Kowalewski (2014); Chen et al. (2015), with $N \in \{3, 5, 10\}$ vehicles. Each vehicle regulates a three-dimensional physical state using information from the others, with modes corresponding to normal communication and communication loss. This serves as a controlled benchmark for differentiation through the SDP layer rather than a real-world platooning solution. All methods learn nearly identical positive stability margins, while dOPT is fastest across all three sizes, completing training about 1.8–$1.9\times$ faster than the next-fastest method. Figure 6.1(right) shows the 10-vehicle training dynamics, and Table 6.2 reports results for all three sizes; additional details appear in Appendix D.2.

7 Conclusion

We introduced dOPT, a framework for differentiating through parametric convex conic programs via first-order equivalent equality-constrained QPs. Explicit reductions for NLPs, QPs, SOCPs, and SDPs accommodate geometrically singular configurations under standard regularity assumptions and enable efficient differentiation independently of the forward solver. Notable limitations include the restriction to convex problems, reliance on high-quality solutions, and limited exploitation of algebraic structure, especially for SDPs. Forward solves remain the main computational bottleneck, which dOPT does not address. These limitations motivate future work on extensions to nonconvex problems, support for approximate solutions (*e.g.*, from L2O (Chen et al., 2022)), more effective use of structure, weaker regularity assumptions, and other spectral cones.

AI USE STATEMENT

In this work, we used generative AI tools to assist with mathematical derivations and proofs. We did not use generative AI tools to generate synthetic datasets, and the remaining required-disclosure tasks were either not used or not applicable to this work. Additionally, we used generative AI tools to draft and polish portions of the manuscript and to assist with literature retrieval and discovery. All AI-assisted mathematical content was independently checked and verified by the authors, suggested references were verified against the original sources, and all AI-assisted writing was reviewed and edited by the authors. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The assumptions and proof of Theorem 4.1 are given in Section 3.3 and Appendices A–B. Explicit instantiations for NLPs, QPs, SOCPs, and SDPs are presented in Section 5, with detailed derivations in Appendix C. Appendices D and E provide experimental and implementation details, additional numerical results, and gradient validation. Our anonymous code repository (`https://github.com/dopt-anon/dOPT`) includes the complete dOPT implementation and its explicit reductions, together with the scripts and configurations needed to generate synthetic instances, run baseline comparisons and the SDP-based learning task, and reproduce all experiments reported in the main paper and appendices.

REFERENCES

Sydney Naphtali Afriat. Theory of maxima and the method of Lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and Jeremy Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, 2019a.

Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M. Moursi. Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, 1(2):107–115, 2019b.

Amir Ali Ahmadi, Raphaël M. Jungers, Pablo A. Parrilo, and Mardavij Roozbehani. Joint spectral radius and path-complete graph Lyapunov functions. *SIAM Journal on Control and Optimization*, 52(1):687–717, 2014.

Farid Alizadeh, Jean-Pierre A. Haeberly, and Michael L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming*, 77(1):111–128, 1997.

Brandon Amos and Jeremy Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145, 2017.

Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and Jeremy Zico Kolter. Differentiable MPC for end-to-end planning and control. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Erling D. Andersen and Knud D. Andersen. The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In *High Performance Optimization*, pp. 197–232. Springer, 2000.

Roberto Andreani, Gabriel Haeser, Leonardo Makoto Mito, and Héctor Ramírez. A minimal face constant rank constraint qualification for reducible conic programming. arXiv preprint arXiv:2304.13881, 2023.

Roberto Andreani, Gabriel Haeser, Leonardo Makoto Mito, and Héctor Ramírez. Weak notions of nondegeneracy in nonlinear semidefinite programming. *Mathematical Programming*, 205(1): 1–32, 2024.

Antoine Bambade, Fabian Schramm, Adrien B. Taylor, and Justin Carpentier. Leveraging augmented-Lagrangian techniques for differentiating over infeasible quadratic programs in machine learning. In *International Conference on Learning Representations*, 2024.

Chenglong Bao, Chao Ding, Fuxiaoyue Feng, and Jingyu Li. Stratification for nonlinear semidefinite programming. arXiv preprint arXiv:2601.08362, 2026.

Shane Barratt. On the differentiability of the solution to convex optimization problems. arXiv preprint arXiv:1804.05098, 2018.

Shane Barratt, Parth Nobel, and Steven Diamond. Moreau: GPU-native differentiable optimization. Software, 2026. URL `https://moreau.so`.

Mathieu Besançon, Joaquim Dias Garcia, Benoît Legat, and Akshay Sharma. Flexible differentiable optimization via model transformations. *INFORMS Journal on Computing*, 36(2):456–478, 2024.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5230–5242, 2022.

Jérôme Bolte, Tam Le, Edouard Pauwels, and Tony Silveti-Falls. Nonsmooth implicit differentiation for machine-learning and optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13537–13549, 2021. URL `https://arxiv.org/abs/2106.04350`.

Jérôme Bolte, Edouard Pauwels, and Cheik Traoré. The adjoint state method for parametric definable optimization without smoothness or uniqueness. arXiv preprint arXiv:2603.26503, 2026.

Joseph Frédéric Bonnans and Héctor Ramírez C. Perturbation analysis of second-order cone programming problems. *Mathematical Programming*, 104(2):205–227, 2005.

Joseph Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer, 2000.

Joseph Frédéric Bonnans, Roberto Cominetti, and Alexander Shapiro. Sensitivity analysis of optimization problems under second order regular constraints. *Mathematics of Operations Research*, 23(4):806–831, 1998. doi: 10.1287/moor.23.4.806.

Joseph Frédéric Bonnans, Roberto Cominetti, and Alexander Shapiro. Second order optimality conditions based on parabolic second order tangent sets. *SIAM Journal on Optimization*, 9(2): 466–492, 1999. doi: 10.1137/S1052623496306760.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Andrew Butler. SCQPTH: An efficient differentiable splitting method for convex quadratic programming. arXiv preprint arXiv:2308.08232, 2023.

Andrew Butler and Roy H. Kwon. Efficient differentiable quadratic programming layers: An ADMM approach. *Computational Optimization and Applications*, 84(2):449–476, 2023.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

Xin Chen, Stefan Schupp, Ibtissem Ben Makhlouf, Erika Ábrahám, Goran Frehse, and Stefan Kowalewski. A benchmark suite for hybrid systems reachability analysis. In *NASA Formal Methods Symposium*, pp. 408–414, 2015.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and Jeremy Zico Kolter. End-to-end differentiable physics for learning and control. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Lijun Ding and Madeleine Udell. A strict complementarity approach to error bound and sensitivity of solution of conic programs. *Optimization Letters*, 17:1551–1574, 2023. doi: 10.1007/s11590-022-01942-1.

Manfredo Perdigão do Carmo. *Differential geometry of curves and surfaces*. Prentice-Hall, 1976.

Anthony Vincent Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, 1983.

Ellen Hidemi Fukuda, Gabriel Haeser, and Leonardo Makoto Mito. On the weak second-order optimality condition for nonlinear semidefinite and second-order cone programming. *Set-Valued and Variational Analysis*, 31, 2023. doi: 10.1007/s11228-023-00676-1. Article 15.

Quill Healey, Parth Nobel, and Stephen Boyd. Differentiating through a quadratic cone program. *Optimization Letters*, pp. 1–16, 2026.

Connor Holmes, Frederike Dümbgen, and Timothy David Barfoot. SDPRLayers: Certifiable backpropagation through polynomial optimization problems in robotics. arXiv preprint arXiv:2405.19309, 2024.

William Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Department of Mathematics, University of Chicago, 1939.

Harold William Kuhn and Albert William Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, 1951.

