DUAL CERTIFIED WHITE-BOX INFERENCE FOR INPUT CONVEX NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Input convex neural networks (ICNNs) are used to learn convex objectives whose minimizers define decisions, making efficient and reliable optimization central to inference. At nonsmooth inputs, automatic differentiation returns a single derivative rather than the full subdifferential governing optimality and descent. Second-order cone ICNNs (SOC-ICNNs) admit an exact representation as value functions of parametric second-order cone programs, providing a white-box approach to recovering their full subdifferentials from optimal dual multipliers and deriving explicit Hessians on smooth regions. Building on this representation, we develop dual-certified inference (DCI), which combines the network and feasible set geometries to obtain exact stationarity certificates and tangent common descent directions. DCI uses local curvature for Newton acceleration and an exact proximal safeguard. We establish global convergence and, under standard regularity conditions, local quadratic convergence near structurally nondegenerate interior minimizers. Numerical experiments validate the recovered geometry and demonstrate the reliability and efficiency of DCI. Code is avaliable at `https://anonymous.4open.science/r/DCI-ICNN-507D/`.

1 INTRODUCTION

Input Convex Neural Networks (ICNNs) (Amos et al., 2017) parameterize functions that are convex in designated inputs and are therefore widely used to learn objectives, energies, and value functions. Applications include optimal transport (Makkuva et al., 2020), system identification (Bünning et al., 2021), and control with learned models (Ławryńczuk, 2022). In optimization-based applications, inference is not merely a forward evaluation. Given a context y, we define the inference objective $F_y(x) := f_\theta(x; y) + \Phi(x; y)$, where $x \in \mathbb{R}^d$ is the decision variable, θ denotes the network parameters, and $\Phi(\cdot; y)$ is an additional term that is finite, convex, and continuously differentiable in x. With y and θ held fixed, the final decision is obtained by solving

$$x^\star(y) \in \arg\min_{x \in \mathcal{X}} F_y(x), \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a nonempty closed convex feasible set. The reliability and deployment efficiency of an ICNN therefore depend on both its approximation quality and our ability to minimize the learned objective quickly and certify the resulting decision.

The main obstacle is the derivative information available at nonsmooth points. Consider the convex function $F(x) = \max\{x, -2x\}$. Its global minimizer is $x = 0$, since $\partial F(0) = [-2, 1]$ contains the origin. Automatic differentiation (AD) may, however, follow the branch $F(x) = x$ and return the gradient $g_{\mathrm{AD}} = 1$. This nonzero derivative incorrectly suggests that the current point is nonstationary. Moreover, its negative is not a descent direction: $F'(0; -1) = 2 > 0$. Thus, the derivative selected from one executed branch can fail both as an optimality certificate and as a search direction.

The same issue appears in constrained ICNN inference. At a ReLU boundary, or when a Euclidean norm residual vanishes, multiple supporting slopes may be valid at the same input. AD returns a derivative selected by the executed computational graph (Bolte & Pauwels, 2020), whereas optimality depends jointly on the complete network subdifferential and the constraint geometry (Rockafellar & Wets, 1998):

$$0 \in \partial F_y(x) + N_{\mathcal{X}}(x) \iff x \in \arg\min_{z \in \mathcal{X}} F_y(z), \quad F_y'(x; d) = \max_{g \in \partial F_y(x)} g^\top d. \tag{2}$$

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.

where $N_{\mathcal{X}}(x)$ denotes the normal cone of the feasible set $\mathcal{X}$ at x. A constrained search direction must also belong to the tangent cone $T_{\mathcal{X}}(x)$ to satisfy first-order feasibility.

Recovering the complete geometry is nontrivial for a deep network. A single boundary unit may admit a continuum of valid slopes, and the admissible choices are coupled recursively across layers. Enumerating activation branches is therefore combinatorial, while general nonsmooth methods address nonsmoothness through gradient sampling (Burke et al., 2005), bundle models (Kiwiel, 1990), or proximal constructions (Rockafellar, 1976). The challenge is richer for second-order cone ICNNs (SOC-ICNNs) (Liu et al., 2026), which extend conventional ICNNs with quadratic and conic modules and represent a broader class of convex functions. Thus the central question is: *Can we recover and exploit the network's subdifferential and curvature information for white-box inference?*

We exploit the exact representation of SOC-ICNNs as value functions of parametric second-order cone programs (SOCPs). From the network's forward states, we characterize the complete set of optimal dual multipliers and prove that their associated support slopes form the network subdifferential. The internal conic dual structure of the network therefore provides exact directional derivatives and explicit local curvature on smooth branches. This representation supplies the geometry needed for certification and descent, together with the curvature required for Newton acceleration. Moreover, the set-valued computation is restricted to multiplier blocks associated with active structural boundaries, so its size is determined by only the local boundary structure.

Building on this representation, we propose *dual-certified inference* (DCI) for fast and reliable SOC-ICNN deployment. DCI projects the origin onto $\partial F_y(x) + N_{\mathcal{X}}(x)$ under a positive definite metric. The projection residual vanishes exactly at a constrained global minimizer. Otherwise, it determines a direction in the tangent cone that decreases every active affine support of F_y. On unconstrained smooth regions, using the explicit Hessian as the metric reduces this direction to a Newton step. We combine the local step with an exact proximal safeguard and establish global convergence, together with local quadratic convergence near structurally nondegenerate interior minimizers under standard regularity conditions. Numerical experiments verify the recovered geometry and demonstrate the reliability and efficiency of DCI.

Our contributions are threefold:

(1) We characterize the complete set of optimal dual multipliers of an SOC-ICNN and prove that its readout exactly recovers the full network subdifferential and all directional derivatives. The same structure also yields explicit curvature on smooth branches.

(2) We develop DCI, which combines the network subdifferential with the geometry of the feasible set to provide exact stationarity certificates and common descent directions in the tangent cone. Its additional set-valued computation is restricted to the active structural boundaries.

(3) We establish global convergence of safeguarded DCI and local quadratic convergence under standard smoothness and regularity conditions. Numerical experiments validate that the recovered white-box geometry enables reliable inference, with substantial speedups on applications.

2 Problem Statement

SOC-ICNN model. For a fixed context y, write $f(x) := f_\theta(x; y)$, where $x \in \mathbb{R}^d$ is the decision input and θ collects the trained parameters. A SOC-ICNN combines a ReLU-ICNN backbone with parallel quadratic and norm modules (Liu et al., 2026). The backbone computes $a_\ell(x) = W_\ell x + U_\ell z_{\ell-1}(x) + b_\ell$, $z_\ell(x) = \mathrm{ReLU}(a_\ell(x))$, for the layer index $\ell = 1, \ldots, L$, $z_0 \equiv 0$. Here a_ℓ and z_ℓ are the preactivation and hidden activation vectors. The matrices W_ℓ and U_ℓ map the original input and the previous hidden activation to layer ℓ, respectively, and b_ℓ is its bias.

The parallel modules use affine residuals $q_h(x) := B_h x + e_h \in \mathbb{R}^{m_h}$ and $u_g(x) := A_g x + d_g \in \mathbb{R}^{k_g}$, where B_h, A_g are module matrices, e_h, d_g are offsets, and m_h, k_g are residual dimensions. With H quadratic modules and G norm modules, the output is

$$f(x) = c^\top z_L(x) + v^\top x + b_0 + \sum_{h=1}^{H} \frac{\alpha_h}{2} \|q_h(x)\|_2^2 + \sum_{g=1}^{G} \lambda_g \|u_g(x)\|_2, \tag{3}$$

where $c \geq 0$ weights the final hidden activation, $v^\top x + b_0$ is an affine output term, and $\alpha_h > 0$, $\lambda_g \geq 0$ are module weights. Together with the coordinatewise restriction $U_\ell \geq 0$, these conditions preserve convexity in x. Standard ReLU ICNNs are recovered when $H = G = 0$.

A forward pass provides $f(x)$ and the internal states $\{a_\ell(x), z_\ell(x)\}_{\ell=1}^{L}$, $\{q_h(x)\}_{h=1}^{H}$, and $\{u_g(x)\}_{g=1}^{G}$, which, together with the fixed parameters, determine the subdifferential and local curvature studied below. Architecture details and geometric proofs are given in Appendix B.

Problem setting and local geometry. We consider the inference problem in equation 1. Fixing the context y, we write $f(x) := f_\theta(x; y)$, $\Phi(x) := \Phi(x; y)$, $F(x) := f(x) + \Phi(x)$. For $M \succ 0$, we write $\|v\|_M := \sqrt{v^\top M v}$. We study the local first- and second-order geometry of the trained SOC-ICNN and use it to solve $\min_{x \in \mathcal{X}} F(x)$.

For $f(x)$, the complete first-order geometry at x is characterized by its subdifferential $\partial f(x) = \{g \in \mathbb{R}^d : f(x') \geq f(x) + g^\top (x' - x),\ \forall x'\}$. Since Φ is finite, convex, and continuously differentiable, standard subdifferential calculus (Rockafellar & Wets, 1998) gives $\partial F(x) = \partial f(x) + \nabla \Phi(x)$, $F'(x; d) = \max_{g \in \partial F(x)} g^\top d$, and constrained optimality satisfies equation 2. Thus, $\partial f(x)$ supplies the network side first-order information needed for certification and descent construction.

Second-order information is available on smooth network branches. The following condition guarantees such local smoothness.

Definition 1 (Structural nondegeneracy)**.** An input x is structurally nondegenerate if $a_{\ell,i}(x) \neq 0 \ \forall\, (\ell, i),\ u_g(x) \neq 0 \ \forall\, g$ with $\lambda_g > 0$.

At a structurally nondegenerate input, the ReLU activation pattern is locally fixed and every positive-weight norm module is smooth; hence f is twice continuously differentiable in a neighborhood of x, with local gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$. At structural boundaries, $\partial f(x)$ may contain multiple subgradients. Accordingly, we recover $\partial f(x)$ at arbitrary inputs and combine it with $\nabla \Phi(x)$ and $N_{\mathcal{X}}(x)$ for certification and descent. On structurally nondegenerate regions, explicit network curvature also supports the construction of the DCI metric.

3 Exact First- and Second-Order Geometry of SOC-ICNNs

Given a trained SOC-ICNN and an input x, we characterize the first- and second-order geometry of its scalar output f. At arbitrary inputs, we recover the complete subdifferential $\partial f(x)$; on structurally nondegenerate regions, we further obtain explicit gradient and Hessian information.

3.1 Structured dual representation

By Lemma 10, $f(x)$ is the value of a parametric SOCP, with an optimal primal solution supplied by the forward pass. Its reduced dual variables are ν, p, and r for the ReLU, quadratic, and norm modules, respectively. For the ReLU backbone, define $\mathcal{K}_\nu := \{\nu = \{\nu_\ell\}_{\ell=1}^{L} : 0 \leq \nu_L \leq c,\ 0 \leq \nu_\ell \leq U_{\ell+1}^\top \nu_{\ell+1},\ \ell < L\}$, where all inequalities are coordinatewise. The quadratic variables satisfy $p \in \mathcal{P} := \prod_{h=1}^{H} \mathbb{R}^{m_h}$, while the norm variables satisfy $\mathcal{K}_r := \prod_{g=1}^{G} \{r_g \in \mathbb{R}^{k_g} : \|r_g\|_2 \leq \lambda_g\}$. Every included quadratic module has $\alpha_h > 0$. For $\xi = (\nu, p, r) \in \mathcal{K}_\nu \times \mathcal{P} \times \mathcal{K}_r$, define

$$\Psi(x; \xi) := v^\top x + b_0 + \sum_{\ell=1}^{L} \nu_\ell^\top (W_\ell x + b_\ell) + \sum_{h=1}^{H} \left[p_h^\top q_h(x) - \frac{1}{2\alpha_h} \|p_h\|_2^2 \right] + \sum_{g=1}^{G} r_g^\top u_g(x),$$

with input-space slope $\mathcal{G}(\xi) := v + \sum_{\ell=1}^{L} W_\ell^\top \nu_\ell + \sum_{h=1}^{H} B_h^\top p_h + \sum_{g=1}^{G} A_g^\top r_g$. For fixed ξ, $\Psi(\cdot; \xi)$ is affine in x with slope $\mathcal{G}(\xi)$. Every ξ therefore defines an affine minorant of f, and an optimal $\xi \in \mathcal{D}^\star(x)$ defines an affine support at x. Map $\mathcal{G}$ reads the corresponding slope in the input space.

Proposition 2 (Dual representation)**.** *For every $x \in \mathbb{R}^d$, $f(x) = \max_{\xi \in \mathcal{K}_\nu \times \mathcal{P} \times \mathcal{K}_r} \Psi(x; \xi)$. Define the complete dual solution set $\mathcal{D}^\star(x) := \arg\max_{\xi \in \mathcal{K}_\nu \times \mathcal{P} \times \mathcal{K}_r} \Psi(x; \xi)$. For every $\xi \in \mathcal{D}^\star(x)$, $f(x) = \Psi(x; \xi),\ f(x') \geq \Psi(x'; \xi)\ \forall x' \in \mathbb{R}^d$. Hence, $\Psi(\cdot; \xi)$ is an affine support of f at x with input-space slope $\mathcal{G}(\xi)$.*

The representation follows from the ReLU epigraph dual, quadratic conjugacy, and the support-function representation of the Euclidean norm (Boyd & Vandenberghe, 2004). Consequently, $\mathcal{D}^\star(x)$

indexes the network supports active at x, and recovering first-order geometry reduces to characterizing this set and reading its slopes through $\mathcal{G}$.

3.2 Exact first-order geometry

The complete dual solution set can be determined directly from the forward-pass states $a_\ell(x)$, $q_h(x)$, and $u_g(x)$. For each ReLU coordinate, define the recursive upper bound by $\overline{\nu}_{L,i}(\nu) := c_i$ at the last layer and $\overline{\nu}_{\ell,i}(\nu) := (U_{\ell+1}^\top \nu_{\ell+1})_i$ for $\ell < L$.

Proposition 3 (Blockwise characterization of the optimal-dual set)**.** *For every x, $\mathcal{D}^\star(x) = \mathcal{D}_{\mathrm{R}}^\star(x) \times \prod_{h=1}^{H}\{\alpha_h q_h(x)\} \times \prod_{g=1}^{G} \mathcal{R}_g^\star(x)$. For each norm module, if $u_g(x) \neq 0$, then $\mathcal{R}_g^\star(x) = \left\{\lambda_g \frac{u_g(x)}{\|u_g(x)\|_2}\right\}$; if $u_g(x) = 0$, then $\mathcal{R}_g^\star(x) = \left\{r_g \in \mathbb{R}^{k_g} : \|r_g\|_2 \leq \lambda_g\right\}$.*

For the ReLU backbone, $\mathcal{D}_{\mathrm{R}}^\star(x)$ consists of all $\nu \in \mathcal{K}_\nu$ satisfying the following coordinatewise conditions. If $a_{\ell,i}(x) < 0$, then $\nu_{\ell,i} = 0$. If $a_{\ell,i}(x) > 0$, then $\nu_{\ell,i} = \overline{\nu}_{\ell,i}(\nu)$. If $a_{\ell,i}(x) = 0$, then $0 \leq \nu_{\ell,i} \leq \overline{\nu}_{\ell,i}(\nu)$. Moreover, $\mathcal{D}^\star(x)$ is nonempty, compact, and convex.

Thus, quadratic multipliers and norm multipliers with nonzero residuals are unique. ReLU multipliers are zero at negative preactivations and recursively determined by later layers at positive preactivations. The remaining free variables are associated with zero ReLU preactivations and zero norm residuals with positive weights.

The input-space image of this complete dual set gives the exact first-order geometry.

Theorem 4 (Exact subdifferential readout)**.** *For every $x \in \mathbb{R}^d$, $\partial f(x) = \mathcal{G}\big(\mathcal{D}^\star(x)\big) = \{\mathcal{G}(\xi) : \xi \in \mathcal{D}^\star(x)\}$. Moreover, for every $d \in \mathbb{R}^d$, $f'(x;d) = \max_{\xi \in \mathcal{D}^\star(x)} \mathcal{G}(\xi)^\top d$. Consequently, f is differentiable at x if and only if $\mathcal{G}(\mathcal{D}^\star(x))$ is a singleton.*

Each $\xi \in \mathcal{D}^\star(x)$ defines an affine support active at x, and Danskin's theorem (Danskin, 1966) recovers the complete set of input-space slopes. Although $\mathcal{P}$ is unbounded, its maximizer is uniquely $p_h^\star(x) = \alpha_h q_h(x)$ and is locally bounded in x; the dual maximum can therefore be locally restricted to a compact set. Since $\mathcal{D}^\star(x)$ is convex and $\mathcal{G}$ is affine, no additional convex hull is required.

Different dual solutions may induce the same input-space slope, so multiplier nonuniqueness does not necessarily imply functional nonsmoothness. Moreover, minimizing the norm of the dual multipliers need not minimize the norm of the resulting input-space slope, because the readout map introduces scaling, coupling, and null directions. DCI therefore constructs its certificate in the input space.

For the inference objective $F = f + \Phi$, with Φ finite, convex, and continuously differentiable,

$$\partial F(x) = \mathcal{G}\big(\mathcal{D}^\star(x)\big) + \nabla\Phi(x). \tag{4}$$

The recovered correspondence also has a closed graph and is locally bounded, properties used in the convergence analysis below.

Proposition 5 (Closedness and local boundedness)**.** *The correspondence $x \mapsto \mathcal{D}^\star(x)$ has nonempty compact values, is locally bounded, and has a closed graph. Consequently, so does $x \mapsto \mathcal{G}\big(\mathcal{D}^\star(x)\big) = \partial f(x)$. In particular, $x_n \to x,\ g_n \in \partial f(x_n),\ g_n \to g \Longrightarrow g \in \partial f(x)$.*

The result follows from compactness of $\mathcal{K}_\nu$ and $\mathcal{K}_r$, local boundedness of $p_h^\star(x) = \alpha_h q_h(x)$, and continuity of f, Ψ, and $\mathcal{G}$.

3.3 Exact second-order geometry on smooth branches

The dual readout above recovers the complete first-order geometry at arbitrary inputs. On a structurally nondegenerate region, the ReLU pattern is fixed and all positive-weight norm modules remain smooth, allowing the local curvature to be recovered explicitly from the network structure.

Proposition 6 (Smooth-branch geometry)**.** *Let $\mathcal{I}_+ := \{g \in \{1,\dots,G\} : \lambda_g > 0\}$, and let $\bar{x}$ be structurally nondegenerate. Then there exists $\rho > 0$ such that, for every $x \in \mathbb{B}(\bar{x}, \rho)$, the ReLU activation pattern is fixed, every $u_g(x)$ with $g \in \mathcal{I}_+$ is nonzero, and $\mathcal{D}^\star(x)$ is a singleton.*

On this neighborhood, the ReLU component is affine. Thus, for some $\bar{a} \in \mathbb{R}^d$ and $\bar{b} \in \mathbb{R}$,

$$f(x) = \bar{a}^\top x + \bar{b} + \sum_{h=1}^{H} \frac{\alpha_h}{2} \|q_h(x)\|_2^2 + \sum_{g \in \mathcal{I}_+} \lambda_g \|u_g(x)\|_2.$$

Writing $\widehat{u}_g(x) := u_g(x)/\|u_g(x)\|_2$, the gradient and Hessian are

$$\nabla f(x) = \bar{a} + \sum_{h=1}^{H} \alpha_h B_h^\top q_h(x) + \sum_{g \in \mathcal{I}_+} \lambda_g A_g^\top \widehat{u}_g(x), \tag{5a}$$

$$\nabla^2 f(x) = \sum_{h=1}^{H} \alpha_h B_h^\top B_h + \sum_{g \in \mathcal{I}_+} \lambda_g A_g^\top \frac{I - \widehat{u}_g(x)\widehat{u}_g(x)^\top}{\|u_g(x)\|_2} A_g \succeq 0. \tag{5b}$$

Moreover, $\nabla^2 f$ is Lipschitz continuous on every sufficiently small closed ball in this neighborhood.

The ReLU backbone contributes no curvature on a fixed activation region. Each quadratic module contributes the constant positive-semidefinite matrix $\alpha_h B_h^\top B_h$, while each norm module contributes curvature in directions orthogonal to its current residual and zero curvature in the radial direction. Since $\inf_{x \in \mathbb{B}(\bar{x},\rho)} \|u_g(x)\|_2 > 0, \quad g \in \mathcal{I}_+$, after reducing ρ if necessary, the norm-module Hessians are locally Lipschitz. If Φ is twice continuously differentiable, then $\nabla F(x) = \nabla f(x) + \nabla \Phi(x),\ \nabla^2 F(x) = \nabla^2 f(x) + \nabla^2 \Phi(x)$.

We have therefore recovered the two network-side geometric objects required for inference: the first-order geometry $\partial f(x)$ at arbitrary inputs and the local curvature $\nabla^2 f(x)$ on structurally nondegenerate regions. The next section combines the former with $\nabla \Phi(x)$ and $N_{\mathcal{X}}(x)$ to construct certified search directions, while the latter provides the local curvature metric used by DCI.

4 Dual-Certified Inference

We now turn the recovered SOC-ICNN subdifferential and local curvature into an iterative method for $\min_{x \in \mathcal{X}} F(x)$, where $F = f + \Phi$. Dual-certified inference (DCI) jointly optimizes over network multipliers and constraint normals to obtain an optimality residual and a search direction. A feasible line search and a proximal safeguard then determine the next iterate.

4.1 Search direction and optimality certificate

At a feasible input x, the directional derivative $F'(x; d) = \max_{g \in \partial F(x)} g^\top d$ describes the first-order variation of the objective. Given a positive definite metric $M \in \mathbb{R}^{d \times d}$, DCI selects the direction

$$d_M(x) := \operatorname*{arg\,min}_{d \in T_{\mathcal{X}}(x)} \left\{ F'(x; d) + \frac{1}{2} \|d\|_M^2 \right\}, \tag{6}$$

where $T_{\mathcal{X}}(x)$ is the tangent cone of $\mathcal{X}$ at x. The tangent constraint imposes first-order feasibility, while the quadratic term controls the scale of the direction through M.

The representation in equation 4 allows this model to be solved through the optimal network multipliers $\xi \in \mathcal{D}^\star(x)$ and normals $n \in N_{\mathcal{X}}(x)$, where $N_{\mathcal{X}}(x)$ is the normal cone of the feasible set at x. Define the constrained first-order set $\mathcal{C}_{\mathcal{X}}(x) := \partial F(x) + N_{\mathcal{X}}(x)$. Projecting the origin onto this set in the M^{-1}-norm gives

$$\left(\xi_M^\dagger(x), n_M^\dagger(x)\right) \in \operatorname*{arg\,min}_{\xi \in \mathcal{D}^\star(x), n \in N_{\mathcal{X}}(x)} \frac{1}{2} \left\| \mathcal{G}(\xi) + \nabla \Phi(x) + n \right\|_{M^{-1}}^2, \tag{7a}$$

$$g_M^\dagger(x) := \mathcal{G}\left(\xi_M^\dagger(x)\right) + \nabla \Phi(x) + n_M^\dagger(x), \quad \mathcal{R}_M(x) := \|g_M^\dagger(x)\|_{M^{-1}}. \tag{7b}$$

Here $\mathcal{G}$ maps optimal network multipliers to network subgradients, and $\mathcal{R}_M(x)$ is the projection residual. The set $\mathcal{C}_{\mathcal{X}}(x)$ is nonempty, closed, and convex, so $g_M^\dagger(x)$ is unique even when its multiplier and normal decomposition is nonunique. By Proposition 19, the unique solution of equation 6 is $d_M(x) = -M^{-1} g_M^\dagger(x)$.

Theorem 7 (Certificate and tangent common descent). *For every $x \in \mathcal{X}$ and $M \succ 0$, $\mathcal{R}_M(x) = 0 \iff x \in \arg\min_{z\in\mathcal{X}} F(z)$. If $\mathcal{R}_M(x) > 0$, then $d_M(x) \in T_{\mathcal{X}}(x)$ and*

$$g^\top d_M(x) \le -\mathcal{R}_M(x)^2, \ \ \forall\, g \in \partial F(x). \tag{8}$$

Consequently, $F'(x; d_M(x)) \le -\mathcal{R}_M(x)^2 < 0$.

Thus, the same projection either certifies global optimality or produces a tangent direction that decreases every affine support of F active at x. If F is μ_M-strongly convex on $\mathcal{X}$ with respect to the M-norm, where $\mu_M > 0$, the residual also bounds the objective gap: $0 \le F(x) - F^\star \le \mathcal{R}_M(x)^2/(2\mu_M)$, where $F^\star := \min_{z\in\mathcal{X}} F(z)$ (see Corollary 18).

4.2 Boundary-local computation and metric selection

By Proposition 3, several coordinates in equation 7a can be eliminated. Quadratic multipliers and norm multipliers with nonzero residuals have explicit values and can be substituted directly. ReLU multipliers are eliminated in reverse layer order: negative preactivations set their multipliers to zero, while positive preactivations express them recursively through later layers. The remaining network variables correspond to zero ReLU preactivations and zero norm residuals with positive weights, with the recursive constraints between layers retained.

Define the corresponding index sets $\mathcal{Z}_{\mathrm{R}}(x) := \{(\ell, i) : a_{\ell,i}(x) = 0\}$ and $\mathcal{Z}_{\mathrm{N}}(x) := \{g : u_g(x) = 0,\ \lambda_g > 0\}$. Suppose that the normal cone admits the representation $N_{\mathcal{X}}(x) = \{B_{\mathcal{X}}(x)\zeta : \zeta \in \mathcal{K}_{\mathcal{X}}(x)\}$, where ζ collects $q_{\mathcal{X}}(x)$ scalar normal parameters, $B_{\mathcal{X}}(x) \in \mathbb{R}^{d\times q_{\mathcal{X}}(x)}$, and $\mathcal{K}_{\mathcal{X}}(x) \subseteq \mathbb{R}^{q_{\mathcal{X}}(x)}$ is a closed convex cone. If $N_{\mathcal{X}}(x) = \{0\}$, we take $q_{\mathcal{X}}(x) = 0$.

Proposition 8 (Boundary-local projection). *After eliminating fixed or recursively determined dual coordinates, the projection in equation 7a contains at most $|\mathcal{Z}_{\mathrm{R}}(x)| + \sum_{g\in\mathcal{Z}_{\mathrm{N}}(x)} k_g + q_{\mathcal{X}}(x)$ scalar decision variables, where k_g is the dimension of the norm residual $u_g(x)$. Its objective is convex quadratic, and its feasible set consists of the recursive ReLU interval constraints, the Euclidean balls associated with zero norm residuals, and the cone $\mathcal{K}_{\mathcal{X}}(x)$. If $\mathcal{Z}_{\mathrm{R}}(x) = \mathcal{Z}_{\mathrm{N}}(x) = \varnothing$ and $q_{\mathcal{X}}(x) = 0$, then $\mathcal{C}_{\mathcal{X}}(x)$ is a singleton and no projection solve is required.*

Thus, the variable dimension of the additional projection problem is determined by the current network boundaries and the normal-cone representation.

Local curvature enters the direction model through the metric. At a structurally nondegenerate iterate x_k, if Φ is twice continuously differentiable near x_k, define $H_k := \nabla^2 f(x_k) + \nabla^2\Phi(x_k) = \nabla^2 F(x_k)$. The network Hessian is given in equation 5b. If $x_k \in \operatorname{int}\mathcal{X}$ and $H_k \succ 0$, choosing $M_k = H_k$ recovers the Newton direction (Nocedal & Wright, 2006) $d_{M_k}(x_k) = -H_k^{-1}\nabla F(x_k)$. For the global analysis, the metrics satisfy

$$0 < mI \preceq M_k \preceq \overline{m} I, \tag{9}$$

where $0 < m \le \overline{m} < \infty$ are fixed constants and I is the identity matrix. A regularized curvature metric is $M_k = \mathrm{clip}_{[m,\overline{m}]}(H_k + \tau_k I)$, where $\tau_k \ge 0$ is a regularization parameter and $\mathrm{clip}_{[m,\overline{m}]}$ clips eigenvalues to $[m, \overline{m}]$. When this curvature is unavailable, any positive definite metric satisfying equation 9 may be used.

4.3 Feasible steps and safeguard

Starting from $x_0 \in \mathcal{X}$, at iteration k we write $g_k^\dagger := g_{M_k}^\dagger(x_k),\ \mathcal{R}_k := \mathcal{R}_{M_k}(x_k),\ d_k := -M_k^{-1} g_k^\dagger$. DCI returns x_k when $\mathcal{R}_k \le \varepsilon$, where $\varepsilon \ge 0$ is the stopping tolerance. Otherwise, the tangent direction is converted into a feasible trial point using a retraction $\mathsf{P}_{\mathcal{X}} : \mathcal{X} \times \mathbb{R}^d \to \mathcal{X}$ satisfying

$$\mathsf{P}_{\mathcal{X}}(x, 0) = x, \ \ \mathsf{P}_{\mathcal{X}}(x, td) = x + td + o(t) \quad \text{as } t \downarrow 0 \tag{10}$$

for every $x \in \mathcal{X}$ and $d \in T_{\mathcal{X}}(x)$. The remainder satisfies $\|o(t)\|_2/t \to 0$, so the retraction preserves the direction to first order. One admissible choice is $\mathsf{P}_{\mathcal{X}}(x, v) = \Pi_{\mathcal{X}}(x + v)$, where $\Pi_{\mathcal{X}}$ denotes Euclidean projection onto $\mathcal{X}$.

Let $0 < \sigma < 1/2$ and $0 < \eta < 1$ be sufficient-decrease parameters, $0 < \beta < 1$ the backtracking contraction, and $J_{\max}$ a fixed nonnegative integer limiting the number of reductions. DCI tests

$t = \beta^j$ for $j = 0, \ldots, J_{\max}$, in this order. For each trial, define the feasible candidate $z_k(t) := \mathsf{P}_{\mathcal{X}}(x_k, td_k)$ and its displacement $s_k(t) := z_k(t) - x_k$. The first step satisfying

$$F'\big(x_k; s_k(t)\big) \le -\eta t \mathcal{R}_k^2, \;\; F\big(z_k(t)\big) \le F(x_k) + \sigma F'\big(x_k; s_k(t)\big) \tag{11}$$

is accepted, with $x_{k+1} = z_k(t)$. The first condition preserves sufficient directional decrease along the actual feasible displacement; the second adapts the Armijo sufficient-decrease principle (Armijo, 1966) to the directional prediction along this displacement. The directional derivative is computed using Theorem 4 and equation 4.

At any fixed point with positive residual, common descent and the first-order consistency in equation 10 ensure that all sufficiently small steps satisfy equation 11 (Lemma 21). The finite schedule gives the uniform lower bound $t \ge \beta^{J_{\max}}$ on accepted steps, relating their objective decrease to the current residual. If none of the tested steps is accepted, DCI applies a metric proximal safeguard based on proximal-point regularization (Rockafellar, 1976):

$$x_{k+1} = \operatorname*{arg\,min}_{z \in \mathcal{X}} \left\{ F(z) + \frac{1}{2\gamma} \|z - x_k\|_{M_k}^2 \right\}, \;\; \gamma > 0, \tag{12}$$

where γ is a fixed proximal parameter. This strongly convex subproblem has a unique solution. Its exact solution decreases the objective and yields a vector in the constrained first-order set at the new iterate, whose M_k^{-1}-norm bounds $\mathcal{R}_{M_k}(x_{k+1})$ from above (Lemma 22). The pseudocode is summarized in Algorithm 1.

4.4 Convergence

Theorem 9 (Global and local convergence)**.** Global convergence. *Assume that f is finite and convex, Φ is finite, convex, and continuously differentiable on $\mathbb{R}^d$, and $\mathcal{X}$ is nonempty, closed, and convex. Let the feasible retraction satisfy equation 10. Suppose that the initial level set $\mathcal{L}_0 := \{x \in \mathcal{X} : F(x) \le F(x_0)\}$ is compact, the metrics satisfy equation 9, $\varepsilon = 0$, and every invoked proximal subproblem is solved exactly. Then $F(x_{k+1}) \le F(x_k)$. Finite termination returns a global minimizer; for an infinite run, $F(x_k) \to F^\star$ and every accumulation point of $\{x_k\}$ is globally optimal. If the minimizer is unique, the entire sequence converges to it.*

Local quadratic convergence. *Suppose additionally that the minimizer $x^\star$ is unique, lies in $\operatorname{int} \mathcal{X}$, and is structurally nondegenerate. Assume that F is twice continuously differentiable near $x^\star$, $\nabla^2 F(x^\star) \succ 0$, and $\nabla^2 F$ is locally Lipschitz continuous. Assume also that the retraction is locally exact: $\mathsf{P}_{\mathcal{X}}(x, v) = x + v$ whenever x and $x + v$ lie in a sufficiently small neighborhood of $x^\star$. If $M_k = \nabla^2 F(x_k)$ in this neighborhood, then an infinite run eventually accepts unit steps and satisfies $\|x_{k+1} - x^\star\|_2 \le C\|x_k - x^\star\|_2^2$ for some constant $C > 0$ and all sufficiently large k.*

The two update types satisfy

$$F(x_k) - F(x_{k+1}) \ge \begin{cases} \sigma\eta\beta^{J_{\max}} \mathcal{R}_k^2, & \text{for an accepted local step,} \\ \dfrac{1}{2\gamma}\|x_{k+1} - x_k\|_{M_k}^2, & \text{for a proximal step.} \end{cases} \tag{13}$$

Moreover, proximal optimality gives $\gamma^{-1} M_k(x_k - x_{k+1}) \in \partial F(x_{k+1}) + N_{\mathcal{X}}(x_{k+1})$. Together with compactness and the closed graph of $x \mapsto \mathcal{C}_{\mathcal{X}}(x)$, these relations yield a globally optimal accumulation point. Monotonicity and continuity then give $F(x_k) \to F^\star$, establishing optimality of every accumulation point. Once DCI recovers the Newton iteration near the structurally nondegenerate minimizer, the standard local Newton analysis (Nocedal & Wright, 2006) yields quadratic convergence. Proofs are given in Sections C.7 and C.8.

5 Experiments

We organize five experiments to address four questions; complete settings and results are given in Section D. Since no existing method directly provides the same complete first- and second-order geometry of ICNNs, we compare DCI with AD-based optimizers, Gradient Sampling (Burke et al., 2005), a high-accuracy primal–dual solver applied to the full lifted conic formulation (Lifted-PD), and component ablations.

Success is independently determined from the objective gap and complete constrained residual; an invalid declared convergence is counted as a wrong stop. DCI-I uses the identity metric and DCI-C uses explicit curvature. Local variants omit the proximal safeguard, whereas Full variants include it.

5.1 Does the Readout Recover the Exact Geometry?

At structurally nondegenerate inputs, the explicit gradient and Hessian agree with independently computed autograd quantities from PyTorch (Paszke et al., 2019) to mean errors 2.12×10^{-15} and 4.12×10^{-16}, respectively. The second-order Taylor residual has log–log slope 3.006. At mixed structural boundaries, where ReLU preactivations and norm residuals vanish simultaneously, the directional readout agrees with one-sided finite differences to 2.37×10^{-8}. These results verify the smooth and nonsmooth geometric formulas; complete tests are given in Section D.2.

5.2 Does the Recovered Geometry Improve Inference?

We evaluate 24 strongly convex objectives whose known minimizers lie at ReLU, norm, or mixed structural boundaries, using two initializations per objective. Across 48 matched runs, DCI-I, DCI-C, and Lifted-PD achieve 100% success, compared with 68.8% for Gradient Sampling and 0% for Single-Branch and L-BFGS (Liu & Nocedal, 1989). Gradient Sampling also records a 12.5% wrong-stop rate (Table 1(a)). These results demonstrate the reliability of DCI on the constructed boundary problems and support the use of the recovered subdifferential for nonsmooth inference.

To isolate curvature, all trajectories remain on smooth branches, where AD-GD and Readout-GD use the same exact gradients. We test quadratic and quadratic–norm objectives with Hessian condition numbers $\kappa \in \{1, \dots, 10^4\}$. DCI-C achieves 100% success throughout, with median iteration counts of one and two on the respective families. Readout-GD and AD-L-BFGS slow as κ increases, while AD-GD closely tracks Readout-GD (Fig. 1). Canonical-Newton uses AD gradients and Hessians (Nocedal & Wright, 2006) and matches DCI-C's iteration counts. At $\kappa = 10^4$, DCI-C's median runtimes are $7.2\times$ and $10.4\times$ faster, respectively, demonstrating the computational benefit.

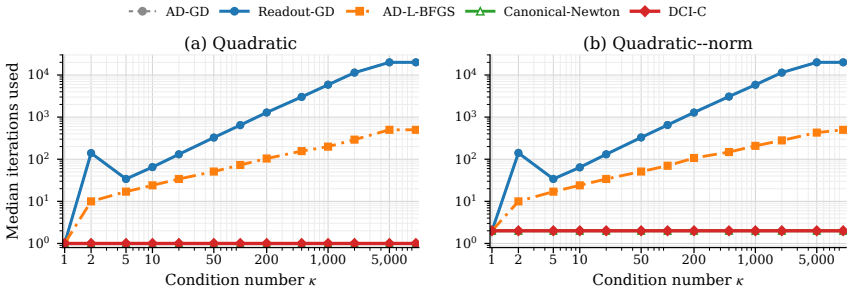


Figure 1: **Inference under increasing condition number.** Median iterations over five instances per κ on smooth branches, isolating curvature effects. DCI-C takes one and two iterations on the two families, while first-order and quasi-Newton baselines slow as conditioning worsens.

5.3 Why Are Constraint Geometry and the Safeguard Needed?

We evaluate 95 objectives with known solutions in unconstrained space and over box, ball, simplex, and second-order cone feasible sets. The suite contains regular, structural-boundary, constraint-boundary, joint-boundary, and stress cases. At the structural, joint, and stress optima, the network-only residual has mean 1.123, while the joint residual is at most 3.33×10^{-8} across all optima.

AD-PGD and Readout-PGD use the projected-gradient update structure (Nocedal & Wright, 2006), with a branch-selected AD vector and a minimum-norm objective subgradient, respectively. DCI-I-Local solves 63.2% of the suite, DCI-C-Local solves 94.7%, and DCI-C-Full solves all instances (Table 1(b)). With the default $J_{\max} = 10$, the safeguard is invoked in only $5/95$ runs. Prox-Only, our metric variant of the proximal-point method (Rockafellar, 1976), takes a proximal step at every iteration and also solves all instances but requires $4.83\times$ the median runtime of DCI-C-Full. In a separate budget sweep over $J_{\max} = 0, \dots, 10$, larger budgets improve Local success and reduce safeguard use, while Full remains 100% successful at every tested budget (Figure 2).

Table 1: **Boundary inference and safeguard comparison.** (a) Success (Succ.) and wrong-stop rates over 48 matched runs per method. (b) Success, certified termination (Cert.), safeguard use (SG), and median runtime over 95 objectives per method. All rates are percentages; a dash denotes an unassessed wrong-stop rate.