Junhyeong Lee, Sangjin Jin, and Yongjae Lee. Decision-focused learning via tangent-space projection of prediction error. arXiv preprint arXiv:2605.01361, 2026.

Adrian Stephen Lewis. Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization*, 13(3):702–725, 2002. doi: 10.1137/S1052623401387623.

Yuxuan Linghu, Zhiyuan Liu, and Qi Deng. A penalty approach for differentiation through black-box quadratic programming solvers. arXiv preprint arXiv:2602.14154, 2026.

Kang Liu and Jianchen Hu. Residual-controlled Douglas–Rachford splitting for differentiable solver layers. arXiv preprint arXiv:2608.14470, 2026.

Yulan Liu, Ying Sun, and Shaohua Pan. Computation of graphical derivatives of normal cone maps to a class of conic constraint sets. *Set-Valued and Variational Analysis*, 27(3):783–806, 2019.

Connor W. Magoon, Fengyu Yang, Noam Aigerman, and Shahar Z. Kovalsky. Differentiation through black-box quadratic programming solvers. In *Advances in Neural Information Processing Systems*, volume 38, pp. 108486–108517, 2025.

Ibtissem Ben Makhlouf and Stefan Kowalewski. Networked cooperative platoon of vehicles for testing methods and verification tools. In *International Workshop on Applied Verification for Continuous and Hybrid Systems*, pp. 37–42, 2014.

Ashkan Mohammadi, Boris Sholimovich Mordukhovich, and M. Ebrahim Sarabi. Parabolic regularity in geometric variational analysis. *Transactions of the American Mathematical Society*, 374: 1711–1763, 2021. doi: 10.1090/tran/8253.

Jorge Nocedal and Stephen Joseph Wright. *Numerical optimization*. Springer, 2006.

Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. SCS: Splitting conic solver, version 3.3.1, 2026. URL `https://github.com/cvxgrp/scs`.

Alex Oshin, Rahul Vodeb Ghosh, and Evangelos A. Theodorou. Scalable deep unfolding of conic optimizers. arXiv preprint arXiv:2606.13825, 2026.

François Pacaud. Sensitivity analysis for parametric nonlinear programming: A tutorial. arXiv preprint arXiv:2504.15851, 2025.

Jianming Pan, Zeqi Ye, Xiao Yang, Xu Yang, Weiqing Liu, Lewen Wang, and Jiang Bian. BPQP: A differentiable convex optimization framework for efficient end-to-end learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Anselm Paulus, Georg Martius, and Vít Musil. LPGD: A general framework for backpropagation through embedded optimization layers. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39989–40014, 2024.

Stephen Michael Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5(1):43–62, 1980. URL http://www.jstor.org/stable/3689393.

Ralph Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, 1970.

Ralph Tyrrell Rockafellar and Roger Wets. *Variational analysis*, volume 317. Springer, 1998.

Andrew Werner Rosemberg, Joaquim Dias Garcia, François Pacaud, Robert B. Parker, Benoît Legat, Kaarthik Sundar, Russell Bent, and Pascal Van Hentenryck. A general and streamlined differentiable optimization framework. arXiv preprint arXiv:2510.25986, 2025.

Maximilian Schaller and Stephen Boyd. Code generation for solving and differentiating through convex optimization problems. arXiv preprint arXiv:2504.14099, 2025.

Alexander Shapiro and Michael K. Fan. First order optimality conditions in semidefinite programming. *SIAM Journal on Optimization*, 7(2):533–550, 1997.

Mattia Solla and Johannes Ovrelid Royset. Optimistic bilevel optimization with composite lower-level problem. arXiv preprint arXiv:2602.05417, 2026.

Defeng Sun. The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications. *Mathematics of Operations Research*, 31(4): 761–776, 2006. doi: 10.1287/moor.1060.0195.

Haixiang Sun, Ye Shi, Jingya Wang, Hoang Duong Tuan, Harold Vincent Poor, and Dacheng Tao. Alternating differentiation for optimization layers. arXiv preprint arXiv:2210.01802, 2022.

Xin Wang, Ralph Tyrrell Rockafellar, and Xuegang Ban. Directional influence function: Estimating training data influence in constrained learning. arXiv preprint arXiv:2607.23388, 2026.

Ziqi Xu, Lin Cheng, Di Wu, and Shengping Gong. Parameter optimization in trajectory planning via differentiable convex programming. arXiv preprint arXiv:2512.03557, 2025.

Ziran Yin and Liwei Zhang. Perturbation analysis of a class of conic programming problems under Jacobian uniqueness conditions. *Journal of Industrial and Management Optimization*, 15(3): 1387–1397, 2019. doi: 10.3934/jimo.2018100.

Zihao Zhao, Kai-Chia Mo, Shing-Hei Ho, Brandon Amos, and Kai Wang. A fully first-order layer for differentiable optimization. arXiv preprint arXiv:2512.02494, 2025.

A PRELIMINARIES

This section introduces additional definitions concerning cone geometry and regularity assumptions (Boyd & Vandenberghe, 2004; Bonnans & Shapiro, 2000; Rockafellar & Wets, 1998). θ is omitted for simplicity.

Definition A.1 (Face)**.** *A face of a convex set C is a convex subset $F \subseteq C$ such that, whenever a strict convex combination of two points in C lies in F, both points lie in F. That is, if $\theta x+(1-\theta)y \in F$ for some $x, y \in C$ and $0 < \theta < 1$, then $x, y \in F$.*

Definition A.2 (Active face)**.** *The* active face *of a convex cone $\mathcal{K} \subseteq \mathbb{R}^n$ at a point $y \in \mathcal{K}$ is defined as*

$$\mathcal{F}_{\mathcal{K}}(y) := \bigcap \{F \subset \mathcal{K} : F \text{ is a face of } \mathcal{K}, y \in F\}$$

Definition A.3 (Non-degeneracy)**.** *The problem P1 is* non-degenerate *at the primal solution z^* if the following conditions hold:*

$$\nabla h\,(z^*)\,\mathbb{R}^n = \mathbb{R}^l,$$
$$\nabla G\,(z^*)\ker(\nabla h\,(z^*)) + \operatorname{lin}\left(\mathcal{T}_{\mathcal{K}}\,(y^*)\right) = \mathbb{R}^m,$$

where $\operatorname{lin}(\mathcal{T}_{\mathcal{K}}(y^*))$ denotes the largest linear subspace contained in the tangent cone $\mathcal{T}_{\mathcal{K}}(y^*)$.

Define the critical directions of Problem P1 at z^* by

$$\mathcal{D}^* := \{d \in \mathbb{R}^n : \nabla G(z^*)d \in \mathcal{C}_{\mathcal{K}}(y^*), \quad \nabla h(z^*)d = 0\}.$$

For $y \in \mathcal{K}$ and $v \in \mathcal{T}_{\mathcal{K}}(y)$, define the second-order tangent set by

$$\mathcal{T}^2_{\mathcal{K}}(y, v) := \left\{w : \operatorname{dist}\left(y + tv + \tfrac{1}{2}t^2 w, \mathcal{K}\right) = o(t^2) \text{ as } t \downarrow 0\right\}.$$

For a set D, let $\sigma(a, D) := \sup_{w \in D}\langle a, w\rangle$ denote its support function.

Definition A.4 (Second-order sufficient condition (SOSC))**.** *Problem P1 satisfies the SOSC at the optimal primal-dual solution (z^*, μ^*, ν^*) if, for every $d \in \mathcal{D}^* \setminus \{0\}$,*

$$d^T \nabla^2_{zz}\mathcal{L}(z^*, \mu^*, \nu^*)d - \sigma\left(-\mu^*, \mathcal{T}^2_{\mathcal{K}}\left(y^*, \nabla G(z^*)d\right)\right) > 0.$$

The second-order sufficient condition requires the Lagrangian curvature, together with the curvature of the cone, to be strictly positive along every nonzero critical direction. Intuitively, this ensures sufficient curvature along the feasible first-order directions relevant to optimality, making the stationary solution locally isolated.