(a) Boundary inference

Method	Succ.	Wrong
Single-Branch	0.0	0.0
L-BFGS	0.0	–
Grad. Sampling	68.8	12.5
DCI-I	100.0	0.0
DCI-C	100.0	0.0
Lifted-PD	100.0	0.0

(b) Constraint and safeguard

Method	Succ.	Cert.	SG	Time (ms)
AD-PGD	49.5	36.8	0.0	86.9
Readout-PGD	60.0	47.4	0.0	71.5
DCI-I-Local	63.2	63.2	0.0	168.1
DCI-C-Local	94.7	94.7	0.0	93.4
DCI-C-Full	100.0	100.0	5.3	135.6
Prox-Only	100.0	100.0	100.0	655.4

5.4 Can DCI Reliably Optimize Trained Application Objectives?

We consider day-ahead battery scheduling using historical data from the Swedish SE3 bidding zone in the Open Power System Data Time series package (Open Power System Data, 2020). Each pre-dispatch context contains day-ahead prices, official load forecasts, causal wind forecasts constructed from past observations, and calendar information. The decision $z \in [-1, 1]^{12}$ parameterizes a 24-hour battery trajectory whose energy, power, and terminal constraints are satisfied by construction. An SOC-ICNN is trained to represent the resulting context-dependent convex operating cost.

We evaluate three independently trained SOC-ICNNs on 30 held-out contexts, giving 90 frozen box-constrained objectives. The comparisons include AD-L-BFGS-B, which uses the bound-constrained limited-memory BFGS method (Byrd et al., 1995). Every method optimizes the same learned objective from the same initial point, thereby isolating inference error from surrogate approximation error. DCI-C is the only non-reference method with 100% success, reaching a median complete residual of 1.18×10^{-6} in two iterations and 3.09 ms. Lifted-PD also solves every objective but requires a median of 537.5 iterations and 241 ms, approximately $78\times$ the runtime of DCI-C (Table 2(a)).

For the operational comparison, Point Forecast substitutes pre-dispatch forecasts into the engineering objective, following point-prediction-driven optimization (Bertsimas & Kallus, 2020). Historical SAA uses nearest-neighbor sample average approximation (Bertsimas & Kallus, 2020), averaging the realized objectives of the seven nearest standardized training contexts. No Storage uses $z = 0$, and the perfect-information (PI) oracle minimizes the engineering objective using realized test-day net load. These application-specific baselines are detailed in Section D.6.2.

Table 2: **Battery-scheduling experiments.** (a) Inference on 90 frozen SOC-ICNN objectives; Wrong denotes an invalid declared convergence. (b) Mean operational regret under the realized engineering cost.

(a) Frozen-objective inference

Method	Success	Wrong	Iter.	ms
Readout-GD	94.4%	0.0%	18	14.9
AD-L-BFGS-B	94.4%	5.6%	7	7.00
DCI-I	96.7%	0.0%	18	21.0
DCI-C	100%	0.0%	2	3.09
Lifted-PD	100%	0.0%	537.5	241

(b) End-to-end scheduling

Pipeline	Regret
No Storage	1.1567
Point Forecast	0.0054
Historical SAA	0.0104
SOC-ICNN	2.1733
PI oracle	0.0000

Across 1089 SOC-ICNN seed–day solves, DCI-C certifies 99.1% of decisions with a median runtime of 3.05 ms. The results identify surrogate approximation as the dominant source of operational regret (Table 2(b)); further analysis is given in Section D.6.2.

6 Conclusion and Discussion

We recover the complete first-order geometry of SOC-ICNNs from their dual structure and develop DCI for certified and efficient inference. While the current method relies on structured convex architectures and exact geometric information, combining DCI with quasi-Newton, active-set, or other hybrid optimization strategies may further improve practical convergence and extend the framework to broader model classes.

AI USE STATEMENT

ChatGPT 5.6Terra assists with the coding work and the first-round review feedback. The authors take responsibility for the final manuscript.

REFERENCES

Alnur Ali, Eric Wong, and J. Zico Kolter. A semismooth Newton method for fast, generic convex programming. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, 2017. URL https://proceedings.mlr.press/v70/ali17a.html.

Amal Alphonse, Constantin Christof, Michael Hintermüller, and Ioannis P. A. Papadopoulos. A globalized inexact semismooth Newton method for nonsmooth fixed-point equations involving variational inequalities. *Computational Optimization and Applications*, 93(1):1–55, 2026. doi: 10.1007/s10589-025-00722-8.

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, 2017. URL https://proceedings.mlr.press/v70/amos17b.html.

Aleksandr Y. Aravkin, James V. Burke, and Michael P. Friedlander. Variational properties of value functions. *SIAM Journal on Optimization*, 23(3):1689–1717, 2013. doi: 10.1137/120899157.

Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966. doi: 10.2140/pjm.1966.16.1. URL https://msp.org/pjm/1966/16-1/pjm-v16-n1-p01-s.pdf.

Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020. doi: 10.1287/mnsc.2018.3253.

Jérôme Bolte and Edouard Pauwels. A mathematical model for automatic differentiation in machine learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/7a674153c63cff1ad7f0e261c369ab2c-Abstract.html.

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, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/70afbf2259b4449d8ae1429e054df1b1-Abstract.html.

Jérôme Bolte, Ryan Boustany, Edouard Pauwels, and Béatrice Pesquet-Popescu. On the complexity of nonsmooth automatic differentiation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=uqg3FhRZaq.

J. Frédéric Bonnans and Alexander Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, 2000. doi: 10.1007/978-1-4612-1394-9.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

Felix Bünning, Adrian Schalbetter, Ahmed Aboudonia, Mathias Hudoba de Badyn, Philipp Heer, and John Lygeros. Input convex neural networks for building MPC. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*. PMLR, 2021. URL https://proceedings.mlr.press/v144/bunning21a.html.

James V. Burke, Adrian S. Lewis, and Michael L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005. doi: 10.1137/030601296.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. doi: 10.1137/0916069.

John M. Danskin. The theory of Max-Min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966. doi: 10.1137/0114053.

Fuxiaoyue Feng, Chao Ding, and Xudong Li. A quadratically convergent semismooth Newton method for nonlinear semidefinite programming without generalized Jacobian regularity. *Mathematical Programming*, 214(1–2):643–683, 2025. doi: 10.1007/s10107-025-02198-0.

Chin-Wei Huang, Ricky T. Q. Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2012.05942.

Krzysztof C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46:105–122, 1990. doi: 10.1007/BF01585731.

Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M. Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous Wasserstein-2 benchmark. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7a6a6127ff85640ec69691fb0f7cb1a2-Abstract.html.

Maciej Ławryńczuk. Input convex neural networks in nonlinear predictive control: A multi-model approach. *Neurocomputing*, 513:273–293, 2022. doi: 10.1016/j.neucom.2022.09.108.

Wonyeol Lee, Sejun Park, and Alex Aiken. On the correctness of automatic differentiation for neural networks with machine-representable parameters. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023. URL https://proceedings.mlr.press/v202/lee23p.html.

Tianyi Lin, Marco Cuturi, and Michael Jordan. A specialized semismooth Newton method for kernel-based optimal transport. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*. PMLR, 2024. URL https://proceedings.mlr.press/v238/lin24a.html.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. doi: 10.1007/BF01589116.

Kang Liu, Jianchen Hu, and Wei Peng. SOC-ICNN: From polyhedral to conic geometry for learning convex surrogate functions, 2026. URL https://arxiv.org/abs/2604.22355.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020. URL https://proceedings.mlr.press/v119/makkuva20a.html.

Andre Milzarek, Fabian Schaipp, and Michael Ulbrich. A semismooth Newton stochastic proximal point algorithm with variance reduction. *SIAM Journal on Optimization*, 34(1):1157–1185, 2024. doi: 10.1137/22M1488181.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5.

Open Power System Data. Data package time series. Data set, 2020. URL https://data.open-power-system-data.org/time_series/2020-10-06/. Version 2020-10-06.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976. doi: 10.1137/0314056.

R. Tyrrell Rockafellar and Roger J. B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1998. doi: 10.1007/978-3-642-02431-3.

A Related Work

ICNN inference and automatic differentiation. ICNNs have been used as learned convex objectives in optimal transport, invertible modeling, system identification, and predictive control (Amos et al., 2017; Makkuva et al., 2020; Huang et al., 2021; Korotin et al., 2021; Bünning et al., 2021; Ławryńczuk, 2022). SOC-ICNNs extend this class with quadratic and second-order-cone modules and admit a structured conic value representation (Liu et al., 2026). Gradient-based inference commonly relies on derivatives supplied by AD. At nonsmooth inputs, however, AD follows an executed computational branch and returns a branch-dependent derivative (Bolte & Pauwels, 2020; Bolte et al., 2021; 2023; Lee et al., 2023). Such a derivative need not encode the complete convex subdifferential required for stationarity certification or directional-derivative calculations. Our work instead exploits the structured dual representation of SOC-ICNNs to recover the complete set of supporting slopes active at the current input.

Nonsmooth optimization and sensitivity analysis. Classical sensitivity results for marginal functions and parametric optimization relate optimal primal or dual solutions to subgradients of value functions (Danskin, 1966; Rockafellar & Wets, 1998; Bonnans & Shapiro, 2000; Aravkin et al., 2013). General nonsmooth optimization methods use gradient sampling, bundle models, proximal regularization, and generalized Newton systems to construct reliable search directions and obtain global or fast local convergence (Burke et al., 2005; Rockafellar, 1976; Ali et al., 2017; Lin et al., 2024; Milzarek et al., 2024; Feng et al., 2025; Alphonse et al., 2026). These methods are typically formulated for generic objective or oracle models and obtain the required nonsmooth information through sampling, local models, or proximal constructions. In contrast, we recover the exact subdifferential directly from the internal dual structure of a trained SOC-ICNN and combine it with feasible-set geometry for certified inference, while using the same network structure to obtain explicit curvature on smooth regions.

B Proofs for the First- and Second-Order Geometry of SOC-ICNNs

This appendix expands the SOC-ICNN model in Section 2 and provides complete proofs for the geometric results in Section 3. We first describe the architecture and its convexity, then establish its exact conic representation, derive the structured dual, and characterize the complete set of optimal multipliers. We subsequently prove the subdifferential formula, establish closedness and local boundedness of the associated mappings, and derive the local Hessian. The canonical multiplier selector and the blockwise subdifferential formula are included below.

B.1 Architecture and notation

Fix a context y and trained network parameters θ. We suppress this dependence and write $f(x) = f_\theta(x; y)$, with $x \in \mathbb{R}^d$ the decision input. The network contains a ReLU ICNN backbone, H quadratic modules, and G Euclidean norm modules. All three components receive x; the quadratic and norm modules act in parallel with the backbone. Their scalar contributions are added to form the network output.

ReLU backbone. Let n_ℓ be the width of hidden layer ℓ, for $\ell = 1, \dots, L$. Its preactivation $a_\ell(x) \in \mathbb{R}^{n_\ell}$ is the vector before the nonlinearity, and its hidden activation $z_\ell(x) \in \mathbb{R}^{n_\ell}$ is the vector after applying ReLU. The recursion is

$$a_\ell(x) = W_\ell x + U_\ell z_{\ell-1}(x) + b_\ell, \;\; z_\ell(x) = \mathrm{ReLU}(a_\ell(x)), \tag{14}$$

where ReLU acts coordinatewise as $\mathrm{ReLU}(t) = \max\{0, t\}$. The matrix $W_\ell \in \mathbb{R}^{n_\ell \times d}$ supplies a direct connection from the decision input to layer ℓ. For $\ell \geq 2$, the matrix $U_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ maps the previous hidden activation to the current layer, and $b_\ell \in \mathbb{R}^{n_\ell}$ is the bias. We use $z_0 \equiv 0$, so the first layer is $a_1(x) = W_1 x + b_1$ and the term $U_1 z_0$ is omitted. The matrices U_ℓ are nonnegative coordinatewise; the input matrices W_ℓ and biases b_ℓ have no sign restrictions. The scalar contribution from the final hidden layer is $c^\top z_L(x)$, where $c \in \mathbb{R}^{n_L}_+$.

Quadratic and norm modules. For $h = 1, \dots, H$, quadratic module h uses the affine residual $q_h(x) = B_h x + e_h$, with $B_h \in \mathbb{R}^{m_h \times d}$ and $e_h \in \mathbb{R}^{m_h}$, and contributes $\alpha_h \|q_h(x)\|_2^2/2$. Here m_h is its residual dimension and $\alpha_h > 0$ is its weight; zero-weight quadratic modules are omitted. For $g = 1, \dots, G$, norm module g uses $u_g(x) = A_g x + d_g$, with $A_g \in \mathbb{R}^{k_g \times d}$ and $d_g \in \mathbb{R}^{k_g}$, and contributes $\lambda_g \|u_g(x)\|_2$, where k_g is its residual dimension and $\lambda_g \geq 0$. The matrices and offsets in these affine residuals have no sign restrictions.

Output and convexity. With $v \in \mathbb{R}^d$ and $b_0 \in \mathbb{R}$ denoting the direct linear output coefficient and output bias, respectively, the network is

$$f(x) = c^\top z_L(x) + v^\top x + b_0 + \sum_{h=1}^{H} \frac{\alpha_h}{2} \|B_h x + e_h\|_2^2 + \sum_{g=1}^{G} \lambda_g \|A_g x + d_g\|_2. \tag{15}$$

This is the model in equation 3. To see its convexity, each coordinate of $z_1(x) = \mathrm{ReLU}(W_1 x + b_1)$ is convex. If the coordinates of $z_{\ell-1}$ are convex, nonnegativity of U_ℓ makes every coordinate of a_ℓ convex. Taking its maximum with zero preserves convexity, so the coordinates of z_ℓ are convex by induction. The nonnegative output weights c preserve convexity of the scalar backbone contribution. Each squared norm and norm of an affine map is convex, and its coefficient is nonnegative. Adding these terms and the affine output term proves that f is finite and convex on $\mathbb{R}^d$. The case $H = G = 0$ recovers a standard ReLU ICNN.

Parameters and forward states. For a fixed context, the matrices, biases, offsets, and output weights above remain constant during inference. In contrast, $a_\ell(x), z_\ell(x), q_h(x)$, and $u_g(x)$ vary with the decision input and are computed during a forward pass. Their distinction is important for the geometric analysis: the parameters determine the maps from multiplier space to input space, while the forward states determine which multiplier conditions apply at the current input. Structural boundaries occur when a ReLU preactivation is zero or a norm residual with positive weight is zero. The results below use these states to recover the complete subdifferential; local Hessians are obtained on the structurally nondegenerate regions of Definition 1.

B.2 Exact primal conic representation

Recall the second-order and rotated second-order cone $\mathcal{Q}^{k+1} = \left\{(t, \omega) \in \mathbb{R} \times \mathbb{R}^k : \|\omega\|_2 \leq t\right\}$, and $\mathcal{Q}_r^{k+2} = \left\{(s, \tau, \eta) : \|\eta\|_2^2 \leq 2s\tau, \; s \geq 0, \; \tau \geq 0\right\}$.

For a fixed decision input x, the SOC-ICNN output in equation 3 admits the equivalent conic representation

$$\begin{aligned} \min_{z,s,\eta,t,\omega} \; & c^\top z_L + v^\top x + b_0 + \sum_{h=1}^{H} \alpha_h s_h + \sum_{g=1}^{G} \lambda_g t_g \\ \text{s.t. } & z_\ell \geq W_\ell x + U_\ell z_{\ell-1} + b_\ell, \;\; z_\ell \geq 0, \\ & \eta_h = B_h x + e_h, \;\; (s_h, 1, \eta_h) \in \mathcal{Q}_r^{m_h+2}, \\ & \omega_g = A_g x + d_g, \;\; (t_g, \omega_g) \in \mathcal{Q}^{k_g+1}. \end{aligned} \tag{16}$$

Lemma 10 (Exact primal value)**.** *For every fixed x, the optimal value of equation 16 equals the SOC-ICNN output $f(x)$ in equation 3. In particular, the forward states $z_\ell(x)$, together with $\eta_h = q_h(x)$, $s_h = \frac{1}{2}\|q_h(x)\|_2^2$, $\omega_g = u_g(x)$, $t_g = \|u_g(x)\|_2$, form an optimal primal solution.*

Proof. Fix x. The equality constraints impose $\eta_h = q_h(x),\ \omega_g = u_g(x)$. The rotated-cone constraint $(s_h, 1, \eta_h) \in \mathcal{Q}_r^{m_h+2}$ implies $\|\eta_h\|_2^2 \le 2s_h$, and hence $s_h \ge \frac{1}{2}\|q_h(x)\|_2^2$. Because $\alpha_h > 0$, the smallest possible contribution of this block is attained at $s_h = \frac{1}{2}\|q_h(x)\|_2^2$.

Likewise, $(t_g, \omega_g) \in \mathcal{Q}^{k_g+1}$ implies $t_g \ge \|\omega_g\|_2 = \|u_g(x)\|_2$. If $\lambda_g > 0$, the minimum contribution is attained at equality. If $\lambda_g = 0$, equality remains a feasible optimal choice.

It remains to consider the ReLU backbone. The forward states satisfy $z_\ell(x) = \max\{0,\, W_\ell x + U_\ell z_{\ell-1}(x) + b_\ell\}$ componentwise and are therefore feasible.

Let $\widetilde{z}$ be any other feasible hidden-state sequence. We show by induction that $\widetilde{z}_\ell \ge z_\ell(x)$ componentwise for every layer. At the first layer, $\widetilde{z}_1 \ge \max\{0, W_1 x + b_1\} = z_1(x)$. Suppose that $\widetilde{z}_{\ell-1} \ge z_{\ell-1}(x)$. Since $U_\ell \ge 0$, $W_\ell x + U_\ell \widetilde{z}_{\ell-1} + b_\ell \ge W_\ell x + U_\ell z_{\ell-1}(x) + b_\ell$. Together with $\widetilde{z}_\ell \ge 0$, this gives $\widetilde{z}_\ell \ge \max\{0,\, W_\ell x + U_\ell z_{\ell-1}(x) + b_\ell\} = z_\ell(x)$. Thus $\widetilde{z}_L \ge z_L(x)$. Since $c \ge 0$, $c^\top \widetilde{z}_L \ge c^\top z_L(x)$. The forward states therefore minimize the ReLU block. Adding the optimal values of the ReLU, quadratic, and norm blocks yields exactly the SOC-ICNN output in equation 3. □

B.3 Derivation of the structured dual representation

We first derive the reduced dual representation of the ReLU backbone.

Lemma 11 (Dual representation of the ReLU backbone)**.** *For every fixed x,*

$$c^\top z_L(x) + v^\top x + b_0 = \max_{\nu \in \mathcal{K}_\nu} \left\{ v^\top x + b_0 + \sum_{\ell=1}^{L} \nu_\ell^\top (W_\ell x + b_\ell) \right\}. \tag{17}$$

Proof. For each layer, write the primal inequalities as $W_\ell x + U_\ell z_{\ell-1} + b_\ell - z_\ell \le 0,\ \ -z_\ell \le 0$. Associate nonnegative multipliers $\nu_\ell \ge 0$ and $\mu_\ell \ge 0$, respectively. The terms in the Lagrangian involving the hidden variables are

$$\sum_{\ell=1}^{L-1} z_\ell^\top \left(U_{\ell+1}^\top \nu_{\ell+1} - \nu_\ell - \mu_\ell \right) + z_L^\top \left(c - \nu_L - \mu_L \right). \tag{18}$$

The infimum over the unrestricted Lagrangian variables z_ℓ is finite if and only if $\mu_L = c - \nu_L$ and $\mu_\ell = U_{\ell+1}^\top \nu_{\ell+1} - \nu_\ell,\ \ell = 1, \dots, L-1$. The conditions $\nu_\ell \ge 0$ and $\mu_\ell \ge 0$ are therefore equivalent to $0 \le \nu_L \le c,\ \ 0 \le \nu_\ell \le U_{\ell+1}^\top \nu_{\ell+1},\ \ell < L$, which define $\mathcal{K}_\nu$.

After eliminating the hidden variables and μ, the dual objective is $v^\top x + b_0 + \sum_{\ell=1}^{L} \nu_\ell^\top (W_\ell x + b_\ell)$.

The primal ReLU epigraph problem is feasible by the forward construction in Lemma 10. It is also bounded below because $z_L \ge 0$ and $c \ge 0$. Standard linear-programming strong duality therefore gives equation 17. □

Lemma 12 (Compactness of the ReLU multiplier set)**.** *The set $\mathcal{K}_\nu$ is a nonempty compact convex polytope.*

Proof. The zero sequence belongs to $\mathcal{K}_\nu$, so the set is nonempty. It is defined by finitely many linear inequalities and is therefore closed and convex.