Definition A.5 ($\mathcal{C}^2$-cone reducibility)**.** *(Bonnans & Shapiro, 2000, Definition 3.135) A closed convex set $\mathcal{K} \subseteq \mathbb{R}^m$ is said to be $\mathcal{C}^2$-cone reducible at $y^* \in \mathcal{K}$ if there exist an open neighborhood U of y^*, a pointed closed convex cone $\mathcal{Q} \subseteq \mathbb{R}^q$, and a twice continuously differentiable mapping $\Xi : U \to \mathbb{R}^q$, for some integer $q \geq 0$, satisfying:*

(i) $\Xi(y^*) = 0$*;*

(ii) $D\Xi(y^*) : \mathbb{R}^m \to \mathbb{R}^q$ *is onto;*

(iii) $\mathcal{K} \cap U = \{y \in U : \Xi(y) \in \mathcal{Q}\}$.

Intuitively, $\mathcal{C}^2$-cone reducibility means that local feasibility can be described by a twice continuously differentiable mapping into a simpler pointed closed convex cone. The class of $\mathcal{C}^2$-cone reducible sets is rich, including all the polyhedral convex sets, second-order cones, positive semidefinite cones, and epigraph cones of Ky Fan matrix k-norms. This property is also preserved under finite Cartesian products (Bonnans & Shapiro, 2000).

Lemma A.1. *(Ding & Udell, 2023) Under strict complementarity and non-degeneracy, the active faces associated with the complementary pair (y^*, μ^*) satisfy*

$$\begin{aligned} \mathcal{F}_{\mathcal{K}}(y^*) &= \mu^{*\perp} \cap \mathcal{K} \\ \mathcal{F}_{\mathcal{K}^*}(\mu^*) &= -\mathcal{N}_{\mathcal{K}}(y^*) = y^{*\perp} \cap \mathcal{K}^*. \end{aligned} \tag{11}$$

Lemma A.1 shows that the active faces follow from the KKT conditions under strict complementarity. In particular, the primal active face is obtained by combining primal cone feasibility Equation 2c with complementarity Equation 2e.

B PROOF OF THEOREM 4.1

Proof. The KKT conditions of the parametric Problem P1 are:

$$\begin{aligned} \nabla_z f_\theta(z^*(\theta)) - \nabla_z h_\theta(z^*(\theta))^T \nu^*(\theta) - \nabla_z G_\theta(z^*(\theta))^T \mu^*(\theta) &= 0 \\ h_\theta(z^*(\theta)) &= 0 \\ y^*(\theta) &\in \mathcal{K} \\ \mu^*(\theta) &\in \mathcal{K}^* \\ \langle \mu^*(\theta), y^*(\theta) \rangle &= 0. \end{aligned} \tag{12}$$

Combining complementarity with primal and dual feasibility shows that y^* and μ^* lie in their respective active faces: $y^* \in \mathcal{K} \cap \mu^{*\perp} = \mathcal{F}_{\mathcal{K}}(y^*)$ and $\mu^* \in \mathcal{K}^* \cap y^{*\perp} = \mathcal{F}_{\mathcal{K}^*}(\mu^*) = -\mathcal{N}_{\mathcal{K}}(y^*)$ by Lemma A.1.

Now we write the differential of the original Problem P1 evaluated at (z^*, μ^*, ν^*) with fixed $\theta = \theta_0$. To simplify notation, once θ is fixed at θ_0, we omit explicit subscripts and function arguments (*e.g.*, $_\theta$ and $(\cdot)$); the same convention is used in the subsequent differential derivations. The differential of the KKT system is:

$$\begin{aligned} &\nabla^2_{zz}\mathcal{L}\mathrm{d}z^* + \nabla^2_{z\theta} f \mathrm{d}\theta - \sum_{k=1}^{l} \nu_k^* \nabla^2_{z\theta} h_k \mathrm{d}\theta \\ &\quad - \sum_{i=1}^{m} \mu_i^* \nabla^2_{z\theta} G_i \mathrm{d}\theta - \nabla_z h^T \mathrm{d}\nu^* - \nabla_z G^T \mathrm{d}\mu^* = 0 \\ &\qquad\qquad \nabla_z h \mathrm{d}z^* + \nabla_\theta h \mathrm{d}\theta = 0 \\ &\qquad\qquad\qquad\qquad \mathrm{d}y^* \in \mathcal{C}_{\mathcal{K}}(y^*) \\ &\qquad\qquad\qquad\qquad \mathrm{d}\mu^* \in \mathcal{C}_{\mathcal{K}^*}(\mu^*), \end{aligned} \tag{13}$$

where the last two conditions involve critical cones at the optimal solution, $\mathcal{C}_{\mathcal{K}}(y^*)$ and $\mathcal{C}_{\mathcal{K}^*}(\mu^*)$, as defined in Section 3.2: $\mathcal{C}_{\mathcal{K}}(y^*) = \mathcal{T}_{\mathcal{K}}(y^*) \cap \mu^{*\perp}$, $\mathcal{C}_{\mathcal{K}^*}(\mu^*) = \mathcal{T}_{\mathcal{K}^*}(\mu^*) \cap y^{*\perp}$.

The differential of the complementarity condition can be omitted because the critical-cone conditions already imply $\langle \mathrm{d}\mu^*, y^* \rangle = \langle \mu^*, \mathrm{d}y^* \rangle = 0$.

We next show that there exist matrices $H^* \in \mathbb{R}^{m \times m}$ and $C^* \in \mathbb{R}^{r_C \times m}$ such that $\mathrm{d}\mu^*$ admits the representation

$$\mathrm{d}\mu^* = H^* \mathrm{d}y^* + C^{*T} \mathrm{d}\hat{\mu}. \tag{14}$$

By C^2-cone reducibility, there exist an open neighborhood $\mathcal{Y}$ of y^*, a pointed closed convex cone $D \subseteq \mathbb{R}^r$, and a C^2 mapping $\Xi : \mathcal{Y} \to \mathbb{R}^r$ such that

$$\Xi(y^*) = 0, \qquad \Xi'(y^*) \text{ is onto}, \qquad \mathcal{K} \cap \mathcal{Y} = \{y \in \mathcal{Y} : \Xi(y) \in D\}.$$

Here Ξ' denotes the Jacobian. Since $\Xi'(y^*)$ has full row rank and Ξ' is continuous, we may choose $\mathcal{Y}$ small enough that $\Xi'(y)$ has full row rank for every $y \in \mathcal{Y}$. The normal-cone reduction formula (Yin & Zhang, 2019, Lemma 2.1) gives

$$\mathcal{N}_{\mathcal{K}}(y) = \Xi'(y)^T \mathcal{N}_D(\Xi(y)) \qquad (y \in \mathcal{K} \cap \mathcal{Y}).$$

For each θ near θ_0, the KKT conditions give $-\mu^*(\theta) \in \mathcal{N}_{\mathcal{K}}(y^*(\theta))$. Applying the reduction formula at $y^*(\theta)$ therefore gives a unique $u(\theta) \in \mathcal{N}_D(\Xi(y^*(\theta)))$ such that

$$-\mu^*(\theta) = \Xi'(y^*(\theta))^T u(\theta).$$

Since Ξ' is C^1 and has full row rank locally, $u(\theta)$ is differentiable along the differentiable primal–dual solution branch. Set $u^* := u(\theta_0) \in D^\circ$, where D° is the negative polar. Since $\Xi'(y^*)^T$ is injective, strict complementarity implies $u^* \in \operatorname{ri} D^\circ = \operatorname{int} D^\circ$. Consequently, $D \cap (u^*)^\perp = \{0\}$.

To identify the critical cone, first note that surjectivity of $\Xi'(y^*)$ ensures metric regularity, and hence metric subregularity, of $y \mapsto \Xi(y) - D$ at $(y^*, 0)$. Applying Liu et al. (2019, Corollary 2.1) to the local representation $\mathcal{K} = \Xi^{-1}(D)$ therefore gives

$$\begin{aligned} \mathcal{T}_{\mathcal{K}}(y^*) &= \{v : \Xi'(y^*)v \in \mathcal{T}_D(\Xi(y^*))\} \\ &= \{v : \Xi'(y^*)v \in D\}, \end{aligned}$$

where the last equality uses $\Xi(y^*) = 0$ and $\mathcal{T}_D(0) = D$. By the definition of the critical cone and $-\mu^* = \Xi'(y^*)^T u^*$, it follows that

$$\begin{aligned}\mathcal{C}_{\mathcal{K}}(y^*) &= \mathcal{T}_{\mathcal{K}}(y^*) \cap (\mu^*)^\perp \\ &= \{v : \Xi'(y^*)v \in D,\ \langle u^*, \Xi'(y^*)v\rangle = 0\} \\ &= \{v : \Xi'(y^*)v \in D \cap (u^*)^\perp\} = \ker \Xi'(y^*).\end{aligned}$$

We therefore choose

$$C(y) := \Xi'(y), \qquad C^* := C(y^*),$$
$$H^* := -\nabla_y^2 \langle u^*, \Xi\rangle(y^*) = -\sum_{i=1}^{r} u_i^* \nabla_y^2 \Xi_i(y^*).$$

Then $\ker C^* = \mathcal{C}_{\mathcal{K}}(y^*)$ and H^* is symmetric. Taking $\hat{\mu}(\theta_0) = -u^*$ gives $\mu^* = C^{*T}\hat{\mu}(\theta_0)$.

Differentiating the multiplier representation at θ_0 gives

$$\begin{aligned}\mathrm{d}\mu^* &= -\left(\sum_{i=1}^{r} u_i^* \nabla_y^2 \Xi_i(y^*)\right)\mathrm{d}y^* - \Xi'(y^*)^T \mathrm{d}u \\ &= H^* \mathrm{d}y^* - C^{*T}\mathrm{d}u.\end{aligned}$$

The KKT comparison below identifies $\mathrm{d}\hat{\mu} = -\mathrm{d}u$ for the reduced-problem multiplier, yielding Equation 14.

Moreover, differentiating $C(y) = \Xi'(y)$ gives, for every $v \in \mathbb{R}^m$,

$$DC(y^*)[v]^T \hat{\mu}(\theta_0) = -\sum_{i=1}^{r} u_i^* \nabla_y^2 \Xi_i(y^*)v = H^* v.$$

then set $C_\theta^* := C(G_\theta(z^*))$, with the reference z^* held fixed. The chain rule yields

$$\mathrm{d}C_\theta^{*T}\hat{\mu}(\theta_0) = H^* \nabla_\theta G_{\theta_0}(z^*)\mathrm{d}\theta. \tag{15}$$

This construction yields a compatible pair C_θ^* and H^*, with Equation 14 and Equation 15 are the two identities needed for the KKT comparison below.

Substituting Equation 14 into the differential KKT system Equation 13 yields

$$\begin{aligned}\nabla_{zz}^2 \mathcal{L}\mathrm{d}z^* + \nabla_{z\theta}^2 f \mathrm{d}\theta - \sum_{k=1}^{l} \nu_k^* \nabla_{z\theta}^2 h_k \mathrm{d}\theta \\ - \sum_{i=1}^{m} \mu_i^* \nabla_{z\theta}^2 G_i \mathrm{d}\theta - \nabla_z h^T \mathrm{d}\nu^* \\ - \nabla_z G^T (H^* \mathrm{d}y^* + C^{*T}\mathrm{d}\hat{\mu}) \quad &= 0 \\ \nabla_z h \mathrm{d}z^* + \nabla_\theta h \mathrm{d}\theta &= 0 \\ C^*(\nabla_z G \mathrm{d}z^* + \nabla_\theta G \mathrm{d}\theta) &= 0\end{aligned} \tag{16}$$

The third equation is obtained by noting that $\mathrm{d}y^* = \nabla_z G \mathrm{d}z^* + \nabla_\theta G \mathrm{d}\theta \in \mathcal{C}_{\mathcal{K}}(y^*) = \ker C^*$.

Finally, we want to show the first-order equivalence between the reduced problem and the original problem, and $\hat{\mu}$ is indeed the dual solution of reduced Problem P2.

For reduced Problem P2, denote $\hat{\mu}$ and $\hat{\nu}$ as the dual variables for the reduced conic constraint $C_\theta^*\big((G_\theta(z^*) - y^*) + \nabla_z G_\theta(z^*)(z - z^*)\big) = 0$ and the affine constraint $h_\theta(z) = 0$, respectively. The KKT conditions are:

$$\begin{aligned}\Big(\nabla_{zz}^2 \mathcal{L}_\theta(z^*) - \nabla_z G_\theta(z^*)^T H^* \nabla_z G_\theta(z^*)\Big)\big(\hat{z}(\theta) - z^*\big) \\ + \nabla_z f_\theta(z^*) - \nabla_z h_\theta(\hat{z}(\theta))^T \hat{\nu}(\theta) \\ - \nabla_z G_\theta(z^*)^T C_\theta^{*T} \hat{\mu}(\theta) \quad &= 0 \\ h_\theta(z^*) + \nabla_z h_\theta(z^*)\big(\hat{z}(\theta) - z^*\big) &= 0 \\ C_\theta^*\Big(G_\theta(z^*) - y^* + \nabla_z G_\theta(z^*)\big(\hat{z}(\theta) - z^*\big)\Big) &= 0\end{aligned} \tag{17}$$

Under the standing assumptions, the system has the unique solution $z^* = \hat{z}(\theta_0)$, $\mu^* = C^{*T}\hat{\mu}(\theta_0)$, and $\nu^* = \hat{\nu}(\theta_0)$ at θ_0.

The differential of the KKT system Equation 17 at θ_0 is:

$$\begin{aligned}\left(\nabla^2_{zz}\mathcal{L} - \nabla_z G^T H^* \nabla_z G\right)\mathrm{d}\hat{z} + \nabla^2_{z\theta} f \mathrm{d}\theta - \sum_{k=1}^{l} \hat{\nu}_k \nabla^2_{z\theta} h_k \mathrm{d}\theta \\ - \nabla_z h^T \mathrm{d}\hat{\nu} - \sum_{i=1}^{m} (C^{*T}\hat{\mu})_i \nabla^2_{z\theta} G_i \mathrm{d}\theta \\ - \nabla_z G^T H^* \nabla_\theta G \mathrm{d}\theta \\ - \nabla_z G^T C^{*T} \mathrm{d}\hat{\mu} \qquad\qquad = 0 \\ \nabla_z h \mathrm{d}\hat{z} + \nabla_\theta h \mathrm{d}\theta = 0 \\ C^* \nabla_z G \mathrm{d}\hat{z} + C^* \nabla_\theta G \mathrm{d}\theta = 0\end{aligned} \tag{18}$$

Here the term, $\nabla_z G^T H^* \nabla_\theta G \mathrm{d}\theta$, follows from Equation 15; Note that Equation 16 is equivalent to Equation 18, which also shows $\hat{\mu}$ is indeed the dual solution. Rewriting Equation 18 in block-matrix form yields Equation 6.