To prove boundedness, define recursively $\beta_L := c,\ \ \beta_\ell := U_{\ell+1}^\top \beta_{\ell+1},\ \ell = L-1, \dots, 1$. Because $c \ge 0$ and $U_\ell \ge 0$, every β_ℓ is finite and componentwise nonnegative. For any $\nu \in \mathcal{K}_\nu$, $0 \le \nu_L \le \beta_L$. If $0 \le \nu_{\ell+1} \le \beta_{\ell+1}$, then $0 \le \nu_\ell \le U_{\ell+1}^\top \nu_{\ell+1} \le U_{\ell+1}^\top \beta_{\ell+1} = \beta_\ell$. Backward induction therefore gives $0 \le \nu_\ell \le \beta_\ell$ for every layer. Hence $\mathcal{K}_\nu$ is bounded and thus compact. □

The quadratic and norm modules admit the following standard representations.

Lemma 13 (Quadratic conjugate and norm support)**.** *For every $q \in \mathbb{R}^m$, $u \in \mathbb{R}^k$, $\alpha > 0$, and $\lambda \ge 0$,*

$$\frac{\alpha}{2}\|q\|_2^2 = \max_{p \in \mathbb{R}^m} \left\{ p^\top q - \frac{1}{2\alpha}\|p\|_2^2 \right\}, \tag{19}$$

$$\lambda \|u\|_2 = \max_{\|r\|_2 \le \lambda} r^\top u. \tag{20}$$

The maximizer in equation 19 is uniquely $p = \alpha q$. The optimizer set in equation 20 is

$$\arg\max_{\|r\|_2 \le \lambda} r^\top u = \begin{cases} \left\{ \lambda \dfrac{u}{\|u\|_2} \right\}, & u \neq 0, \\ \left\{ r \in \mathbb{R}^k : \|r\|_2 \le \lambda \right\}, & u = 0. \end{cases} \tag{21}$$

Proof. Completing the square gives $p^\top q - \frac{1}{2\alpha}\|p\|_2^2 = \frac{\alpha}{2}\|q\|_2^2 - \frac{1}{2\alpha}\|p - \alpha q\|_2^2$. The maximum is therefore $\alpha\|q\|_2^2/2$, uniquely attained at $p = \alpha q$.

For the norm block, Cauchy–Schwarz gives $r^\top u \le \|r\|_2\|u\|_2 \le \lambda\|u\|_2$. When $u \neq 0$, equality is attained uniquely by $r = \lambda u/\|u\|_2$. When $u = 0$, every feasible r attains value zero. □

Proof of Proposition 2. Apply Lemma 11 to the ReLU backbone, equation 19 to every quadratic module, and equation 20 to every norm module. Their dual variables belong to independent feasible blocks, so the maximum of the sum equals the sum of the blockwise maxima. This gives $f(x) = \max_{\xi \in \mathcal{K}_\nu \times \mathcal{P} \times \mathcal{K}_r} \Psi(x; \xi)$.

Let $\xi \in \mathcal{D}^\star(x)$. Its feasibility does not depend on the input. Therefore, for every $x' \in \mathbb{R}^d$, $\Psi(x'; \xi) \le f(x')$. Optimality at x gives $\Psi(x; \xi) = f(x)$. Since $\Psi(\cdot; \xi)$ is affine with slope $\mathcal{G}(\xi)$, $\Psi(x'; \xi) = \Psi(x; \xi) + \mathcal{G}(\xi)^\top(x' - x)$. Combining these identities yields $f(x') \ge f(x) + \mathcal{G}(\xi)^\top(x' - x),\ \ x' \in \mathbb{R}^d$. Thus $\Psi(\cdot; \xi)$ is an affine support of f at x, and $\mathcal{G}(\xi) \in \partial f(x)$. □

B.4 COMPLETE DUAL SOLUTION SET

Proof of Proposition 3. The dual feasible set is a Cartesian product, and Ψ is a sum of functions depending on separate multiplier blocks. Consequently, its argmax factorizes into the Cartesian product of the blockwise argmax sets.

For a quadratic module, Lemma 13 gives the unique optimizer $p_h^\star(x) = \alpha_h q_h(x)$. For a norm module, the same lemma gives exactly $\mathcal{R}_g^\star(x)$.

It remains to characterize the ReLU multipliers. Consider the forward primal states from Lemma 10. The two inequalities associated with coordinate (ℓ, i) are $a_{\ell,i}(x) - z_{\ell,i}(x) \le 0,\ \ -z_{\ell,i}(x) \le 0$, with multipliers $\nu_{\ell,i}$ and $\mu_{\ell,i}$.

If $a_{\ell,i}(x) < 0$, then $z_{\ell,i}(x) = 0$, and the first inequality has strictly negative slack. Complementary slackness therefore gives $\nu_{\ell,i} = 0$.

If $a_{\ell,i}(x) > 0$, then $z_{\ell,i}(x) = a_{\ell,i}(x) > 0$. The second inequality is strictly slack, so $\mu_{\ell,i} = 0$. Lagrangian stationarity from equation 18 gives $\nu_{\ell,i} = \overline{\nu}_{\ell,i}(\nu)$.

If $a_{\ell,i}(x) = 0$, both primal inequalities are active. Dual feasibility and stationarity impose only $0 \le \nu_{\ell,i} \le \overline{\nu}_{\ell,i}(\nu)$. These conditions are also sufficient: together with the forward primal states, they satisfy primal feasibility, dual feasibility, stationarity, and complementary slackness. They therefore characterize the complete ReLU optimal set.

Combining the three blocks proves the stated product representation. The ReLU optimizer set is a nonempty closed convex subset of the compact polytope $\mathcal{K}_\nu$: equivalently, it is the argmax set of a linear function over $\mathcal{K}_\nu$. Each quadratic optimizer set is a singleton, and each norm optimizer set is either a singleton or a compact Euclidean ball. Their finite Cartesian product is therefore nonempty, compact, and convex. □

B.4.1 CANONICAL MINIMUM-MULTIPLIER SELECTOR

Although DCI minimizes the norm of the input-space readout rather than the multiplier norm, a canonical multiplier provides a deterministic subgradient representative.

Define

$$\widehat{\nu}_L(x) = c \odot \mathbf{1}_{\{a_L(x) > 0\}}, \tag{22}$$

$$\widehat{\nu}_\ell(x) = \left(U_{\ell+1}^\top \widehat{\nu}_{\ell+1}(x)\right) \odot \mathbf{1}_{\{a_\ell(x) > 0\}},\ \ \ell = L-1, \ldots, 1, \tag{23}$$

and

$$\widehat{r}_g(x) = \begin{cases} \lambda_g \dfrac{u_g(x)}{\|u_g(x)\|_2}, & u_g(x) \neq 0, \\ 0, & u_g(x) = 0. \end{cases} \tag{24}$$

Together with $\widehat{p}_h(x) = \alpha_h q_h(x)$, define $\widehat{\xi}(x) := (\widehat{\nu}(x), \widehat{p}(x), \widehat{r}(x))$.

Corollary 14 (Canonical multiplier selector)**.** *For every x, the canonical multiplier is the unique minimum-Euclidean-norm element of the complete dual solution set:*

$$\widehat{\xi}(x) = \operatorname*{arg\,min}_{\xi \in \mathcal{D}^\star(x)} \|\xi\|_2. \tag{25}$$

Proof. We first verify that $\widehat{\xi}(x) \in \mathcal{D}^\star(x)$. The quadratic and norm blocks satisfy the characterization in Proposition 3 by construction. For the ReLU block, $\widehat{\nu}_L$ equals c on strictly active final-layer coordinates and zero otherwise. Recursively, $\widehat{\nu}_\ell$ equals its induced upper bound on strictly active coordinates and zero on inactive or zero-preactivation coordinates. Hence it satisfies every equality and interval condition in $\mathcal{D}^\star_{\mathrm{R}}(x)$.

The quadratic multipliers are fixed throughout $\mathcal{D}^\star(x)$. For a zero norm residual, the optimizer set is a Euclidean ball centered at the origin, whose unique minimum-norm element is zero. For a nonzero norm residual, the multiplier is fixed.

It remains to prove minimality for the ReLU block. We first show that $\widehat{\nu}_\ell(x) \leq \nu_\ell$ componentwise for every $\nu \in \mathcal{D}^\star_{\mathrm{R}}(x)$.

At the final layer: if $a_{L,i}(x) > 0$, then both $\widehat{\nu}_{L,i}$ and $\nu_{L,i}$ equal c_i; if $a_{L,i}(x) < 0$, both equal zero; if $a_{L,i}(x) = 0$, then $\widehat{\nu}_{L,i} = 0 \leq \nu_{L,i}$. Thus $\widehat{\nu}_L \leq \nu_L$.

Suppose now that $\widehat{\nu}_{\ell+1} \leq \nu_{\ell+1}$. Since $U_{\ell+1} \geq 0$, $U_{\ell+1}^\top \widehat{\nu}_{\ell+1} \leq U_{\ell+1}^\top \nu_{\ell+1}$. If $a_{\ell,i}(x) > 0$, both multipliers equal their respective recursive upper bounds, so $\widehat{\nu}_{\ell,i} = \left(U_{\ell+1}^\top \widehat{\nu}_{\ell+1}\right)_i \leq \left(U_{\ell+1}^\top \nu_{\ell+1}\right)_i = \nu_{\ell,i}$. If $a_{\ell,i}(x) < 0$, both are zero. If $a_{\ell,i}(x) = 0$, then $\widehat{\nu}_{\ell,i} = 0 \leq \nu_{\ell,i}$. Backward induction proves $0 \leq \widehat{\nu} \leq \nu$ componentwise for every optimal ReLU multiplier.

Consequently, $\|\widehat{\nu}\|_2 \leq \|\nu\|_2$, with strict inequality whenever $\nu \neq \widehat{\nu}$. Combining the ReLU, quadratic, and norm blocks proves that $\widehat{\xi}(x)$ is the unique minimum-norm element of $\mathcal{D}^\star(x)$. □

The corresponding canonical input-space slope is

$$g^{\mathrm{can}}(x) := \mathcal{G}\big(\widehat{\xi}(x)\big) = v + \sum_{\ell=1}^{L} W_\ell^\top \widehat{\nu}_\ell(x) + \sum_{h=1}^{H} \alpha_h B_h^\top q_h(x) + \sum_{g=1}^{G} A_g^\top \widehat{r}_g(x). \tag{26}$$

This is one valid supporting slope, but it does not generally minimize the norm of the readout in input space.

B.5 Exact subdifferential and directional readout

Proof of Theorem 4. Fix $x \in \mathbb{R}^d$. By Proposition 2, every $\xi \in \mathcal{D}^\star(x)$ defines an affine support of f at x. Hence

$$\mathcal{G}\big(\mathcal{D}^\star(x)\big) \subseteq \partial f(x). \tag{27}$$

We next justify the application of Danskin's theorem despite the unbounded quadratic multiplier space $\mathcal{P}$. Choose a compact neighborhood $X \subset \mathbb{R}^d$ containing x. Since every q_h is affine, $C_h := \max_{x' \in X} \alpha_h \|q_h(x')\|_2 < \infty$. Choose $R_h > C_h$ and define $\mathcal{P}_X := \prod_{h=1}^{H} \{p_h \in \mathbb{R}^{m_h} : \|p_h\|_2 \leq R_h\}$. For every $x' \in X$, the unique maximizing quadratic multiplier is $p_h^\star(x') = \alpha_h q_h(x')$, which lies in the interior of the corresponding ball. Therefore, throughout X, $f(x') = \max_{\xi \in \mathcal{K}_\nu \times \mathcal{P}_X \times \mathcal{K}_r} \Psi(x'; \xi)$. The restricted multiplier set is compact and independent of x', and its optimal set agrees with the original optimal set for every $x' \in X$.

Danskin's directional theorem now yields, for every direction d, $f'(x; d) = \max_{\xi \in \mathcal{D}^\star(x)} \nabla_x \Psi(x; \xi)^\top d$. Since $\nabla_x \Psi(x; \xi) = \mathcal{G}(\xi)$, $f'(x; d) = \max_{\xi \in \mathcal{D}^\star(x)} \mathcal{G}(\xi)^\top d$.

It remains to identify the full subdifferential. By Proposition 3, the set $\mathcal{D}^\star(x)$ is nonempty, compact, and convex. Because $\mathcal{G}$ is affine, $C(x) := \mathcal{G}\big(\mathcal{D}^\star(x)\big)$ is also nonempty, compact, and convex. Its support function satisfies $\sigma_{C(x)}(d) = \max_{g \in C(x)} g^\top d = f'(x;d)$. For a finite convex function, the directional derivative is also the support function of its subdifferential: $f'(x;d) = \max_{g \in \partial f(x)} g^\top d$. Thus $C(x)$ and $\partial f(x)$ are nonempty closed convex sets with identical support functions. They must therefore coincide: $\partial f(x) = \mathcal{G}\big(\mathcal{D}^\star(x)\big)$.

Finally, a finite convex function is differentiable at x if and only if its subdifferential is a singleton. The differentiability statement follows immediately. □

B.5.1 Blockwise subdifferential decomposition

Define the ReLU readout set

$$\mathcal{S}_{\mathrm{R}}(x) := \left\{ \sum_{\ell=1}^{L} W_\ell^\top \nu_\ell : \nu \in \mathcal{D}_{\mathrm{R}}^\star(x) \right\}. \tag{28}$$

Corollary 15 (Blockwise subdifferential decomposition)**.** *For every x,*

$$\partial f(x) = v + \mathcal{S}_{\mathrm{R}}(x) + \sum_{h=1}^{H} \alpha_h B_h^\top q_h(x) + \sum_{g=1}^{G} A_g^\top \mathcal{R}_g^\star(x), \tag{29}$$

where all additions involving sets are Minkowski sums.

Proof. Insert the product representation from Proposition 3 into the affine readout map $\mathcal{G}$. The image of a Cartesian product under a sum of blockwise linear maps is the Minkowski sum of the corresponding block images. The quadratic block is single-valued, while the ReLU and norm blocks may be set-valued. Applying Theorem 4 gives equation 29. □

The decomposition shows that quadratic modules contribute a unique smooth slope, zero ReLU preactivations contribute recursively coupled interval-valued slopes, and zero norm residuals contribute affine images of Euclidean balls.

B.5.2 Context-conditioned extension

Corollary 16 (Fixed-context geometry)**.** *Suppose that the parameters of the SOC-ICNN depend on a context y, while the convexity sign conditions in equation 3 are preserved for every admissible context. For each fixed y, all results in Proposition 3 and Theorem 4 hold for the partial subdifferential with respect to the decision input: $\partial_x f(x;y) = \mathcal{G}_y\big(\mathcal{D}^\star(x;y)\big)$.*

Proof. Fix y. All context-dependent matrices and offsets then become constants, and the resulting function of x has exactly the SOC-ICNN form analyzed above. Applying the preceding results and restoring the dependence on y gives the claim. □

B.6 Stability of the optimal-dual and readout correspondences

Proof of Proposition 5. We prove nonemptiness, compactness, local boundedness, and closedness in turn.

Nonempty compact values. Nonemptiness follows directly from the explicit blockwise characterization in Proposition 3. The ReLU block is a closed subset of the compact set $\mathcal{K}_\nu$, each norm block is either a singleton or a closed Euclidean ball, and each quadratic block is the singleton $\{\alpha_h q_h(x)\}$. Hence $\mathcal{D}^\star(x)$ is compact for every x.

Local boundedness. Fix $\bar{x}$. Let X be a compact neighborhood of $\bar{x}$. The ReLU multipliers lie in the input-independent compact set $\mathcal{K}_\nu$, and the norm multipliers lie in input-independent compact balls. For the quadratic multipliers, $p_h^\star(x) = \alpha_h q_h(x)$. Since q_h is affine, it is bounded on X. Thus there exists a common bounded set containing $\mathcal{D}^\star(x)$ for every $x \in X$.

Closed graph. Suppose $x_n \to x,\ \ \xi_n \to \xi,\ \ \xi_n \in \mathcal{D}^\star(x_n)$. The dual feasible set is closed and independent of x, so ξ is feasible. Since each ξ_n is optimal, $\Psi(x_n;\xi_n) = f(x_n)$. Both f and Ψ are continuous. Passing to the limit gives $\Psi(x;\xi) = f(x)$. Thus $\xi \in \mathcal{D}^\star(x)$, proving that the graph of $\mathcal{D}^\star$ is closed.

We now consider the readout image. Suppose $x_n \to x,\ \ g_n \to g,\ \ g_n \in \mathcal{G}\big(\mathcal{D}^\star(x_n)\big)$. Choose $\xi_n \in \mathcal{D}^\star(x_n)$ such that $g_n = \mathcal{G}(\xi_n)$. Local boundedness gives a convergent subsequence, which we denote by $\xi_{n_j} \to \bar{\xi}$. Closedness of the multiplier graph gives $\bar{\xi} \in \mathcal{D}^\star(x)$. Since $\mathcal{G}$ is continuous,

$$g = \lim_{j\to\infty} \mathcal{G}(\xi_{n_j}) = \mathcal{G}(\bar{\xi}) \in \mathcal{G}\big(\mathcal{D}^\star(x)\big).$$

Hence the readout correspondence has a closed graph. Its values are nonempty and compact because they are continuous images of nonempty compact sets. It is locally bounded because $\mathcal{G}$ is affine and $\mathcal{D}^\star$ is locally bounded. By Theorem 4, this readout correspondence is exactly $x \mapsto \partial f(x)$. □

B.7 Smooth branches and explicit local curvature

Let $\mathcal{I}_+ := \{g \in \{1,\dots,G\} : \lambda_g > 0\}$.

Proof of Proposition 6. Let $\bar{x}$ be structurally nondegenerate; thus every ReLU preactivation is nonzero at $\bar{x}$, and $u_g(\bar{x}) \neq 0$ for every $g \in \mathcal{I}_+$. The finite collection $\{|a_{\ell,i}(\bar{x})| : \ell = 1,\dots,L,\ i\} \cup \{\|u_g(\bar{x})\|_2 : g \in \mathcal{I}_+\}$ contains only strictly positive numbers. Since all preactivations and affine residuals are continuous, there exists $\rho > 0$ such that, for every $x \in \mathbb{B}(\bar{x},\rho)$,

1. each $a_{\ell,i}(x)$ has the same sign as $a_{\ell,i}(\bar{x})$; and 2. $u_g(x) \neq 0$ for every $g \in \mathcal{I}_+$.

Thus the ReLU activation pattern is fixed throughout the neighborhood. Replacing every active ReLU by the identity and every inactive ReLU by zero, a layerwise induction shows that each hidden state is affine in x. Consequently, the entire ReLU-output component, including the affine term $v^\top x + b_0$, can be written as $\bar{a}^\top x + \bar{b}$ for constants $\bar{a} \in \mathbb{R}^d$ and $\bar{b} \in \mathbb{R}$.

The objective therefore has the local form

$$f(x) = \bar{a}^\top x + \bar{b} + \sum_{h=1}^{H} \frac{\alpha_h}{2}\|q_h(x)\|_2^2 + \sum_{g\in\mathcal{I}_+} \lambda_g \|u_g(x)\|_2.$$

The blockwise multiplier characterization also shows that every multiplier is uniquely determined in this neighborhood: no ReLU preactivation is zero, all positive-weight norm residuals are nonzero, zero-weight norm balls reduce to the singleton $\{0\}$, and the quadratic multipliers are always unique. Hence $\mathcal{D}^\star(x)$ is a singleton.

For a quadratic module, $\nabla_x \frac{\alpha_h}{2}\|B_h x + e_h\|_2^2 = \alpha_h B_h^\top q_h(x)$, and $\nabla_x^2 \frac{\alpha_h}{2}\|B_h x + e_h\|_2^2 = \alpha_h B_h^\top B_h$.