□

B.1 A Concrete Example for Scalar Cone Constraints

We illustrate the decomposition of $\mathrm{d}\mu^*$ in Equation 14 for a cone locally defined at y^* by a twice continuously differentiable scalar-valued function $\phi : \mathbb{R}^m \to \mathbb{R}$ with $\phi(y^*) = 0$ and $\nabla\phi(y^*) \neq 0$: $\mathcal{K} = \{y : \phi(y) \leq 0\}$.

By strict complementarity, there exists a scalar $\alpha > 0$ such that

$$\mu^* = -\alpha \nabla_y \phi(y^*).$$

Differentiating this relation with respect to the parameters, we obtain

$$\mathrm{d}\mu^* = -\alpha \nabla^2_{yy} \phi(y^*) \mathrm{d}y^* - \mathrm{d}\alpha \nabla_y \phi(y^*).$$

Comparing with Equation 14, we can identify the components:

- $H^* = -\alpha \nabla^2_{yy} \phi(y^*)$ captures the curvature effect (how the normal direction rotates as y^* moves along the cone boundary),
- $C^* = \nabla_y \phi(y^*)^T$ provides the basis for the normal space,
- $\mathrm{d}\hat{\mu} = -\mathrm{d}\alpha$ is the coefficient for the normal component.

C Complementary Details for Common Reductions (Section 5)

C.1 Details on Second-Order Cone Programs (Section 5.3)

Proof of Lemma 5.1. For $j \in J_{\mathrm{na}}$, the function $\phi(x, t) = \|x\|_2 - t$ is smooth near y_j^*, with $\phi(y_j^*) = 0$ and $\nabla\phi(y_j^*) \neq 0$. Thus $\Xi_j = \phi$ is a local reduction to $\mathbb{R}_-$. Strict complementarity gives

$$\mu_j^* = -\alpha_j \nabla\phi(y_j^*), \qquad \alpha_j > 0.$$

Since the last component of $\nabla\phi$ is -1 and $\mu_j^* = (u_j^*, s_j^*)$, we have $\alpha_j = s_j^*$. The construction in Appendix B therefore yields

$$H_j^* = -s_j^* \nabla^2 \phi(y_j^*), \qquad C_{j,\theta}^* = \nabla\phi(G_j(z^*))^T \qquad C_j^* := C_{j,\theta_0}^* = \nabla\phi(y_j^*)^T.$$

Thus, $\ker C_j^* = \{\mathrm{d}y : \nabla\phi(y_j^*)^T \mathrm{d}y = 0\}$, which is the critical cone in Equation 7.

For $j \in J_{\mathrm{ap}}$, use the identity reduction $\Xi_j(y_j) = y_j$ to $\mathcal{K}_{\mathrm{SOC}_j}$. Its Jacobian is I and its second derivative vanishes, so the same construction gives

$$H_j^* = 0, \qquad C_{j,\theta}^* = C_j^* = I.$$

In particular, $\ker C_j^* = \{0\}$ agrees with the critical cone in Equation 7. □

C.2 Details on Semidefinite Programs (Section 5.4)

We provide the details underlying Lemma 5.2. Section 3.2 and the proof of Theorem 4.1 give

$$Z^* \in \mathcal{F}_{\mathcal{K}}(Z^*), \quad \mathrm{d}Z^* \in \mathcal{C}_{\mathcal{K}}(Z^*)$$
$$S^* \in \mathcal{F}_{\mathcal{K}^*}(S^*) = -\mathcal{N}_{\mathcal{K}}(Z^*), \quad \mathrm{d}S^* \in \mathcal{C}_{\mathcal{K}^*}(S^*)$$

Define the basis transform operator $T^* : \mathbb{S}^n \to \mathbb{S}^n$ by

$$T^*(X) = U^{*T} X U^* = \begin{bmatrix} T^*(X)_{00} & T^*(X)_{01} \\ T^*(X)_{01}^T & T^*(X)_{11} \end{bmatrix},$$

where $T^*(X)_{00} \in \mathbb{R}^{|J|\times|J|}$, $T^*(X)_{01} \in \mathbb{R}^{|J|\times(n-|J|)}$, and $T^*(X)_{11} \in \mathbb{R}^{(n-|J|)\times(n-|J|)}$. Note that U^* is the eigenbasis of Z^*, which is fixed and does not depend on the perturbation, thus we can treat U^* as a constant matrix and ignore its differential when deriving the reduced problem.

Now we introduce the cone geometries at the optimal solution.

Lemma C.1.

$$S^{*\perp} = \{Z \in \mathbb{S}_n : \langle T^*(Z)_{00}, T^*(S^*)_{00} \rangle = 0\} \tag{19}$$

Proof of Lemma C.1.

$$\begin{aligned} S^{*\perp} &= \{Z \in \mathbb{S}_n : \langle Z, S^* \rangle = 0\} \\ &= \{Z \in \mathbb{S}_n : \langle T^*(Z), T^*(S^*) \rangle = 0\} \\ &= \{Z \in \mathbb{S}_n : \left\langle \begin{bmatrix} T^*(Z)_{00} & T^*(Z)_{01} \\ T^*(Z)_{01}^T & T^*(Z)_{11} \end{bmatrix}, \begin{bmatrix} T^*(S^*)_{00} & 0 \\ 0 & 0 \end{bmatrix} \right\rangle = 0\} \\ &= \{Z \in \mathbb{S}_n : \langle T^*(Z)_{00}, T^*(S^*)_{00} \rangle = 0\} \end{aligned}$$

□

Lemma C.2. *The active face of Z^*, $\mathcal{F}_{\mathcal{K}}(Z^*) = \mathcal{K} \cap S^{*\perp}$, is equivalent to*

$$\mathcal{F}_{\mathcal{K}}(Z^*) = \{Z \in \mathbb{S}_n^+ : Z \succeq 0, \langle S^*, Z \rangle = 0\} = \{Z : T^*(Z) = \begin{bmatrix} 0 & 0 \\ 0 & T^*(Z)_{11} \end{bmatrix}, T^*(Z)_{11} \succeq 0\} \tag{20}$$

Proof of Lemma C.2. Under strict complementarity, $T^*(S^*)_{00} \succ 0$. Since $T^*(Z)_{00} \succeq 0$, the identity $\langle T^*(Z)_{00}, T^*(S^*)_{00} \rangle = 0$ implies that $T^*(Z)_{00} = 0$. The positive semidefiniteness of $T^*(Z)$ then yields $T^*(Z)_{01} = 0$. □

Lemma C.3. *The tangent cone of $\mathcal{K}$ at Z^* is equivalent to*

$$\mathcal{T}_{\mathcal{K}}(Z^*) = \{\mathrm{d}Z \in \mathbb{S}_n : T^*(\mathrm{d}Z)_{00} \succeq 0\} \tag{21}$$

Proof of Lemma C.3. We first prove necessity. If $\mathrm{d}Z \in \mathcal{T}_{\mathcal{K}}(Z^*)$, then there exist $\alpha_k \geq 0$ and $Z_k \succeq 0$ such that $\alpha_k(Z_k - Z^*) \to \mathrm{d}Z$. Since $T^*(Z^*)_{00} = 0$, the 00 principal block of $\alpha_k T^*(Z_k - Z^*)$ is $\alpha_k T^*(Z_k)_{00} \succeq 0$. Taking the limit gives $T^*(\mathrm{d}Z)_{00} \succeq 0$.