For a positive-weight norm module, write $u = u_g(x),\ \ A = A_g,\ \ \widehat{u} = \frac{u}{\|u\|_2}$. Because $u \neq 0$, $\nabla_x \lambda_g \|u\|_2 = \lambda_g A^\top \widehat{u}$. For an arbitrary perturbation δu,

$$D\widehat{u}[\delta u] = D\left(\frac{u}{\|u\|_2}\right)[\delta u] = \frac{\delta u}{\|u\|_2} - \frac{u}{\|u\|_2^2}\frac{u^\top \delta u}{\|u\|_2} = \frac{1}{\|u\|_2}\left(I - \widehat{u}\widehat{u}^\top\right)\delta u.$$

Since $\delta u = A\delta x$, $D\left(\lambda_g A^\top \widehat{u}\right)[\delta x] = \lambda_g A^\top \frac{I - \widehat{u}\widehat{u}^\top}{\|u\|_2} A\delta x$.

Summing all blocks gives

$$\nabla f(x) = \bar{a} + \sum_{h=1}^{H} \alpha_h B_h^\top q_h(x) + \sum_{g\in\mathcal{I}_+} \lambda_g A_g^\top \widehat{u}_g(x),$$

and

$$\nabla^2 f(x) = \sum_{h=1}^{H} \alpha_h B_h^\top B_h + \sum_{g\in\mathcal{I}_+} \lambda_g A_g^\top \frac{I - \widehat{u}_g(x)\widehat{u}_g(x)^\top}{\|u_g(x)\|_2} A_g.$$

Each quadratic term is positive semidefinite. Moreover, $I - \widehat{u}_g\widehat{u}_g^\top$ is the orthogonal projector onto the subspace perpendicular to $\widehat{u}_g$, and is therefore positive semidefinite. Hence $\nabla^2 f(x) \succeq 0$.

It remains to establish local Lipschitz continuity of the Hessian. Choose a closed ball $\overline{\mathbb{B}}(\bar{x}, \rho') \subset \mathbb{B}(\bar{x}, \rho)$. For every $g \in \mathcal{I}_+$, continuity and compactness imply the existence of $\delta_g > 0$ such that

$$\|u_g(x)\|_2 \geq \delta_g \ \ \forall\, x \in \overline{\mathbb{B}}(\bar{x}, \rho').$$

The mapping

$$u \mapsto \frac{I - (u/\|u\|_2)(u/\|u\|_2)^\top}{\|u\|_2}$$

is continuously differentiable on $\mathbb{R}^{k_g} \setminus \{0\}$. The compact convex image of $\overline{\mathbb{B}}(\bar{x}, \rho')$ under the affine map u_g stays a positive distance from the origin. The derivative is therefore bounded on this image, and the mean-value inequality shows that the mapping is Lipschitz there. Composition with the affine map $u_g(x)$, multiplication by fixed matrices, and finite summation preserve Lipschitz continuity. The quadratic contribution to the Hessian is constant. Therefore $\nabla^2 f$ is Lipschitz continuous on every sufficiently small closed ball contained in the nondegenerate neighborhood. □

B.8 Geometry of the complete inference objective

Since $\Phi : \mathbb{R}^d \to \mathbb{R}$ is finite, convex, and continuously differentiable, the convex sum rule gives

$$\partial F(x) = \partial f(x) + \nabla\Phi(x) = \mathcal{G}\big(\mathcal{D}^\star(x)\big) + \nabla\Phi(x).$$

If Φ is twice continuously differentiable on a structurally nondegenerate neighborhood, then $\nabla F(x) = \nabla f(x) + \nabla\Phi(x)$, and $\nabla^2 F(x) = \nabla^2 f(x) + \nabla^2\Phi(x)$. If, in addition, $\nabla^2\Phi$ is locally Lipschitz, then $\nabla^2 F$ is locally Lipschitz on the same neighborhood.

C Proofs for Dual-Certified Inference

This section provides complete proofs for the constrained DCI results in Section 4. Throughout, $\mathcal{X} \subseteq \mathbb{R}^d$ is a nonempty closed convex set and $F(x) = f(x) + \Phi(x)$ is finite and convex on $\mathbb{R}^d$. The feasible retraction $\mathsf{P}_{\mathcal{X}}$ satisfies equation 10. We use the normal-cone convention

$$N_{\mathcal{X}}(x) = \left\{ n \in \mathbb{R}^d : n^\top (z - x) \leq 0 \ \forall\, z \in \mathcal{X} \right\},$$

and define the tangent cone by $T_{\mathcal{X}}(x) = \operatorname{cl}\left\{ t(z - x) : t \geq 0,\ z \in \mathcal{X} \right\}$. For a closed convex set,

$$T_{\mathcal{X}}(x) = N_{\mathcal{X}}(x)^\circ = \left\{ d : n^\top d \leq 0 \ \forall\, n \in N_{\mathcal{X}}(x) \right\}. \tag{30}$$

C.1 The constrained first-order set

Recall that $\partial F(x) = \mathcal{G}\big(\mathcal{D}^\star(x)\big) + \nabla\Phi(x)$. For $x \in \mathcal{X}$, define $\mathcal{C}_{\mathcal{X}}(x) := \partial F(x) + N_{\mathcal{X}}(x)$.

Lemma 17 (Geometry of the constrained first-order set)**.** *For every $x \in \mathcal{X}$, the set $\mathcal{C}_{\mathcal{X}}(x)$ is nonempty, closed, and convex. Consequently, for every $M \succ 0$, the metric projection*

$$p_M(x) := \operatorname*{arg\,min}_{p \in \mathcal{C}_{\mathcal{X}}(x)} \frac{1}{2}\|p\|^2_{M^{-1}} \tag{31}$$

exists and is unique.

Proof. Because F is finite and convex on $\mathbb{R}^d$, $\partial F(x)$ is nonempty, compact, and convex for every x. The normal cone $N_{\mathcal{X}}(x)$ is a nonempty closed convex cone because it contains the origin.

The Minkowski sum of two convex sets is convex. Moreover, the sum of a compact set and a closed set is closed in finite-dimensional space. Therefore, $\mathcal{C}_{\mathcal{X}}(x) = \partial F(x) + N_{\mathcal{X}}(x)$ is nonempty, closed, and convex.

The function $p \mapsto \frac{1}{2} p^\top M^{-1} p$ is continuous, coercive, and strictly convex because $M \succ 0$. It therefore has a unique minimizer over the nonempty closed convex set $\mathcal{C}_{\mathcal{X}}(x)$. □

By equation 4, every $s \in \partial F(x)$ can be represented as $s = \mathcal{G}(\xi) + \nabla\Phi(x)$ for some $\xi \in \mathcal{D}^\star(x)$. Hence the projection in equation 31 is equivalent to

$$\min_{\substack{\xi \in \mathcal{D}^\star(x) \\ n \in N_{\mathcal{X}}(x)}} \frac{1}{2} \left\| \mathcal{G}(\xi) + \nabla\Phi(x) + n \right\|_{M^{-1}}^2 . \tag{32}$$

Although the optimizing pair (ξ, n) may be nonunique, its readout $p_M(x) = \mathcal{G}(\xi) + \nabla\Phi(x) + n$ is unique. The notation in the main text satisfies $g_M^\dagger(x) = p_M(x),\ \mathcal{R}_M(x) = \|p_M(x)\|_{M^{-1}}$.

C.2 Constrained certificate and common descent

Proof of Theorem 7. Fix $x \in \mathcal{X}$, let $p := g_M^\dagger(x),\ q := M^{-1}p,\ d := -q$.

Exact stationarity certificate. By definition, $\mathcal{R}_M(x) = 0 \iff p = 0 \iff 0 \in \partial F(x) + N_{\mathcal{X}}(x)$. The convex Fermat rule gives $0 \in \partial F(x) + N_{\mathcal{X}}(x) \iff x \in \arg\min_{z \in \mathcal{X}} F(z)$. This proves $\mathcal{R}_M(x) = 0 \iff x \in \arg\min_{z \in \mathcal{X}} F(z)$.

Projection variational inequality. Since p is the metric projection of the origin onto $\mathcal{C}_{\mathcal{X}}(x)$, it satisfies

$$(c - p)^\top M^{-1} p \geq 0 \ \ \forall\, c \in \mathcal{C}_{\mathcal{X}}(x). \tag{33}$$

For an arbitrary $g \in \partial F(x)$, the vector

$$c = g + 0$$

belongs to $\mathcal{C}_{\mathcal{X}}(x)$, since $0 \in N_{\mathcal{X}}(x)$. Substituting $c = g$ into equation 33 gives

$$g^\top M^{-1} p \geq p^\top M^{-1} p = \mathcal{R}_M(x)^2.$$

Because $d = -M^{-1}p$,

$$g^\top d \leq -\mathcal{R}_M(x)^2 \ \ \forall\, g \in \partial F(x),$$

which proves equation 8.

Tangential feasibility. Choose one decomposition

$$p = \bar{g} + \bar{n},\ \ \bar{g} \in \partial F(x),\ \ \bar{n} \in N_{\mathcal{X}}(x).$$

For any $n \in N_{\mathcal{X}}(x)$, the vector $\bar{g} + n$ belongs to $\mathcal{C}_{\mathcal{X}}(x)$. Applying equation 33 with $c = \bar{g} + n$ gives

$$(n - \bar{n})^\top q \geq 0 \ \ \forall\, n \in N_{\mathcal{X}}(x).$$

Taking $n = 0$ yields

$$\bar{n}^\top q \leq 0,$$

whereas taking $n = 2\bar{n}$ yields

$$\bar{n}^\top q \geq 0.$$

Therefore,

$$\bar{n}^\top q = 0.$$

Returning to the general inequality gives

$$n^\top q \geq 0 \ \ \forall\, n \in N_{\mathcal{X}}(x).$$

Equivalently,

$$n^\top d \leq 0 \ \ \forall\, n \in N_{\mathcal{X}}(x).$$

By equation 30,

$$d \in T_{\mathcal{X}}(x).$$

Directional descent. For a finite convex function,

$$F'(x;d) = \max_{g \in \partial F(x)} g^\top d.$$

The common-descent inequality therefore implies

$$F'(x;d) \le -\mathcal{R}_M(x)^2.$$

If $\mathcal{R}_M(x) > 0$, the right-hand side is strictly negative, proving

$$F'\big(x; d_M(x)\big) \le -\mathcal{R}_M(x)^2 < 0.$$

□

Corollary 18 (Residual objective-gap bound)**.** *Suppose that F is μ_M-strongly convex on $\mathcal{X}$ with respect to the M-norm. Then, with $F^\star := \min_{z \in \mathcal{X}} F(z)$,*

$$0 \le F(x) - F^\star \le \frac{\mathcal{R}_M(x)^2}{2\mu_M}.$$

Proof. Let $p = g_M^\dagger(x)$ and choose a decomposition $p = \bar{g} + \bar{n}$ with $\bar{g} \in \partial F(x)$ and $\bar{n} \in N_{\mathcal{X}}(x)$. Strong convexity gives, for every $z \in \mathcal{X}$,

$$F(z) \ge F(x) + \bar{g}^\top (z - x) + \frac{\mu_M}{2}\|z - x\|_M^2.$$

Since $\bar{n}^\top (z - x) \le 0$, we have

$$\bar{g}^\top (z - x) = p^\top (z - x) - \bar{n}^\top (z - x) \ge p^\top (z - x).$$

Hence

$$F(z) \ge F(x) + p^\top (z - x) + \frac{\mu_M}{2}\|z - x\|_M^2.$$

Minimizing the right-hand side over all $z \in \mathbb{R}^d$, which can only lower the minimum relative to restricting z to $\mathcal{X}$, gives

$$\begin{aligned} F^\star &\ge F(x) + \min_{s \in \mathbb{R}^d} \left\{ p^\top s + \frac{\mu_M}{2} s^\top M s \right\} \\ &= F(x) - \frac{1}{2\mu_M} p^\top M^{-1} p \\ &= F(x) - \frac{\mathcal{R}_M(x)^2}{2\mu_M}. \end{aligned}$$

Rearranging proves the claim. □

C.2.1 Equivalent constrained steepest-descent problem

The following proposition establishes the equivalence between the direction model in equation 6 and the joint projection in equation 7a.

Proposition 19 (Metric steepest feasible descent)**.** *Let $p = g_M^\dagger(x)$ and $d_M(x) = -M^{-1}p$. Then*

$$d_M(x) = \operatorname*{arg\,min}_{d \in T_{\mathcal{X}}(x)} \left\{ F'(x;d) + \frac{1}{2}\|d\|_M^2 \right\}. \tag{34}$$

The optimal value is

$$-\frac{1}{2}\mathcal{R}_M(x)^2.$$

Proof. For a nonempty set $S \subseteq \mathbb{R}^d$, let $\sigma_S(d) := \sup_{s \in S} s^\top d$ denote its support function, and let δ_S denote its indicator function, equal to zero on S and $+\infty$ elsewhere. Because

$$T_{\mathcal{X}}(x) = N_{\mathcal{X}}(x)^\circ,$$

the support function of the normal cone satisfies

$$\sigma_{N_{\mathcal{X}}(x)}(d) = \begin{cases} 0, & d \in T_{\mathcal{X}}(x), \\ +\infty, & d \notin T_{\mathcal{X}}(x). \end{cases}$$

Therefore,

$$\begin{aligned} F'(x;d) + \delta_{T_{\mathcal{X}}(x)}(d) &= \sigma_{\partial F(x)}(d) + \sigma_{N_{\mathcal{X}}(x)}(d) \\ &= \sigma_{\mathcal{C}_{\mathcal{X}}(x)}(d). \end{aligned}$$

Thus the problem in equation 34 is equivalent to

$$\min_{d \in \mathbb{R}^d} \left\{ \sigma_{\mathcal{C}_{\mathcal{X}}(x)}(d) + \frac{1}{2}\|d\|_M^2 \right\}.$$

For every d,

$$\sigma_{\mathcal{C}_{\mathcal{X}}(x)}(d) \geq p^\top d.$$

Hence

$$\sigma_{\mathcal{C}_{\mathcal{X}}(x)}(d) + \frac{1}{2}\|d\|_M^2 \geq p^\top d + \frac{1}{2} d^\top M d \geq -\frac{1}{2} p^\top M^{-1} p.$$

At $d = -M^{-1}p$, the projection variational inequality implies

$$c^\top d \leq p^\top d \ \ \forall\, c \in \mathcal{C}_{\mathcal{X}}(x).$$

Since $p \in \mathcal{C}_{\mathcal{X}}(x)$, this gives

$$\sigma_{\mathcal{C}_{\mathcal{X}}(x)}(d) = p^\top d.$$

Therefore the lower bound is attained at $d = -M^{-1}p$, with value

$$-\frac{1}{2} p^\top M^{-1} p = -\frac{1}{2} \mathcal{R}_M(x)^2.$$

Strict convexity of the quadratic term gives uniqueness. □

C.3 Boundary-local projection

Assume that, at the current point x, the feasible-set normal cone admits the representation

$$N_{\mathcal{X}}(x) = \{ B_{\mathcal{X}}(x)\zeta : \zeta \in \mathcal{K}_{\mathcal{X}}(x) \}, \tag{35}$$

where $B_{\mathcal{X}}(x) \in \mathbb{R}^{d \times q_{\mathcal{X}}(x)}$ and $\mathcal{K}_{\mathcal{X}}(x) \subseteq \mathbb{R}^{q_{\mathcal{X}}(x)}$ is a simple closed convex cone. When $N_{\mathcal{X}}(x) = \{0\}$, we take $q_{\mathcal{X}}(x) = 0$.

For a polyhedron $\mathcal{X} = \{x \in \mathbb{R}^d : Cx \leq h\}$, let $\mathcal{I}_{\mathcal{X}}(x) = \{j : C_j x = h_j\}$. Then

$$N_{\mathcal{X}}(x) = \left\{ C_{\mathcal{I}_{\mathcal{X}}(x)}^\top \zeta : \zeta \in \mathbb{R}_+^{|\mathcal{I}_{\mathcal{X}}(x)|} \right\}. \tag{36}$$

Thus, $B_{\mathcal{X}}(x) = C_{\mathcal{I}_{\mathcal{X}}(x)}^\top$, $\mathcal{K}_{\mathcal{X}}(x) = \mathbb{R}_+^{|\mathcal{I}_{\mathcal{X}}(x)|}$, and $q_{\mathcal{X}}(x) = |\mathcal{I}_{\mathcal{X}}(x)|$.

For the Euclidean ball $\mathbb{B}_r := \{x : \|x\|_2 \leq r\}$ with $r > 0$,

$$N_{\mathbb{B}_r}(x) = \begin{cases} \{0\}, & \|x\|_2 < r, \\ \{\alpha x : \alpha \geq 0\}, & \|x\|_2 = r. \end{cases} \tag{37}$$

Hence a boundary point contributes one nonnegative scalar multiplier.

For the Lorentz cone $\mathcal{Q}^d = \{(t,u) \in \mathbb{R} \times \mathbb{R}^{d-1} : \|u\|_2 \leq t\}$,

$$N_{\mathcal{Q}^d}(t,u) = \begin{cases} \{0\}, & \|u\|_2 < t, \\ \{\alpha\,(-1,\, u/\|u\|_2) : \alpha \geq 0\}, & t = \|u\|_2 > 0, \\ -\mathcal{Q}^d, & (t,u) = 0. \end{cases} \tag{38}$$

Thus, a nonzero Lorentz-cone boundary point contributes one nonnegative scalar, while the apex contributes one full cone block.

Proof of Proposition 8. By Proposition 3, each quadratic multiplier is uniquely determined by

$$p_h = \alpha_h q_h(x).$$

Likewise, if $u_g(x) \neq 0$, then

$$r_g = \lambda_g \frac{u_g(x)}{\|u_g(x)\|_2}$$

is uniquely determined. If $u_g(x) = 0$ and $\lambda_g = 0$, then its dual ball is the singleton $\{0\}$. Only indices

$$g \in \mathcal{Z}_{\mathrm{N}}(x)$$

leave a free vector

$$r_g \in \mathbb{R}^{k_g}, \ \ \|r_g\|_2 \leq \lambda_g.$$

These blocks contribute

$$\sum_{g \in \mathcal{Z}_{\mathrm{N}}(x)} k_g$$

scalar variables.

For the ReLU block, the elimination proceeds backward through the network. At layer L,

- a negative preactivation fixes $\nu_{L,i} = 0$; - a positive preactivation fixes $\nu_{L,i} = c_i$; - a zero preactivation leaves one scalar variable in $[0, c_i]$.

Suppose all nonboundary variables in layer $\ell + 1$ have already been eliminated as affine functions of the surviving zero-preactivation variables. At layer ℓ,

- if $a_{\ell,i}(x) < 0$, then $\nu_{\ell,i} = 0$; - if $a_{\ell,i}(x) > 0$, then

$$\nu_{\ell,i} = \left(U_{\ell+1}^{\top} \nu_{\ell+1}\right)_i,$$

so it can be eliminated as an affine function of later-layer free variables; - if $a_{\ell,i}(x) = 0$, then one scalar variable remains, satisfying

$$0 \leq \nu_{\ell,i} \leq \left(U_{\ell+1}^{\top} \nu_{\ell+1}\right)_i.$$

Thus, the ReLU block leaves at most one scalar variable for every coordinate in $\mathcal{Z}_{\mathrm{R}}(x)$. Multipliers associated with strictly positive preactivations are affine functions of later-layer free variables and introduce no additional decision variables.

By equation 35, the feasible-set normal component is parameterized by $\zeta \in \mathcal{K}_{\mathcal{X}}(x) \subseteq \mathbb{R}^{q_{\mathcal{X}}(x)}$, and therefore contributes at most $q_{\mathcal{X}}(x)$ scalar variables. After all substitutions, the projected vector is

$$\mathcal{G}(\xi) + \nabla\Phi(x) + B_{\mathcal{X}}(x)\zeta,$$

which is affine in the surviving variables. Its squared M^{-1}-norm is therefore convex quadratic, and the feasible set consists of recursive affine intervals, Euclidean balls, and the cone $\mathcal{K}_{\mathcal{X}}(x)$.