Conversely, suppose that $T^*(\mathrm{d}Z)_{00} \succeq 0$. For any $\epsilon > 0$, define $\mathrm{d}Z_\epsilon$ by

$$T^*(\mathrm{d}Z_\epsilon) = T^*(\mathrm{d}Z) + \begin{bmatrix} \epsilon I & 0 \\ 0 & 0 \end{bmatrix}. \tag{22}$$

For sufficiently large $\alpha > 0$, $T^*(Z^*)_{11} + \alpha^{-1} T^*(\mathrm{d}Z)_{11} \succ 0$. Moreover, since $T^*(\mathrm{d}Z)_{00} + \epsilon I \succ 0$, the corresponding Schur complement satisfies

$$\alpha^{-1}\big(T^*(\mathrm{d}Z)_{00} + \epsilon I\big) - \alpha^{-2} T^*(\mathrm{d}Z)_{01}\big(T^*(Z^*)_{11} + \alpha^{-1} T^*(\mathrm{d}Z)_{11}\big)^{-1} T^*(\mathrm{d}Z)_{01}^T \succeq 0 \tag{23}$$

for all sufficiently large $\alpha > 0$. Therefore,

$$T^*(Z^* + \alpha^{-1}\mathrm{d}Z_\epsilon) = \begin{bmatrix} \alpha^{-1}\big(T^*(\mathrm{d}Z)_{00} + \epsilon I\big) & \alpha^{-1} T^*(\mathrm{d}Z)_{01} \\ \alpha^{-1} T^*(\mathrm{d}Z)_{01}^T & T^*(Z^*)_{11} + \alpha^{-1} T^*(\mathrm{d}Z)_{11} \end{bmatrix} \succeq 0. \tag{24}$$

Setting $Z_\epsilon := Z^* + \alpha^{-1}\mathrm{d}Z_\epsilon \succeq 0$ gives $\mathrm{d}Z_\epsilon = \alpha(Z_\epsilon - Z^*)$, and hence $\mathrm{d}Z_\epsilon \in \mathcal{T}_\mathcal{K}(Z^*)$. Since $\mathrm{d}Z_\epsilon \to \mathrm{d}Z$ as $\epsilon \downarrow 0$ and the tangent cone is closed, $\mathrm{d}Z \in \mathcal{T}_\mathcal{K}(Z^*)$.

Thus,

$$\mathcal{T}_\mathcal{K}(Z^*) = \{\mathrm{d}Z \in \mathbb{S}_n : T^*(\mathrm{d}Z)_{00} \succeq 0\}. \tag{25}$$

□

The conditions for the dual variable S^* are similar. We have

Lemma C.4.

$$Z^{*\perp} = \{S \in \mathbb{S}_n : \langle T^*(S)_{11}, T^*(Z^*)_{11}\rangle = 0\} \tag{26}$$

Lemma C.5. *The active face of S^*, $\mathcal{F}_{\mathcal{K}^*}(S^*) = \mathcal{K}^* \cap Z^{*\perp} = -\mathcal{N}_\mathcal{K}(Z^*)$, is equivalent to*

$$\mathcal{F}_{\mathcal{K}^*}(S^*) = -\mathcal{N}_\mathcal{K}(Z^*) = \{S \in \mathbb{S}_n : S \succeq 0, SZ^* = 0\} = \{S : T^*(S) = \begin{bmatrix} T^*(S)_{00} & 0 \\ 0 & 0 \end{bmatrix}, T^*(S)_{00} \succeq 0\} \tag{27}$$

Lemma C.6. *The tangent cone of $\mathcal{K}^*$ at S^* is*

$$\mathcal{T}_{\mathcal{K}^*}(S^*) = \{\mathrm{d}S \in \mathbb{S}_n : T^*(\mathrm{d}S)_{11} \succeq 0\} \tag{28}$$

Lemma C.7. *The critical cone, $\mathcal{C}_{\mathcal{K}^*}(S^*) = \mathcal{T}_{\mathcal{K}^*}(S^*) \cap Z^{*\perp}$, is equivalent to*

$$\mathcal{C}_{\mathcal{K}^*}(S^*) = \{\mathrm{d}S \in \mathbb{S}_n : T^*(\mathrm{d}S)_{11} = 0\} \tag{29}$$

With these cone-geometric characterizations in place, we are ready to prove Lemma 5.2.

Proof of Lemma 5.2. In this problem, we have $G_\theta(Z) = G_{\theta_0}(Z) = Z$. Thus we omit the subscript θ in the following derivations.

Differentiating the complementarity condition $S^*Z^* = 0$ gives

$$\mathrm{d}S^*Z^* + S^*\mathrm{d}Z^* = 0 \tag{30}$$

This is equivalent to

$$T^*(\mathrm{d}S^*)T^*(Z^*) + T^*(S^*)T^*(\mathrm{d}Z^*) = 0 \tag{31}$$

Note that this step only multiplies by the fixed matrix U^*, so $\mathrm{d}U^*$ does not appear in the following derivations.

Substituting the cone structures $\mathrm{d}S^* \in \mathcal{C}_{\mathcal{K}^*}(S^*)$, $Z^* \in \mathcal{F}_\mathcal{K}(Z^*)$, $S^* \in \mathcal{F}_{\mathcal{K}^*}(S^*)$, and $\mathrm{d}Z^* \in \mathcal{C}_\mathcal{K}(Z^*)$, given by Lemmas C.7, C.2, and C.5 and by Equation 8, into Equation 31 yields

$$\begin{bmatrix} T^*(\mathrm{d}S^*)_{00} & T^*(\mathrm{d}S^*)_{01} \\ T^*(\mathrm{d}S^*)_{01}^T & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & T^*(Z^*)_{11} \end{bmatrix} + \begin{bmatrix} T^*(S^*)_{00} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & T^*(\mathrm{d}Z^*)_{01} \\ T^*(\mathrm{d}Z^*)_{01}^T & T^*(\mathrm{d}Z^*)_{11} \end{bmatrix} = 0 \tag{32}$$

$$\Leftrightarrow \begin{bmatrix} 0 & T^*(\mathrm{d}S^*)_{01}T^*(Z^*)_{11} + T^*(S^*)_{00}T^*(\mathrm{d}Z^*)_{01} \\ 0 & 0 \end{bmatrix} = 0 \tag{33}$$

Thus, $T^*(\mathrm{d}S^*)_{01} = -T^*(S^*)_{00}T^*(\mathrm{d}Z^*)_{01}T^*(Z^*)_{11}^{-1}$, and $\mathrm{d}S^*$ can be represented as

$$\mathrm{d}S^* = U^* \begin{bmatrix} 0 & T^*(\mathrm{d}S^*)_{01} \\ T^*(\mathrm{d}S^*)_{01}^T & 0 \end{bmatrix} U^{*T} + U^* \begin{bmatrix} T^*(\mathrm{d}S^*)_{00} & 0 \\ 0 & 0 \end{bmatrix} U^{*T} \tag{34}$$

After some rewriting, we have

$$\mathrm{d}S^* = -\left(Z^{*\dagger}\mathrm{d}Z^*S^* + S^*\mathrm{d}Z^*Z^{*\dagger}\right) + \sum_{j,j' \in J} T^*(\mathrm{d}S^*)_{00,jj'}u_j^*u_{j'}^{*T} \tag{35}$$

The first term describes the change in S^* along the critical cone $\mathcal{C}_\mathcal{K}(Z^*)$, whereas the second lies in $\mathcal{C}_\mathcal{K}(Z^*)^\perp$. Vectorizing Equation 35 gives

$$\begin{aligned}\mathrm{vec}(\mathrm{d}S^*) &= -(Z^{*\dagger} \otimes S^* + S^* \otimes Z^{*\dagger})\mathrm{vec}(\mathrm{d}Z^*) + (U_0^* \otimes U_0^*)\mathrm{vec}(T^*(\mathrm{d}S^*)_{00}) \\ &= -(Z^{*\dagger} \otimes S^* + S^* \otimes Z^{*\dagger})\mathrm{vec}(\mathrm{d}Z^*) + \sum_{j,j' \in J} T^*(\mathrm{d}S^*)_{00,jj'}\mathrm{vec}(u_j^* u_{j'}^{*T}) \end{aligned} \tag{36}$$

From the first term, we have $H^* = -(Z^{*\dagger} \otimes S^* + S^* \otimes Z^{*\dagger})$.