The total number of scalar variables is at most

$$|\mathcal{Z}_{\mathrm{R}}(x)| + \sum_{g \in \mathcal{Z}_{\mathrm{N}}(x)} k_g + q_{\mathcal{X}}(x).$$

If $\mathcal{Z}_{\mathrm{R}}(x) = \mathcal{Z}_{\mathrm{N}}(x) = \varnothing$ and $q_{\mathcal{X}}(x) = 0$, then every network multiplier is uniquely determined, the normal cone is $\{0\}$, and no projection optimization is required. □

C.4 Finite line search and proximal safeguard

We first record why Euclidean projection is an admissible feasible retraction.

Lemma 20 (First-order consistency of Euclidean projection)**.** *Let $\mathcal{X}$ be nonempty, closed, and convex, $x \in \mathcal{X}$, and $d \in T_{\mathcal{X}}(x)$. Then*

$$\Pi_{\mathcal{X}}(x + td) = x + td + o(t) \ \ \textit{as } t \downarrow 0. \tag{39}$$

Moreover, $\Pi_{\mathcal{X}}(y) = y$ for every $y \in \mathcal{X}$.

Proof. For a closed convex set, the tangent-cone characterization gives

$$d \in T_{\mathcal{X}}(x) \iff \operatorname{dist}(x + td, \mathcal{X}) = o(t) \text{ as } t \downarrow 0.$$

Since

$$\|\Pi_{\mathcal{X}}(x + td) - (x + td)\|_2 = \operatorname{dist}(x + td, \mathcal{X}),$$

equation 39 follows. The second statement is the defining fixed-point property of the Euclidean projection. □

Lemma 21 (Existence of a projected directional Armijo step)**.** *Let $x \in \mathcal{X}$ satisfy $\mathcal{R}_M(x) > 0$, let*

$$d = -M^{-1} g_M^{\dagger}(x),$$

and define

$$z(t) = \mathsf{P}_{\mathcal{X}}(x, td), \quad s(t) = z(t) - x.$$

For every $\eta, \sigma \in (0, 1)$, there exists $\bar{t} > 0$ such that

$$F'(x; s(t)) \leq -\eta t \mathcal{R}_M(x)^2 \tag{40}$$

and

$$F(z(t)) \leq F(x) + \sigma F'(x; s(t)) \tag{41}$$

for every $t \in (0, \bar{t}]$.

Proof. By Theorem 7,

$$d \in T_{\mathcal{X}}(x)$$

and

$$F'(x; d) \leq -\mathcal{R}_M(x)^2 < 0.$$

The first-order consistency of the retraction gives

$$\frac{s(t)}{t} \longrightarrow d.$$

Because F is finite and convex on $\mathbb{R}^d$,

$$F'(x; v) = \max_{g \in \partial F(x)} g^{\top} v$$

is the support function of the nonempty compact set $\partial F(x)$, and is therefore finite, positively homogeneous, and continuous in v. Consequently,

$$\frac{F'(x; s(t))}{t} = F'\left(x; \frac{s(t)}{t}\right) \longrightarrow F'(x; d) \leq -\mathcal{R}_M(x)^2.$$

Since $\eta < 1$, equation 40 holds for all sufficiently small $t > 0$.

A finite convex function is Hadamard directionally differentiable. Hence, whenever $s(t)/t \to d$,

$$\frac{F(x + s(t)) - F(x)}{t} \longrightarrow F'(x; d).$$

Together with the preceding limit,

$$\frac{F(x + s(t)) - F(x) - \sigma F'(x; s(t))}{t} \longrightarrow (1 - \sigma) F'(x; d) < 0.$$

Thus equation 41 also holds for all sufficiently small $t > 0$. Taking the smaller of the two resulting thresholds proves the claim. □

Thus, an unlimited projected backtracking search would always terminate. DCI uses a finite backtracking budget to bound the local search cost and invokes the proximal safeguard if the tested sequence does not reach a sufficiently small step.

Lemma 22 (Proximal safeguard). *For $x_k \in \mathcal{X}$, $M_k \succ 0$, and $\gamma > 0$, the problem*

$$x_{k+1} = \operatorname*{arg\,min}_{z \in \mathcal{X}} \left\{ F(z) + \frac{1}{2\gamma} \|z - x_k\|_{M_k}^2 \right\} \tag{42}$$

has a unique solution. This solution satisfies

$$\frac{1}{\gamma} M_k (x_k - x_{k+1}) \in \partial F(x_{k+1}) + N_{\mathcal{X}}(x_{k+1}), \tag{43}$$

and

$$F(x_k) - F(x_{k+1}) \geq \frac{1}{2\gamma} \|x_{k+1} - x_k\|_{M_k}^2. \tag{44}$$

In particular,

$$\mathcal{R}_{M_k}(x_{k+1}) \leq \frac{1}{\gamma} \|x_{k+1} - x_k\|_{M_k}. \tag{45}$$

Proof. The proximal objective is strongly convex because $M_k \succ 0$. It is also coercive on $\mathcal{X}$. Indeed, since F is finite and convex, choose $g_k \in \partial F(x_k)$. Then

$$F(z) \geq F(x_k) + g_k^\top (z - x_k),$$

so the proximal objective is bounded below by an affine function plus a positive-definite quadratic. It therefore tends to $+\infty$ as $\|z\|_2 \to \infty$.

Since $\mathcal{X}$ is nonempty and closed, a minimizer exists. Strong convexity makes it unique.

The first-order optimality condition is

$$0 \in \partial F(x_{k+1}) + N_{\mathcal{X}}(x_{k+1}) + \frac{1}{\gamma} M_k (x_{k+1} - x_k),$$

which is equivalent to equation 43.

Because $x_k \in \mathcal{X}$ is a feasible comparison point,

$$F(x_{k+1}) + \frac{1}{2\gamma} \|x_{k+1} - x_k\|_{M_k}^2 \leq F(x_k).$$

Rearranging gives equation 44.

By equation 43, the vector $\gamma^{-1} M_k (x_k - x_{k+1})$ belongs to the constrained first-order set at x_{k+1}. Therefore,

$$\mathcal{R}_{M_k}(x_{k+1}) \leq \left\| \gamma^{-1} M_k (x_k - x_{k+1}) \right\|_{M_k^{-1}} = \gamma^{-1} \|x_{k+1} - x_k\|_{M_k},$$

which proves equation 45. □

C.5 Complete safeguarded DCI iteration

For the global theory, DCI only requires a sequence of metrics satisfying equation 9. On a structurally nondegenerate iterate where Φ is twice differentiable, the explicit network curvature gives

$$H_k := \nabla^2 f(x_k) + \nabla^2 \Phi(x_k) = \nabla^2 F(x_k).$$

A curvature-based choice is

$$M_k = \operatorname{clip}_{[m, \overline{m}]}\big(H_k + \tau_k I\big), \quad \tau_k \geq 0,$$

where $\operatorname{clip}_{[m, \overline{m}]}$ denotes spectral clipping of the eigenvalues to $[m, \overline{m}]$. When this curvature is unavailable, any positive-definite fallback satisfying equation 9 may be used. The complete safeguarded iteration is summarized below.

Algorithm 1 Safeguarded Dual-Certified Inference

Require: $x_0 \in \mathcal{X}$, tolerance $\varepsilon \geq 0$, $0 < \sigma < 1/2$, $0 < \eta < 1$, $0 < \beta < 1$, $J_{\max} \geq 0$, and $\gamma > 0$.

for $k = 0, 1, \ldots$ **do**
  Evaluate the SOC-ICNN at x_k and construct $\mathcal{D}^\star(x_k)$ from the forward-pass states.
  Construct a representation of $N_{\mathcal{X}}(x_k)$ and choose $M_k \succ 0$ satisfying equation 9.
  Solve equation 7a to obtain $g_k^\dagger = g_{M_k}^\dagger(x_k)$ and $\mathcal{R}_k = \|g_k^\dagger\|_{M_k^{-1}}$.
  if $\mathcal{R}_k \leq \varepsilon$ **then**
    return x_k.
  end if
  Set $d_k = -M_k^{-1} g_k^\dagger$.
  Set accepted $\leftarrow$ false.
  for $j = 0, \ldots, J_{\max}$ **do**
    Set $t = \beta^j$, $z_k(t) = \mathsf{P}_{\mathcal{X}}(x_k, t d_k)$, and $s_k(t) = z_k(t) - x_k$.
    if both conditions in equation 11 hold **then**
      Set $x_{k+1} \leftarrow z_k(t)$ and accepted $\leftarrow$ true.
      break.
    end if
  end for
  if accepted $=$ false **then**
    Compute x_{k+1} by the metric-proximal safeguard equation 12.
  end if
end for

C.6 Closedness of constrained stationarity

The geometry appendix establishes the closed-graph property of $x \mapsto \partial F(x)$. The constrained convergence proof additionally requires stability after adding the normal cone of a general closed convex feasible set.

Lemma 23 (Closed graph of the constrained first-order mapping)**.** *Let $x_j \to x$ with $x_j, x \in \mathcal{X}$, and let*

$$c_j \in \partial F(x_j) + N_{\mathcal{X}}(x_j), \;\; c_j \to c.$$

Then

$$c \in \partial F(x) + N_{\mathcal{X}}(x).$$

Proof. Choose decompositions

$$c_j = g_j + n_j, \;\; g_j \in \partial F(x_j), \;\; n_j \in N_{\mathcal{X}}(x_j).$$

Because F is finite and convex on $\mathbb{R}^d$, its subdifferential is locally bounded. Since $x_j \to x$, the sequence $\{g_j\}$ is bounded. Passing to a subsequence if necessary, let

$$g_j \to g.$$

The closed-graph property of the convex subdifferential gives

$$g \in \partial F(x).$$

Moreover,

$$n_j = c_j - g_j \to c - g =: n.$$

For every $z \in \mathcal{X}$,

$$n_j^\top (z - x_j) \leq 0.$$

Passing to the limit gives

$$n^\top (z - x) \leq 0 \;\; \forall\, z \in \mathcal{X}.$$

Hence

$$n \in N_{\mathcal{X}}(x).$$

Therefore

$$c = g + n \in \partial F(x) + N_{\mathcal{X}}(x).$$

Since every convergent subsequence yields the same conclusion, the entire mapping has a closed graph. □

C.7 GLOBAL CONVERGENCE

Proof of the global part of Theorem 9. If the algorithm terminates finitely with $\varepsilon = 0$, then $\mathcal{R}_k = 0$, and Theorem 7 proves global optimality. We therefore consider an infinite run. Let

$$\mathcal{A} = \{k : \text{iteration } k \text{ accepts a projected local step}\},$$

and

$$\mathcal{S} = \{k : \text{iteration } k \text{ invokes the proximal safeguard}\}.$$

Monotonicity and boundedness. If $k \in \mathcal{A}$, let t_k be the accepted step and $s_k = s_k(t_k)$. The two line-search conditions give

$$\begin{aligned} F(x_{k+1}) &\le F(x_k) + \sigma F'(x_k; s_k) \\ &\le F(x_k) - \sigma\eta t_k \mathcal{R}_k^2 \le F(x_k). \end{aligned}$$

If $k \in \mathcal{S}$, Lemma 22 gives

$$F(x_{k+1}) \le F(x_k).$$

Thus

$$F(x_{k+1}) \le F(x_k)$$

for every k, and all iterates remain in the compact level set $\mathcal{L}_0$.

Since F is continuous and $\mathcal{L}_0$ is compact, F is bounded below on $\mathcal{L}_0$. Therefore the nonincreasing sequence $\{F(x_k)\}$ converges to some value $\bar{F}$.

We consider two cases.

Case 1: infinitely many accepted local steps. For every $k \in \mathcal{A}$,

$$t_k \ge \beta^{J_{\max}},$$

and therefore

$$F(x_k) - F(x_{k+1}) \ge \sigma\eta\beta^{J_{\max}} \mathcal{R}_k^2.$$

Summing over accepted iterations gives

$$\sigma\eta\beta^{J_{\max}} \sum_{k \in \mathcal{A}} \mathcal{R}_k^2 \le F(x_0) - \inf_{\mathcal{L}_0} F < \infty.$$

Hence

$$\mathcal{R}_k \to 0 \ \text{ along } k \in \mathcal{A}.$$

Let

$$p_k = g_{M_k}^{\dagger}(x_k).$$

The metric bound $M_k \preceq \overline{m} I$ implies

$$\mathcal{R}_k^2 = p_k^\top M_k^{-1} p_k \ge \frac{1}{\overline{m}} \|p_k\|_2^2.$$

Thus

$$p_k \to 0 \ \text{ along } k \in \mathcal{A}.$$

Compactness of $\mathcal{L}_0$ gives a subsequence

$$x_{k_j} \to \bar{x}, \ \ k_j \in \mathcal{A}.$$

Since

$$p_{k_j} \in \partial F(x_{k_j}) + N_{\mathcal{X}}(x_{k_j})$$

and $p_{k_j} \to 0$, Lemma 23 gives

$$0 \in \partial F(\bar{x}) + N_{\mathcal{X}}(\bar{x}).$$

Therefore $\bar{x}$ is a global minimizer of F over $\mathcal{X}$. Continuity gives

$$\bar{F} = \lim_{j \to \infty} F(x_{k_j}) = F(\bar{x}) = F^\star.$$

Case 2: only finitely many accepted local steps. In this case, every sufficiently large iteration invokes the proximal safeguard. From equation 44,

$$F(x_k) - F(x_{k+1}) \geq \frac{1}{2\gamma}\|x_{k+1} - x_k\|_{M_k}^2.$$

Summing over all sufficiently large k yields

$$\sum_k \|x_{k+1} - x_k\|_{M_k}^2 < \infty.$$

Using $M_k \succeq mI$,

$$\|x_{k+1} - x_k\|_2 \to 0.$$

Define the proximal residual

$$r_{k+1} = \frac{1}{\gamma} M_k(x_k - x_{k+1}).$$

By equation 43,

$$r_{k+1} \in \partial F(x_{k+1}) + N_{\mathcal{X}}(x_{k+1}).$$

Moreover,

$$\|r_{k+1}\|_2 \leq \frac{\overline{m}}{\gamma}\|x_{k+1} - x_k\|_2 \to 0.$$

Compactness gives a subsequence

$$x_{k_j+1} \to \bar{x}.$$

Applying Lemma 23 yields

$$0 \in \partial F(\bar{x}) + N_{\mathcal{X}}(\bar{x}).$$

Thus $\bar{x}$ is globally optimal and, as in the first case,

$$\bar{F} = F^\star.$$

All accumulation points are optimal. In both cases, the full objective sequence satisfies

$$F(x_k) \to F^\star.$$

Let $\widetilde{x}$ be any accumulation point, with $x_{k_j} \to \widetilde{x}$. Continuity gives

$$F(\widetilde{x}) = \lim_{j\to\infty} F(x_{k_j}) = F^\star.$$

Hence every accumulation point is a global solution of $\min_{x\in\mathcal{X}} F(x)$. □

Corollary 24 (Convergence under uniqueness)**.** *Under the assumptions of the global part of Theorem 9, if the constrained minimizer $x^\star$ is unique, then*

$$x_k \to x^\star.$$

Proof. The sequence lies in the compact set $\mathcal{L}_0$, and every accumulation point is a global minimizer. Uniqueness implies that $x^\star$ is the only possible accumulation point. A bounded sequence with a unique accumulation point converges to that point. □

C.8 Interior Newton regime and quadratic convergence

Proof of the local part of Theorem 9. By Corollary 24,

$$x_k \to x^\star.$$

Because $x^\star \in \operatorname{int}\mathcal{X}$, there exists a convex neighborhood $\mathcal{N}_0$ of $x^\star$ whose closure is contained in $\operatorname{int}\mathcal{X}$. Hence, for all sufficiently large k,

$$N_{\mathcal{X}}(x_k) = \{0\}.$$

Structural nondegeneracy provides a neighborhood $\mathcal{N}_1$ in which the ReLU activation pattern is fixed, all positive-weight norm residuals are nonzero, and the optimal-dual set $\mathcal{D}^\star(x)$ is a singleton. On

$$\mathcal{N} = \mathcal{N}_0 \cap \mathcal{N}_1,$$

the complete readout therefore equals the ordinary gradient:

$$g^\dagger_{M_k}(x_k) = \nabla F(x_k).$$

Shrink $\mathcal{N}$, if necessary, to a convex ball contained in the neighborhoods where the Hessian metric is used and the retraction is exact, and choose constants $\mu_H, L_H > 0$ satisfying the following bounds.

$$\nabla^2 F(x) \succeq \mu_H I \ \ \forall\, x \in \mathcal{N}, \tag{46}$$

and

$$\|\nabla^2 F(x) - \nabla^2 F(z)\|_2 \le L_H \|x - z\|_2 \ \ \forall\, x, z \in \mathcal{N}. \tag{47}$$

Such constants exist by positive definiteness at $x^\star$, continuity, and local Lipschitz continuity of the Hessian.

For all sufficiently large k,

$$M_k = H_k := \nabla^2 F(x_k),$$

and the DCI direction is exactly the Newton direction

$$d_k = -H_k^{-1} \nabla F(x_k). \tag{48}$$

The residual satisfies

$$\mathcal{R}_k^2 = \nabla F(x_k)^\top H_k^{-1} \nabla F(x_k) = d_k^\top H_k d_k = -\nabla F(x_k)^\top d_k. \tag{49}$$

Eventual acceptance of the unit step. Continuity of ∇F, the identity $\nabla F(x^\star) = 0$, and the uniform inverse bound from equation 46 imply $d_k = -H_k^{-1}\nabla F(x_k) \to 0$. Hence both x_k and $x_k + d_k$ eventually lie in $\mathcal{N}$. Its convexity ensures that the segment joining them also lies in $\mathcal{N}$. Local exactness of the retraction therefore gives

$$z_k(1) = x_k + d_k, \ \ s_k(1) = d_k.$$

Because F is differentiable in $\mathcal{N}$,

$$F'(x_k; s_k(1)) = \nabla F(x_k)^\top d_k = -\mathcal{R}_k^2 \le -\eta \mathcal{R}_k^2,$$

so the slope test holds for every $\eta \in (0, 1)$.

The Lipschitz-Hessian Taylor estimate gives

$$F(x_k + d_k) \le F(x_k) + \nabla F(x_k)^\top d_k + \frac{1}{2} d_k^\top H_k d_k + \frac{L_H}{6} \|d_k\|_2^3.$$

Using equation 49,

$$F(x_k + d_k) \le F(x_k) - \frac{1}{2}\mathcal{R}_k^2 + \frac{L_H}{6}\|d_k\|_2^3.$$

From equation 46,

$$\mathcal{R}_k^2 = d_k^\top H_k d_k \ge \mu_H \|d_k\|_2^2.$$

Therefore,

$$\frac{L_H}{6}\|d_k\|_2^3 \le \frac{L_H}{6\mu_H}\|d_k\|_2 \mathcal{R}_k^2.$$

Since $d_k \to 0$ and $\sigma < 1/2$, for all sufficiently large k,

$$\frac{L_H}{6\mu_H}\|d_k\|_2 \le \frac{1}{2} - \sigma.$$

Hence

$$F(x_k + d_k) \le F(x_k) - \sigma \mathcal{R}_k^2 = F(x_k) + \sigma F'(x_k; s_k(1)).$$

Both line-search conditions are therefore satisfied at $t = 1$, so unit steps are eventually accepted.

Quadratic error estimate. Let

$$e_k = x_k - x^\star.$$

Because $\nabla F(x^\star) = 0$,

$$\nabla F(x_k) = \int_0^1 \nabla^2 F(x^\star + te_k) e_k \, dt.$$

Using the Newton update,

$$\begin{aligned} e_{k+1} &= e_k - H_k^{-1} \nabla F(x_k) \\ &= H_k^{-1} \int_0^1 \left[H_k - \nabla^2 F(x^\star + te_k) \right] e_k \, dt. \end{aligned}$$

By equation 46,

$$\|H_k^{-1}\|_2 \le \frac{1}{\mu_H}.$$

By equation 47,

$$\left\| H_k - \nabla^2 F(x^\star + te_k) \right\|_2 \le L_H (1-t) \|e_k\|_2.$$

Therefore,

$$\begin{aligned} \|e_{k+1}\|_2 &\le \frac{1}{\mu_H} \int_0^1 L_H (1-t) \|e_k\|_2^2 \, dt \\ &= \frac{L_H}{2\mu_H} \|e_k\|_2^2. \end{aligned}$$

Thus

$$\|x_{k+1} - x^\star\|_2 \le C \|x_k - x^\star\|_2^2$$

with

$$C = \frac{L_H}{2\mu_H},$$

which proves $\|x_{k+1} - x^\star\|_2 \le C\|x_k - x^\star\|_2^2$. □

C.9 Unconstrained specialization

When

$$\mathcal{X} = \mathbb{R}^d,$$

the normal cone is

$$N_{\mathcal{X}}(x) = \{0\}$$

at every input. The constrained projection reduces to

$$\xi_M^\dagger(x) \in \operatorname*{arg\,min}_{\xi \in \mathcal{D}^\star(x)} \frac{1}{2} \left\| \mathcal{G}(\xi) + \nabla\Phi(x) \right\|_{M^{-1}}^2,$$

and

$$g_M^\dagger(x) = \mathcal{G}\big(\xi_M^\dagger(x)\big) + \nabla\Phi(x).$$

The certificate becomes

$$\mathcal{R}_M(x) = 0 \iff 0 \in \partial F(x) \iff x \in \arg\min F,$$

and the common-descent inequality becomes

$$g^\top d_M(x) \le -\mathcal{R}_M(x)^2 \ \ \forall\, g \in \partial F(x).$$

Thus the original unconstrained DCI is recovered exactly as a special case of the constrained framework.

D Additional Experimental Details and Results

This appendix gives the complete constructions, evaluation criteria, and numerical results for the experiments in Section 5. Experiments 1–4 use controlled objectives with analytically known geometry. Experiment 5 trains context conditioned SOC-ICNNs and evaluates inference on their frozen objectives, separating optimization error from approximation error of the learned objective. The convergence theorem concerns the exact algorithm. The experiments use numerical implementations with the activity tolerances, approximate proximal solves, and line-search variants specified below.

D.1 Common Model and Notation

Generic inference problem. The paper studies

$$x^\star(y) \in \arg\min_{x\in\mathcal{X}} F_y(x), \;\; F_y(x) = f_\theta(x;y) + \Phi(x;y), \tag{50}$$

where $x \in \mathbb{R}^{d_x}$ is the decision variable optimized at inference time, y is an observed context held fixed during that optimization, $\mathcal{X}$ is a closed convex feasible set, $f_\theta(\cdot;y)$ is the learned convex network objective, and Φ denotes any additional explicit convex term. Experiments 1–4 suppress y and construct F directly. Experiment 5 uses a daily context and denotes the battery decision by z.

Full SOC-ICNN objective. To avoid overloading z, this appendix denotes the hidden activation at layer ℓ by h_ℓ. For fixed context y,

$$\begin{aligned} a_1(x) &= W_1 x + b_1, & h_1(x) &= [a_1(x)]_+, \\ a_\ell(x) &= W_\ell x + U_\ell h_{\ell-1}(x) + b_\ell, & h_\ell(x) &= [a_\ell(x)]_+, \; \ell = 2,\dots,L. \end{aligned} \tag{51}$$

The ICNN sign restrictions are $U_\ell \geq 0$ elementwise and $c \geq 0$. The quadratic and norm residuals are

$$q_h(x) = B_h x + e_h, \;\; u_g(x) = A_g x + d_g,$$

and the scalar objective is

$$f(x) = c^\top h_L(x) + v^\top x + b_0 + \sum_{h=1}^{H} \frac{\alpha_h}{2}\|q_h(x)\|_2^2 + \sum_{g=1}^{G} \lambda_g \|u_g(x)\|_2, \tag{52}$$

with $\alpha_h > 0$ and $\lambda_g \geq 0$. The network therefore does *not* directly output a decision. It outputs a scalar convex objective as a function of the decision, and inference then minimizes that objective.

Table 3: **Model symbols used throughout the appendix.**

Symbol	Meaning
d_x	Decision dimension. This notation is used to distinguish dimension from the norm offset d_g and from day indices in Experiment 5.
L, m	ReLU depth and hidden width.
W_ℓ	Decision-to-layer matrix, $W_\ell \in \mathbb{R}^{m\times d_x}$.
U_ℓ	Nonnegative hidden-to-hidden matrix, $U_\ell \in \mathbb{R}_+^{m\times m}$.
a_ℓ, h_ℓ	ReLU preactivation and activation in equation 51.
c, v, b_0	Final hidden weight, direct linear decision coefficient, and scalar bias.
H	Number of quadratic modules.
B_h, e_h, α_h	Matrix, offset, and coefficient of $q_h = B_h x + e_h$.
G	Number of Euclidean-norm modules.
A_g, d_g, λ_g	Matrix, offset, and coefficient of $u_g = A_g x + d_g$.
k_g	Dimension of u_g, so $A_g \in \mathbb{R}^{k_g\times d_x}$.

Set-valued geometry and DCI symbols. A ReLU coordinate is structurally ambiguous when $a_{\ell,j}(x) = 0$; a norm block is ambiguous when $u_g(x) = 0$. At such points $\partial F(x)$ is set-valued. For constrained problems, the complete first-order object is $\partial F(x) + N_{\mathcal{X}}(x)$, where $N_{\mathcal{X}}(x)$ is the convex normal cone.

For $M \succ 0$, DCI computes

$$(s_M^\dagger, n_M^\dagger) \in \arg \min_{\substack{s \in \partial F(x) \\ n \in N_{\mathcal{X}}(x)}} \frac{1}{2}(s+n)^\top M^{-1}(s+n). \tag{53}$$

We use

$$g_M^\dagger = s_M^\dagger + n_M^\dagger, \;\; \mathcal{R}_M(x) = \|g_M^\dagger\|_{M^{-1}}, \;\; d_M = -M^{-1} g_M^\dagger.$$

DCI-I sets $M = I$. DCI-C uses the explicit smooth-branch curvature metric. All final certificates are recomputed using the Euclidean residual

$$\mathcal{R}_{\mathcal{X}}(x) = \operatorname{dist}(0, \partial F(x) + N_{\mathcal{X}}(x)) \,. \tag{54}$$

Table 4: **Experimental map.** The controlled experiments are not five different learned models; they isolate different pieces of the same inference geometry.

Exp.	Object	Main question
1	Frozen random deep SOC-ICNN $f(x)$.	Are the analytic gradient, Hessian, and boundary readout correct?
2	Strongly convex controlled ReLU–norm objective with known $x^\star = 0$.	How does the recovered subdifferential affect inference at structural boundaries?
3	Smooth controlled objectives with prescribed Hessian condition number.	Once first-order geometry is exact, what does explicit curvature add?
4	Controlled strongly convex objectives over free/box/ball/simplex/SOC sets.	Why are the normal cone, curvature metric, and safeguard separately needed?
5	Context-conditioned learned battery-cost surface.	Does DCI remain reliable and efficient on objectives produced by trained SOC-ICNNs?

Computational environment and timing. All experiments use Python 3.11, PyTorch 2.0, CUDA 12, and an NVIDIA GeForce RTX 4060 Ti. Controlled geometric calculations use float64. Reported times include the network evaluation and the optimization operations performed by each method; certificate recomputation is excluded unless explicitly stated. The pointwise oracle benchmark in Experiment 2 separately reports its warm-up and repetition protocol.

D.2 Experiment 1: Validation of the Exact SOC-ICNN Geometry

What this experiment does. This experiment uses the full deep model in equation 51. We freeze random parameters and evaluate the same scalar function $f : \mathbb{R}^{d_x} \to \mathbb{R}$ in several independent ways.

A forward pass exposes

$$\{a_\ell(x), h_\ell(x)\}_{\ell=1}^L, \;\; \{q_h(x)\}_{h=1}^H, \;\; \{u_g(x)\}_{g=1}^G.$$

A point is structurally nondegenerate when

$$\min_{\ell,j} |a_{\ell,j}(x)| > 10^{-9}, \;\; \min_g \|u_g(x)\|_2 > 10^{-9}.$$

At such points the subdifferential is a singleton and the analytic readout must equal the ordinary gradient. At the deliberately constructed mixed boundary, the ReLU multiplier is an interval and the zero-norm multiplier is a Euclidean ball, so the experiment checks the complete directional support function rather than one branch-selected derivative.

Random SOC-ICNN construction. For decision dimension d_x, depth L, and width m, standard-normal base matrices are scaled as

$$W_\ell = 0.65\widetilde{W}_\ell/\sqrt{d_x}, \;\; U_\ell = 0.55|\widetilde{U}_\ell|/\sqrt{m}.$$

Hidden biases are $0.35\widetilde{b}_\ell$, positive output weights are $0.6|\widetilde{c}|/\sqrt{m}$, the direct linear coefficient is $0.25\widetilde{v}$, and the output bias is $0.1\widetilde{b}_0$. Quadratic modules use

$$B_h = 0.75\widetilde{B}_h/\sqrt{d_x}, \; e_h = 0.35\widetilde{e}_h, \; \alpha_h \sim U[0.4, 1.2],$$

and norm modules use

$$A_g = 0.70\widetilde{A}_g/\sqrt{d_x}, \; d_g = 0.45\widetilde{d}_g, \; \lambda_g \sim U[0.4, 1.2].$$

The nonnegativity of U_ℓ and of the output weights preserves input convexity.

First-order, curvature, and Taylor tests. The gradient test uses 250 nondegenerate inputs for a depth-4, width-64 model with $d_x = 20$, two quadratic modules and two 20-dimensional norm modules. The curvature test uses 100 inputs for depth 3, width 32, $d_x = 10$, two quadratic and two 10-dimensional norm modules. The explicit smooth-branch Hessian is

$$H_{\exp} = \sum_h \alpha_h B_h^\top B_h + \sum_g \lambda_g A_g^\top \frac{I - \widehat{u}_g \widehat{u}_g^\top}{\|u_g\|_2} A_g, \;\; \widehat{u}_g = u_g / \|u_g\|_2.$$

For the Taylor test, 500 directions are sampled at each radius $r \in \{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$ and we measure

$$E_2(\delta) = |f(x+\delta) - f(x) - \nabla f(x)^\top \delta - \tfrac{1}{2}\delta^\top H_{\exp}(x)\delta|.$$

The selected anchor has minimum ReLU margin 1.800×10^{-2} and minimum norm residual 2.197; all 1500 perturbations remain on the same branch. Gradient errors use the Euclidean norm, Hessian errors use the Frobenius norm, and relative errors divide by the corresponding reference norm.

Mixed structural boundary. At $x_0 = 0 \in \mathbb{R}^2$ we use one zero ReLU and one zero norm residual:

$$f(x) = v^\top x + b_0 + 0.7[(0.4, 0.9)x + 0.8]_+ + 1.3[(1, -0.7)x]_+ + \frac{0.9}{2}\left\| \begin{bmatrix} 0.8 & -0.2 \\ 0.1 & 0.6 \end{bmatrix} x + \begin{bmatrix} 0.25 \\ -0.35 \end{bmatrix} \right\|_2^2 + 1.1 \left\| \begin{bmatrix} 1 & 0.25 \\ -0.15 & 0.85 \end{bmatrix} x \right\|_2,$$

with $v = (0.2, -0.15)^\top$. We sample 1000 directions, use one-sided finite differences with step 10^{-7}, sample 5000 admissible dual branches, and test 5000 global support inequalities. The canonical gap is the difference between the complete directional derivative and the derivative returned by the canonical branch; its positive value records directions for which the single branch omits an active support.

Table 5: **Validation of the exact SOC-ICNN geometry.** AD is an independent automatic-differentiation reference.

Component	Diagnostic	Result
First order	Mean / max gradient error	2.118×10^{-15} / 3.731×10^{-15}
First order	Mean relative gradient error	1.297×10^{-16}
First order	Mean readout time	0.866 ms
Curvature	Mean / max Hessian error	4.117×10^{-16} / 6.046×10^{-16}
Curvature	Mean relative Hessian error	1.305×10^{-16}
Curvature	Mean minimum eigenvalue	0.205
Taylor	Empirical log–log slope	3.006
Boundary	Mean / max finite-difference error	2.374×10^{-8} / 3.245×10^{-8}
Boundary	Max readout-maximization error	8.882×10^{-16}
Boundary	Canonical-gap fraction	1.000
Boundary	Minimum canonical gap	0.942
Boundary	Maximum support violation	0

Table 6: **Second-order Taylor residual on the fixed smooth branch.**

Radius	Mean	Median	Maximum	Same branch
1.000×10^{-4}	8.825×10^{-15}	7.105×10^{-15}	3.730×10^{-14}	1.000
3.000×10^{-4}	2.374×10^{-13}	1.865×10^{-13}	1.089×10^{-12}	1.000
1.000×10^{-3}	8.946×10^{-12}	7.569×10^{-12}	3.128×10^{-11}	1.000

These checks validate the two distinct objects used by DCI: ordinary gradient and curvature on smooth branches, and the complete support-function geometry at structural boundaries.

D.3 Experiment 2: Complete First-Order Geometry at Controlled Structural Boundaries

Experiment 2 uses a controlled special case of the ReLU–quadratic–norm family in equation 52, chosen so that the unique minimizer lies on prescribed structural boundaries. Every run solves

$\min_{x\in\mathbb{R}^{d_x}} F(x)$, with $x^\star = 0$ and its subdifferential known by construction. This lets us compare a single canonical slope with the complete set of supporting slopes without surrogate-model error.

Symbols specific to Experiment 2. The objective contains m_{R} scalar ReLU terms and m_{N} norm modules. The matrix $W \in \mathbb{R}^{m_{\mathrm{R}}\times d_x}$ has row $w_i^\top$; b_i is the ReLU bias and $\alpha_i > 0$ its output coefficient. Norm module g has matrix $A_g \in \mathbb{R}^{k_g\times d_x}$, offset $d_g \in \mathbb{R}^{k_g}$, and weight $\lambda_g > 0$. The linear coefficient is $\ell \in \mathbb{R}^{d_x}$.

At a point x, define

$$\mathcal{Z}_{\mathrm{R}}(x) = \{i : w_i^\top x + b_i = 0\}, \qquad \mathcal{Z}_{\mathrm{N}}(x) = \{g : A_g x + d_g = 0\}.$$

Only these blocks contribute free local multipliers. The number of scalar variables in the boundary-local projection is

$$p(x) = |\mathcal{Z}_{\mathrm{R}}(x)| + \sum_{g\in\mathcal{Z}_{\mathrm{N}}(x)} k_g, \tag{55}$$

whereas the full lifted support representation uses $p_{\mathrm{lift}} = m_{\mathrm{R}} + \sum_{g=1}^{m_{\mathrm{N}}} k_g$ scalar multipliers. The scaling cases below report $p = p(0)$, p_{lift}, and d_x.

At the known optimum, the complete subdifferential is

$$\begin{aligned}\partial F(0) = \ell &+ \sum_{i:b_i>0} \alpha_i w_i + \sum_{i\in\mathcal{Z}_{\mathrm{R}}(0)} [0,\alpha_i] w_i \\ &+ \sum_{g:d_g\neq 0} \lambda_g A_g^\top \frac{d_g}{\|d_g\|_2} + \sum_{g\in\mathcal{Z}_{\mathrm{N}}(0)} A_g^\top \{r\in\mathbb{R}^{k_g} : \|r\|_2 \le \lambda_g\}.\end{aligned} \tag{56}$$

The affine term ℓ is chosen so that $0 \in \partial F(0)$, while the canonical subgradient remains nonzero. Strong convexity makes $x^\star = 0$ the unique minimizer.

Controlled objective family. We use the strongly convex objective

$$F(x) = \tfrac{1}{2} x^\top Q x + \ell^\top x + \sum_{i=1}^{m_{\mathrm{R}}} \alpha_i [w_i^\top x + b_i]_+ + \sum_{g=1}^{m_{\mathrm{N}}} \lambda_g \|A_g x + d_g\|_2, \tag{57}$$

where $[s]_+ = \max\{0, s\}$. The quadratic matrix is $Q = 0.8I + 0.35 R_Q^\top R_Q / d_x$, where $R_Q \in \mathbb{R}^{d_x\times d_x}$ has independent standard-normal entries and I is the identity matrix. ReLU rows are normalized Gaussian directions, with $\alpha_i \sim U[0.5, 1.5]$, where $U[a, b]$ denotes the uniform distribution on $[a, b]$. Boundary ReLU biases are zero; the remaining biases have random signs and magnitudes in $[0.8, 1.2]$. Norm matrices have Gaussian entries scaled by $1/\sqrt{d_x}$, with $\lambda_g \sim U[0.5, 1.5]$, zero offsets for boundary modules, and random nonzero offsets with norms in $[0.8, 1.2]$.

For each zero ReLU preactivation, we select a multiplier in $[0.2\alpha_i, 0.8\alpha_i]$; for each zero norm residual, we select a multiplier with norm in $[0.2\lambda_g, 0.8\lambda_g]$. The affine term ℓ cancels the resulting sum of supporting slopes. Instances are retained only when the canonical subgradient at the optimum has Euclidean norm at least 0.1.

Small correctness suite and methods. We test $d_x \in \{10, 30\}$, ReLU/norm/mixed boundaries, and boundary-block counts of two and four, with two generated instances per setting. This gives 24 objectives, each evaluated from two Gaussian initializations of scale 1.5, for 48 matched runs per method. Each objective additionally contains six smooth ReLU terms and two smooth norm modules; all norm residuals have dimension two.

Single-Branch uses the canonical subgradient with the same curvature metric as DCI-C, without a proximal safeguard. Gradient Sampling uses at most 40 sampled gradients, a sampling radius decreasing from 0.05 to 10^{-7}, and 250 iterations. L-BFGS uses memory 20 and 400 iterations. Lifted-PD applies a primal–dual solver to the full lifted support representation, with tolerance 10^{-10} and at most 10,000 iterations.

DCI-I uses the identity metric, while DCI-C uses the explicit curvature with damping 10^{-6} and eigenvalues clipped to $[10^{-3}, 10^{3}]$. Both use a projection tolerance of 10^{-11}, at most 5000 projection

iterations, and 200 outer iterations. The computed metric residual $\widehat{\mathcal{R}}_k$ has stopping tolerance 10^{-7}. The implementation tests $t = 2^{-j}$, $j = 0, \ldots, 8$, and accepts the first step satisfying $F(x_k + td_k) \leq F(x_k) - 0.1t\widehat{\mathcal{R}}_k^2$. If backtracking fails, the primal–dual routine approximately solves the metric proximal subproblem with $\gamma = 100$ within 10,000 iterations, and its final primal iterate is used for the update.

Evaluation criteria. The normalized objective gap is $\mathrm{Gap}(x) = \max\{F(x) - F(x^\star), 0\}/(1 + |F(x^\star)|)$. Final residuals are recomputed in the Euclidean norm using the same numerical projection for all methods. DCI and the final evaluation treat ReLU preactivations with absolute value at most 10^{-7} and norm residuals with Euclidean norm at most 10^{-7} as structural boundaries, retaining their full multiplier intervals and balls, respectively. Success requires normalized gap $\leq 10^{-8}$, distance to $x^\star \leq 2 \times 10^{-4}$, and numerical residual $\leq 2 \times 10^{-5}$, irrespective of the solver's stopping status. Wrong denotes the percentage of all runs that declare convergence but fail at least one success criterion. Reported solver runtimes exclude final residual recomputation.