Since $T^*(\mathrm{d}S^*)_{00}$ is symmetric, grouping the symmetric terms in the second term of Equation 35 gives C^* with rows $\mathrm{vec}(B_{jj'})^T$ for $j, j' \in J$ with $j' \le j$, where

$$B_{jj} = u_j^* u_j^{*T}, \qquad B_{jj'} = u_j^* u_{j'}^{*T} + u_{j'}^* u_j^{*T}, \quad j' < j. \tag{37}$$

The corresponding coefficients form $\mathrm{d}\hat{\mu}$. □

Example: multiple zero eigenvalues and the critical cone. Consider $Z^* = \mathrm{diag}(0, 0, 1) \in \mathbb{S}^3_+$ and the perturbations $\delta Z_1 = \epsilon e_1 e_1^T$ and $\delta Z_2 = \epsilon e_2 e_2^T$, where $\epsilon > 0$ and e_1, e_2 are the standard basis vectors associated with the two zero eigenvalues. Although $Z^* + \delta Z_1$ and $Z^* + \delta Z_2$ have the same eigenvalues $(0, \epsilon, 1)$, their normal cones are

$$\begin{aligned}\mathcal{N}_{\mathbb{S}^3_+}(Z^* + \delta Z_1) &= \{-\alpha e_2 e_2^T : \alpha \ge 0\}, \\ \mathcal{N}_{\mathbb{S}^3_+}(Z^* + \delta Z_2) &= \{-\alpha e_1 e_1^T : \alpha \ge 0\}.\end{aligned}$$

Thus, arbitrarily small perturbations within the zero-eigenspace can select different normal directions, precluding a single smooth local representation of the normal-cone mapping along these directions.

Under strict complementarity, however, the critical-cone condition $U_0^{*T} \delta Z U_0^* = 0$, with $U_0^* = [e_1, e_2]$, restricts admissible perturbations to the form

$$\delta Z = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ a & b & c \end{bmatrix}.$$

It therefore excludes all perturbations within the zero-eigenspace, including δZ_1 and δZ_2, while retaining cross-eigenspace and positive-eigenspace directions. In these admissible directions, the curvature term used by the reduced problem is well defined.

D Additional Experimental Details

Experiments in this paper were run on an M2 MacBook Air with 8 CPU cores and 16GB RAM.

We corrected a variable-name error in the latest publicly available FFOLayer implementation at the time of our experiments (commit `28905f3e1750fca5b8918954d5d2ea5bed0cbacc`). Before this one-line fix, its gradients differed substantially from those produced by both dOPT and CVXPYLayers; after the fix, all three methods are numerically consistent. All reported *FFOLayer (Modified)* results use the corrected code, which is included in our repository.

D.1 Details for Synthetic Experiments (Section 6.1)

For each problem class and dimension, we evaluate ten randomly generated instances and report the mean of each metric.

Random SOCPs. We generate problems of the form

$$\min_z \; q^\top z \quad \text{s.t.} \quad \|A_j z + b_j\|_2 \le c_j^\top z + d_j, \quad Fz = g, \quad Gz \le h. \tag{38}$$

For a decision dimension n, we use $\lfloor n/2 \rfloor$ second-order cones, each with vector dimension $\lfloor n/3 \rfloor$, together with two equalities and ten linear inequalities. We draw q, A_j, b_j, c_j, F, and G independently from a standard normal distribution. After drawing $z_0 \sim \mathcal{N}(0, I)$, we set $d_j = \|A_j z_0 + b_j\|_2 - c_j^\top z_0$, $g = Fz_0$, and $h = Gz_0 + \mathbf{1}$, ensuring that every generated instance is feasible.

Random SDPs. We generate

$$\min_{Z \succeq 0} \langle C, Z \rangle \quad \text{s.t.} \quad A \operatorname{vec}(Z) = b, \tag{39}$$

where $Z \in \mathbb{S}^n$ and $m = \lfloor (n(n+1)/2 - 1)/2 \rfloor$ equality constraints are used. We construct $C = Q \Lambda Q^\top$, where Q is obtained from the QR factorization of a Gaussian matrix and the diagonal entries of Λ are logarithmically spaced from 1 to 10. The entries of $A \in \mathbb{R}^{m \times n^2}$ are drawn independently from a standard normal distribution. Finally, we construct a positive-definite matrix $Z_0 = U \Lambda_0 U^\top$ in the same way, with eigenvalues logarithmically spaced from 1 to $10^{1.5}$, and set $b = A \operatorname{vec}(Z_0)$ to ensure feasibility.

D.2 Details for vehicle platoon benchmark (Section 6.2)

Benchmark. We adapt the networked cooperative vehicle-platoon benchmark of Makhlouf & Kowalewski (2014); Chen et al. (2015). A platoon with N controlled vehicles has state

$$x = (e_1, \dot{e}_1, a_1, \ldots, e_N, \dot{e}_N, a_N) \in \mathbb{R}^{3N}, \tag{40}$$

where e_j, $\dot{e}_j$, and a_j are the spacing error, relative velocity, and acceleration of vehicle j, respectively. We use the $N \in \{3, 5, 10\}$ instances, giving state dimensions 9, 15, and 30. We index the two modes consistently as 0 (normal communication) and 1 (total communication loss):

$$\dot{x} = A^{\mathrm{ct}}_{\sigma(t)} x + B^{\mathrm{ct}}_L a_L(t), \qquad \sigma(t) \in \{0, 1\}, \tag{41}$$

where a_L is the acceleration command of the manually driven leader. The leader is external to the $3N$-dimensional state, and its input acts through B^{ct}_L. Both $a_L(t)$ and B^{ct}_L are treated as known; they are not inferred by the learning model.

For the three-vehicle instance, we use the two 9×9 continuous-time matrices printed in the original benchmark. The five- and ten-vehicle instances provide only the connected dynamics, so we extend the communication-loss pattern of the three-vehicle instance.

We convert each continuous-time pair $(A^{\mathrm{ct}}_i, B^{\mathrm{ct}}_L)$ to the discrete-time model

$$x_{k+1} = A_i x_k + B_i u_k \tag{42}$$

by exact zero-order-hold discretization with sampling period $\Delta t = 0.1$.

The input matrices B_i are treated as known physical quantities throughout; the sampled input u_k is also observed. Only the two mode matrices A_0, A_1 and the switching time are learned.

Trajectory data. We generate 16 trajectories of length $T = 30$ from the same ground-truth switching sequence. Initial states are sampled independently as $x_0 \sim 0.35\mathcal{N}(0, I)$. To model a leader acceleration command rather than white-noise actuation, u_k is piecewise constant: every five time steps a new value is sampled uniformly from $[-2, 1]$. The ground-truth system uses mode 0 for transitions $k < 14$ and mode 1 for $k \geq 14$. We add independent Gaussian observation noise to every state coordinate, with standard deviation equal to 1% of that coordinate's empirical standard deviation over the clean training trajectories. Ground-truth mode labels are retained only for evaluation and are not exposed to initialization or training.

Initialization. We first fit a single pooled dynamics matrix using every observed transition

$$\bar{A} = \arg\min_A \sum_{j,k} \left\| x^{(j)}_{k+1} - A x^{(j)}_k - \bar{B} u^{(j)}_k \right\|_2^2 + \lambda_{\mathrm{ridge}} \|A\|_F^2, \qquad \lambda_{\mathrm{ridge}} = 10^{-5}, \tag{43}$$

where $\bar{B} = (B_0 + B_1)/2$. To avoid ill-conditioned SDP, we break the otherwise exact mode symmetry using

$$A^{(0)}_0 = \bar{A} - \Delta, \qquad A^{(0)}_1 = \bar{A} + \Delta, \qquad \Delta = 10^{-4} \|\bar{A}\|_F \frac{R}{\|R\|_F}, \tag{44}$$

with a dense Gaussian matrix R. The switch time is initialized to $\tau^{(0)} = 20.3$.

Table D.1: Performance on the adapted vehicle-platoon benchmark for $N \in \{3, 5, 10\}$ vehicles, with state dimension $d = 3N$. Training entries are total times for all epochs.

Method	Metric			
	Number of vehicles N	3	5	10
	State dimension d	9	15	30
dOPT	Stability margin $t^\star$	8.4×10^{-3}	6.3×10^{-3}	3.4×10^{-3}
	Forward [ms]	**12.7**	41.4	468.4
	Backward [ms]	**1.4**	**3.6**	**26.1**
	Training [s]	**4.3**	**12.0**	**128.8**
FFOLayer (Mod)	Stability margin $t^\star$	8.5×10^{-3}	6.1×10^{-3}	3.4×10^{-3}
	Forward [ms]	14.3	**41.0**	**459.0**
	Backward [ms]	16.2	43.6	472.7
	Training [s]	8.0	21.4	241.9
FFOLayer (Lifted)	Stability margin $t^\star$	8.3×10^{-3}	6.3×10^{-3}	3.4×10^{-3}
	Forward [ms]	28.1	84.3	797.4
	Backward [ms]	30.5	88.9	815.1
	Training [s]	15.1	44.2	405.9
CVXPYLayers	Stability margin $t^\star$	8.5×10^{-3}	6.3×10^{-3}	3.4×10^{-3}
	Forward [ms]	93.3	345.3	5592.1
	Backward [ms]	6.6	84.3	4948.6
	Training [s]	28.6	115.2	13047.9

Model architecture and training. The learning architecture contains a single differentiable SDP layer, and A_0 and A_1 are trainable parameters. For the switching time, we use a soft-to-hard switching-time schedule. In the numerical formulation, we enforce $P \succeq 10^{-3}I$ and choose $\alpha = 0.015d$. Specifically, we parameterize the switching time by an unconstrained scalar s, mapped to the valid range as

$$\tau = (T-1)\,\mathrm{sigmoid}(s). \tag{45}$$

During the first 50 epochs, we use the differentiable gate

$$q_k = \mathrm{sigmoid}\left(\frac{k-\tau}{\gamma}\right), \qquad \widetilde{A}_k = (1-q_k)A_0 + q_k A_1, \qquad \widetilde{B}_k = (1-q_k)B_0 + q_k B_1. \tag{46}$$

Here $q_k \in (0,1)$ is the soft probability of using mode 1 at transition k. Consequently, $\widetilde{A}_k$ and $\widetilde{B}_k$ are the effective time-dependent dynamics used only to propagate the soft-switch model; they are convex combinations of the two modes, not additional learned matrices. Starting from the observed initial state, the soft rollout is defined recursively by

$$\widehat{x}_0 = x_0, \qquad \widehat{x}_{k+1} = \widetilde{A}_k\widehat{x}_k + \widetilde{B}_k u_k, \tag{47}$$

and its fitting loss is the MSE between the complete predicted and observed trajectories, $\mathcal{L}_{\mathrm{roll}} = \mathrm{MSE}((\widehat{x}_k)_{k=0}^{T-1}, (x_k)_{k=0}^{T-1})$. The temperature γ is geometrically annealed from 3.0 to 0.25. This soft stage permits gradients to identify τ jointly with the two mode matrices. We use Adam with learning rate 10^{-3} for (A_0, A_1) and 5×10^{-2} for s.

After epoch 50, we harden the switching sequence to $\sigma(k) = \mathbb{I}[k \geq \tau]$, freeze τ, and optimize only A_0, A_1 for another 200 epochs. Rather than recursively feeding back $\widehat{x}_k$, the hard-stage one-step loss predicts every next state directly from the corresponding observed current state:

$$\mathcal{L}_{\mathrm{1step}} = \frac{1}{N_{\mathrm{traj}}(T-1)} \sum_{j,k} \left\| x_{k+1}^{(j)} - A_{\sigma(k)} x_k^{(j)} - B_{\sigma(k)} u_k^{(j)} \right\|_2^2. \tag{48}$$

This teacher-forced objective isolates transition identification from the compounding error of a long recursive rollout. The hard stage uses Adam with learning rate 3×10^{-3}. Thus the discontinuity near epoch 50 in Figure 6.1 (Right) is expected: it coincides with both the soft-to-hard change in the switching model and the change from $\mathcal{L}_{\mathrm{roll}}$ to $\mathcal{L}_{\mathrm{1step}}$.

E Gradient Accuracy

We validate gradient accuracy on the random SOCP and SDP instances described in Section 6.1 and Appendix D. We first compare the gradients produced by dOPT and CVXPYLayers. For each

method, we vectorize the gradients with respect to all problem parameters and concatenate them into a single vector. We report the relative error

$$\frac{\|g_{\text{candidate}} - g_{\text{reference}}\|_2}{\max\{\|g_{\text{reference}}\|_2, 10^{-12}\}}. \tag{49}$$

In each "A vs. B" comparison, A is the candidate and B is the reference.

We additionally verify the gradient using the envelope theorem (Afriat, 1971). Define the value function as $V(\theta) := f_\theta(z^*(\theta))$, and let $\mathcal{L}_\theta(z, \mu, \nu)$ denote the Lagrangian. The theorem establishes an equivalence between the derivative of the value function and the partial derivative of the Lagrangian evaluated at the optimal primal–dual solution:

$$\nabla_\theta V(\theta) = \partial_\theta \mathcal{L}_\theta(z, \mu, \nu)|_{(z,\mu,\nu)=(z^*(\theta),\mu^*(\theta),\nu^*(\theta))}. \tag{50}$$

Here, ∂_θ differentiates only the explicit dependence on θ, holding (z, μ, ν) fixed. The right-hand side is therefore an analytic gradient, which we use as ground truth (GT). Each entry below is the mean relative error over ten random instances, using the same forward solution across methods.

Table E.1: Gradient relative errors on random SOCPs.

Gradient comparison	$n = 20$	$n = 100$	$n = 500$
$\nabla_\theta \mathbf{1}^\top z^*(\theta)$: dOPT vs. CVXPYLayers	1.38×10^{-2}	3.24×10^{-3}	1.60×10^{-1}
$\nabla_\theta V(\theta)$: dOPT vs. GT	3.67×10^{-9}	5.81×10^{-9}	4.10×10^{-9}
$\nabla_\theta V(\theta)$: CVXPYLayers vs. GT	6.88×10^{-4}	2.55×10^{-4}	1.50×10^{-3}

Table E.2: Gradient relative errors on random SDPs.

Gradient comparison	$n = 20$	$n = 50$	$n = 100$
$\nabla_\theta \mathbf{1}^\top z^*(\theta)$: dOPT vs. CVXPYLayers	2.24×10^{-2}	8.21×10^{-3}	2.49×10^{-2}
$\nabla_\theta V(\theta)$: dOPT vs. GT	1.09×10^{-7}	1.22×10^{-7}	1.03×10^{-7}
$\nabla_\theta V(\theta)$: CVXPYLayers vs. GT	5.89×10^{-3}	4.51×10^{-3}	1.11×10^{-2}

The discrepancy between dOPT and CVXPYLayers remains modest, although it is larger at the highest tested dimension. For $\nabla_\theta V(\theta)$, dOPT agrees consistently with the analytic GT, while the CVXPYLayers error becomes more pronounced at larger dimensions.