Table 7: **Aggregate controlled-boundary results.** Forty-eight runs per method. Gap, distance, and numerical residual are medians over all runs; iteration counts and runtimes are means. A dash indicates that convergence declarations were not identified by the solver interface.

Method	Success	Wrong	Gap	Dist.	Residual	Iter.	Time (s)
Single-Branch	0.0%	0.0%	3.264×10^{-3}	1.303×10^{-1}	4.143×10^{-1}	44.2	0.714
L-BFGS	0.0%	–	2.068×10^{-5}	1.492×10^{-2}	7.239×10^{-2}	76.8	0.710
Gradient Sampling	68.8%	12.5%	7.029×10^{-9}	1.358×10^{-7}	1.592×10^{-7}	93.1	25.067
DCI-I	100.0%	0.0%	3.623×10^{-12}	9.465×10^{-9}	1.239×10^{-8}	23.4	0.925
DCI-C	100.0%	0.0%	4.416×10^{-11}	3.879×10^{-9}	4.996×10^{-9}	13.3	0.672
Lifted-PD	100.0%	0.0%	5.570×10^{-12}	4.165×10^{-11}	2.391×10^{-11}	303.3	0.094

Table 8: **Success rate by boundary type.** Each cell uses four matched runs. Active counts the boundary blocks; Single and GS denote Single-Branch and Gradient Sampling.

d_x	Type	Active	Single	GS	DCI-I	DCI-C
10	mixed	2	0%	75%	100%	100%
10	mixed	4	0%	25%	100%	100%
10	norm	2	0%	50%	100%	100%
10	norm	4	0%	0%	100%	100%
10	relu	2	0%	100%	100%	100%
10	relu	4	0%	50%	100%	100%
30	mixed	2	0%	100%	100%	100%
30	mixed	4	0%	75%	100%	100%
30	norm	2	0%	50%	100%	100%
30	norm	4	0%	100%	100%	100%
30	relu	2	0%	100%	100%	100%
30	relu	4	0%	100%	100%	100%

Scaling design. We use three sweeps with five independent instances per case and one initialization per instance. The total-size sweep fixes d_x and p while varying p_{lift}; the active-size sweep fixes d_x and p_{lift} while varying p. The dimension sweep fixes p while jointly increasing the input dimension and the number of modules. Table 9 lists the cases, and Table 10 reports the aggregate results over 60 runs per method using the same evaluation criteria.

Pointwise boundary-oracle microbenchmark. The oracle is evaluated at the known minimizer $x^\star = 0$ with zero activity tolerance. Each instance receives five warm-up calls followed by 50 timed calls. Mean and SD report the mean and standard deviation across the five per-instance mean oracle times; Projection reports the corresponding mean time spent on the local projection. These are pointwise oracle timings, not complete inference runtimes.

At fixed input and boundary dimensions, the local projection times vary little as the full representation grows. These measurements distinguish the cost of the reduced multiplier problem from the

Table 9: **Large-scale case design.** Active is the number of boundary blocks, k_g is the norm residual dimension, $p = p(0)$, and Lifted denotes p_{lift}.

Sweep	Case	d_x	Active	ReLU	Norm	k_g	p	Lifted
total size	N-small	100	4	128	8	4	10	160
total size	N-medium	100	4	512	32	4	10	640
total size	N-large	100	4	2048	128	4	10	2560
active size	q-2	100	2	512	64	4	5	768
active size	q-4	100	4	512	64	4	10	768
active size	q-8	100	8	512	64	4	20	768
active size	q-16	100	16	512	64	4	40	768
active size	q-32	100	32	512	64	4	80	768
dimension	d-25	25	4	100	8	4	10	132
dimension	d-50	50	4	200	12	4	10	248
dimension	d-100	100	4	400	25	4	10	500
dimension	d-200	200	4	800	50	4	10	1000

Table 10: **Large-suite aggregate optimization results.** The numerical residual is the median over all runs; iteration counts and runtimes are means. Success uses the same criteria as the small-scale suite.

Method	Runs	Success	Residual	Iter.	Time (s)
DCI-C	60	100.0%	4.570×10^{-9}	15.7	29.213
DCI-I	60	100.0%	1.598×10^{-8}	56.7	69.449
L-BFGS	60	0.0%	2.155×10^{-1}	223.1	6.475
Lifted-PD	60	100.0%	2.398×10^{-10}	1065.3	11.118

remaining oracle computations; the dimension sweep additionally captures the joint growth of input dimension and model size.

D.4 Experiment 3: Explicit Curvature under Ill Conditioning

What is isolated. Every trajectory in this experiment remains strictly away from ReLU and zero-norm boundaries. Therefore

$$\partial F(x) = \{\nabla F(x)\},$$

and the network-side first-order oracle is single-valued along the entire trajectory. In particular, AD-GD and Readout-GD use the same first-order information, so any improvement of DCI-C over these methods is attributable to curvature rather than set-valued geometry.

Controlled smooth objectives. The decision dimension is $d_x = 20$. For a random orthogonal matrix R, let $z = Rx$. We construct

$$F_{\mathrm{quad}}(x) = \tfrac{1}{2} z^\top Q z + a^\top z + \sum_{j=1}^{8} \rho_j [w_j^\top z + b_j]_+, \tag{58}$$

$$F_{\mathrm{qnorm}}(x) = F_{\mathrm{quad}}(x) + \sum_{g=1}^{3} \lambda_g \|D_g z + e_g\|_2. \tag{59}$$

The eight ReLU directions have norm 0.65, $\rho_j \in [0.25, 0.75]$, and biases alternating between $+5$ and -5. In the quadratic–norm family, D_g is diagonal with entries in $[0.12, 0.28]$, $\lambda_g \in [0.25, 0.50]$, and e_g has one nonzero coordinate of magnitude four. The target full-Hessian condition numbers are

$$\kappa \in \{1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10^4\}.$$

For each κ, the quadratic part uses logarithmically spaced eigenvalues from 1 to κ, followed by a random permutation. In the quadratic–norm family, the analytic norm curvature at the origin is subtracted from the quadratic diagonal so that the *full* Hessian at $x^\star = 0$ has the prescribed condition number. The affine term cancels all smooth-module slopes at the origin. Starts have radius one with $z_{0,i} = s_i/\sqrt{d_x}$ and $s_i \in \{-1, 1\}$. Five seeds for each family and condition number yield 130 matched objectives.

Table 11: **DCI-C boundary-oracle timing.** Times are milliseconds.

Sweep	Case	d_x	p	Lifted	Mean	SD	Projection
active size	q-2	100	5	768	624.1	195.2	608.3
active size	q-4	100	10	768	675.1	23.5	659.8
active size	q-8	100	20	768	727.1	68.3	709.7
active size	q-16	100	40	768	738.5	14.5	724.6
active size	q-32	100	80	768	1054.4	90.4	1042.3
dimension	d-25	25	10	132	60.3	7.5	59.3
dimension	d-50	50	10	248	132.5	76.8	130.4
dimension	d-100	100	10	500	710.5	59.3	700.8
dimension	d-200	200	10	1000	1893.4	34.1	1852.5
total size	N-small	100	10	160	687.1	16.7	680.5
total size	N-medium	100	10	640	686.3	43.2	676.5
total size	N-large	100	10	2560	740.5	52.5	697.3

Table 12: **DCI-I boundary-oracle timing.** Times are milliseconds.

Sweep	Case	d_x	p	Lifted	Mean	SD	Projection
active size	q-2	100	5	768	550.3	197.6	547.2
active size	q-4	100	10	768	672.7	44.8	669.2
active size	q-8	100	20	768	745.7	80.1	740.7
active size	q-16	100	40	768	727.5	19.2	724.2
active size	q-32	100	80	768	961.3	110.8	958.3
dimension	d-25	25	10	132	56.2	5.0	55.7
dimension	d-50	50	10	248	125.6	76.3	124.7
dimension	d-100	100	10	500	717.2	15.2	715.4
dimension	d-200	200	10	1000	1878.7	69.1	1873.4
total size	N-small	100	10	160	675.3	27.3	674.4
total size	N-medium	100	10	640	678.4	23.0	676.2
total size	N-large	100	10	2560	676.5	95.9	660.7

Methods and success criteria. AD-GD and Readout-GD use identical Armijo search. DCI-C-Local uses the analytic readout and explicit Hessian without the nonsmooth safeguard. Canonical-Newton uses AD gradients and the full AD Hessian. AD-L-BFGS uses memory 12. Iteration budgets are 20,000 for gradient descent, 80 for Newton methods, and 500 for L-BFGS. Armijo uses parameter 10^{-4}, contraction 0.5, and at most 45 reductions. Metric clipping is $[10^{-10}, 10^{10}]$. Success requires normalized gap $\leq 10^{-10}$, residual $\leq 2.5 \times 10^{-7}$, and decision error $\leq 2 \times 10^{-5}$. For compactness, the tables abbreviate DCI-C-Local as DCI-C.

Table 13: **Smooth-branch isolation checks.**

Diagnostic	Result
Mean analytic–AD gradient error	2.854×10^{-14}
Mean explicit–AD Hessian error	2.734×10^{-13}
Minimum branch-stability rate	1.000
Minimum ReLU margin	4.586
Minimum norm residual	3.921

The branch-stability diagnostics and the near-identical AD-GD/Readout-GD iteration profiles verify that this experiment isolates curvature rather than set-valued geometry. DCI-C and Canonical-Newton have identical iteration counts throughout the sweep, while the structured Hessian is cheaper in the tested implementation, yielding median runtime reductions of $7.2\times$ on the quadratic family and $10.4\times$ on the quadratic–norm family at $\kappa = 10^4$.

Table 14: **Conditioning study.** Each entry is success (%)/median iterations over five seeds. Representative condition numbers are shown; Figure 1 plots the full 13-point sweep.

Family	Method	1	10	10^2	10^3	10^4
Quadratic	AD-GD	100/1	100/65	100/645	0/5947	0/20000
Quadratic	Readout-GD	100/1	100/65	100/646	0/5856	0/20000
Quadratic	AD-L-BFGS	100/1	100/24	100/73	100/199	0/500
Quadratic	Canonical-Newton	100/1	100/1	100/1	100/1	100/1
Quadratic	DCI-C	100/1	100/1	100/1	100/1	100/1
Quadratic–norm	AD-GD	100/2	100/64	80/650	0/5859	0/20000
Quadratic–norm	Readout-GD	100/2	100/64	80/650	0/5859	0/20000
Quadratic–norm	AD-L-BFGS	100/2	100/24	100/70	60/208	0/500
Quadratic–norm	Canonical-Newton	100/2	100/2	100/2	100/2	100/2
Quadratic–norm	DCI-C	100/2	100/2	100/2	100/2	100/2

Table 15: **Largest condition number $\kappa = 10^4$.** Values are medians over five seeds.

Family	Method	Success	Iter.	Gap	Residual	Time (s)
Quadratic	AD-GD	0%	20000	4.218×10^{-7}	7.689×10^{-3}	12.9449
Quadratic	Readout-GD	0%	20000	4.218×10^{-7}	7.689×10^{-3}	9.1467
Quadratic	AD-L-BFGS	0%	500	3.613×10^{-14}	4.562×10^{-6}	0.7932
Quadratic	Canonical-Newton	100%	1	0	2.943×10^{-12}	0.0076
Quadratic	DCI-C	100%	1	0	3.243×10^{-12}	0.0011
Quadratic–norm	AD-GD	0%	20000	2.921×10^{-7}	7.689×10^{-3}	31.5713
Quadratic–norm	Readout-GD	0%	20000	2.921×10^{-7}	7.689×10^{-3}	22.9283
Quadratic–norm	AD-L-BFGS	0%	500	1.343×10^{-14}	4.951×10^{-6}	1.5096
Quadratic–norm	Canonical-Newton	100%	2	0	1.429×10^{-13}	0.0430
Quadratic–norm	DCI-C	100%	2	0	1.429×10^{-13}	0.0041

D.5 Experiment 4: Generalized Feasible-Set Geometry and Safeguarding

What is being optimized. We solve $\min_{x \in \mathcal{X}} F(x)$ with a known feasible minimizer $x^\star$. Unlike Experiment 2, constrained optimality can require a nonzero normal vector. The experiment examines the joint use of the objective subdifferential and feasible-set geometry, then compares local inference with and without a proximal safeguard.

Symbols specific to Experiment 4. Write $\Delta x = x - x^\star$. The matrix $Q \succ 0$ supplies quadratic curvature, and m_{R} and m_{N} denote the numbers of ReLU and norm modules. ReLU module i has direction $w_i \in \mathbb{R}^{d_x}$, offset b_i, and coefficient $\alpha_i > 0$; norm module g has matrix $A_g \in \mathbb{R}^{3 \times d_x}$, offset $d_g \in \mathbb{R}^3$, and weight $\lambda_g > 0$. At the known optimum, select $\theta_i^\star \in [0.25, 0.75]$ when $b_i = 0$, and set $\theta_i^\star = 1$ for $b_i > 0$ and 0 for $b_i < 0$. For $d_g = 0$, choose $r_g^\star$ with norm in $[0.25\lambda_g, 0.65\lambda_g]$; otherwise set $r_g^\star = \lambda_g d_g / \|d_g\|_2$. The selected module support is

$$g_{\mathrm{net}}^\star = \sum_{i=1}^{m_{\mathrm{R}}} \alpha_i \theta_i^\star w_i + \sum_{g=1}^{m_{\mathrm{N}}} A_g^\top r_g^\star.$$

Choose $n^\star \in N_{\mathcal{X}}(x^\star)$, with $n^\star = 0$ in the unconstrained case, and set $a = -(g_{\mathrm{net}}^\star + n^\star)$. This gives $0 \in \partial F(x^\star) + N_{\mathcal{X}}(x^\star)$; strong convexity makes $x^\star$ the unique minimizer.

Controlled objective class. All objectives have $d_x = 10$ and take the form

$$\begin{aligned} F(x) = \tfrac{1}{2}\Delta x^\top Q \Delta x + a^\top \Delta x + \sum_{i=1}^{m_{\mathrm{R}}} \alpha_i [w_i^\top \Delta x + b_i]_+ \\ + \sum_{g=1}^{m_{\mathrm{N}}} \lambda_g \|A_g \Delta x + d_g\|_2, \qquad x \in \mathcal{X}, \end{aligned} \tag{60}$$

where $[t]_+ = \max\{0, t\}$. The matrix Q is a random orthogonal rotation of a geometric spectrum from 1 to the condition number specified in Table 16. ReLU directions are normalized Gaussian

Table 16: **Experiment 4 difficulty regimes.** Zero modules lie at structural boundaries at the known optimum.

Regime	ReLUs	Zero ReLUs	Norms	Zero norms	$\kappa(Q)$
Regular	4	0	1	0	10
Constraint boundary	4	0	1	0	30
Structural boundary	6	2	2	1	40
Joint boundary	6	2	2	1	50
Stress	10	6	3	2	100

vectors, $\alpha_i \in [0.40, 0.95]$, and nonzero ReLU offsets alternate between ± 2.5. Norm matrices have the form $A_g = 0.35\widetilde{A}_g$, where $\widetilde{A}_g$ has independent standard-normal entries; $\lambda_g \in [0.40, 0.85]$ and nonzero norm offsets have norm 3.5. Modules designated as structural boundaries have zero offsets.

Feasible sets and regimes. We use unconstrained space, the box $[-1, 1]^{d_x}$, the unit Euclidean ball, the probability simplex, and the second-order cone $\{(t, u) : t \geq \|u\|_2\}$. Box constraint-boundary, joint-boundary, and stress cases activate one, two, and five faces, respectively. Ball regular optima have radius 0.35; the other ball optima have radius one and normal magnitudes 0.8, 1.2, and 1.8, respectively. Simplex support sizes are nine, five, and two for the corresponding boundary regimes, with the equality normal always included. For a unit vector u, second-order cone regular optima are $(0.9, 0.3u)$, ordinary boundary optima are $(1, u)$, and stress optima are $(0.25, 0.25u)$, with opposed initial directions in the stress cases. Unconstrained problems use the regular, structural-boundary, and stress regimes; each constrained set uses the regular, constraint-boundary, joint-boundary, and stress regimes. Five seeds per setting give 95 objectives and 570 main solver runs, with a common initialization across methods for each objective.

Joint constrained readout. For $M \succ 0$, the joint projection solves

$$(s^\dagger_M, n^\dagger_M) \in \underset{s \in \partial F(x),\, n \in N_{\mathcal{X}}(x)}{\arg\min} \tfrac{1}{2}(s+n)^\top M^{-1}(s+n).$$

Its reduced variables consist of intervals for boundary ReLUs, Euclidean balls for zero norm residuals, free equality-normal coefficients, and nonnegative inequality-normal coefficients. Accelerated projected gradient uses at most 8000 iterations and tolerance 10^{-11}. The network-only residual omits the normal cone and targets $\operatorname{dist}(0, \partial F(x))$, whereas the joint residual targets $\operatorname{dist}(0, \partial F(x) + N_{\mathcal{X}}(x))$. Both are evaluated numerically with the activity tolerances below.

Compared methods. AD-PGD combines a branch-selected AD vector with Euclidean feasible projection. Readout-PGD instead uses the minimum-norm objective subgradient without jointly optimizing over the normal cone. DCI-I-Local uses the joint projection with $M = I$. For $u_g(x) = A_g(x - x^\star) + d_g$ and $\widehat{u}_g(x) = u_g(x)/\|u_g(x)\|_2$, DCI-C-Local uses

$$M_C(x) = Q + \sum_{g : \|u_g(x)\|_2 > 10^{-7}} \lambda_g A_g^\top \frac{I - \widehat{u}_g(x)\widehat{u}_g(x)^\top}{\|u_g(x)\|_2} A_g + 10^{-7} I,$$

with eigenvalues clipped to $[10^{-5}, 10^6]$. DCI-C-Full uses the same local path and invokes a metric proximal step after finite line-search failure. Prox-Only uses the same curvature metric but takes a proximal step at every outer iteration. AD-PGD and Readout-PGD provide projected-gradient references; the DCI variants compare identity and curvature metrics, followed by the addition of the proximal safeguard.

Local feasible search and safeguard. For the computed joint vector $g^\dagger_k = s^\dagger_{M_k} + n^\dagger_{M_k}$, set $d_k = -M_k^{-1} g^\dagger_k$. Local DCI tests $z_k(t) = \Pi_{\mathcal{X}}(x_k + t d_k)$, where $\Pi_{\mathcal{X}}$ is Euclidean projection, and writes $s_k(t) = z_k(t) - x_k$. The default schedule is $t = 2^{-j}$, $j = 0, \ldots, 10$. The numerical implementation accepts a trial when

$$F'(x_k; s_k(t)) < 0, \qquad F(z_k(t)) \leq F(x_k) + 10^{-4} F'(x_k; s_k(t)).$$

Table 17: **Certificate validation at the constructed optima.** Network residuals are means, joint residuals are maxima, and projection iterations and times are medians over five instances. Times are in milliseconds.

Constraint	Regime	Network res.	Max joint res.	Proj. iter.	Proj. ms
ball	constraint boundary	0.800	2.293×10^{-16}	2	0.49
ball	joint boundary	0.892	1.630×10^{-8}	191	11.90
ball	regular	0.000	1.520×10^{-16}	0	0.32
ball	stress	1.043	5.634×10^{-9}	286	19.16
box	constraint boundary	0.800	1.327×10^{-16}	2	0.53
box	joint boundary	1.514	1.093×10^{-9}	158	7.14
box	regular	0.000	1.173×10^{-16}	0	0.32
box	stress	1.970	1.823×10^{-9}	157	10.68
simplex	constraint boundary	1.079	4.968×10^{-10}	105	3.90
simplex	joint boundary	1.530	1.064×10^{-8}	633	36.57
simplex	regular	1.107	2.001×10^{-16}	2	0.54
simplex	stress	1.166	1.519×10^{-9}	229	15.17
SOC	constraint boundary	1.131	3.675×10^{-16}	2	0.50
SOC	joint boundary	1.399	3.104×10^{-10}	130	6.04
SOC	regular	0.000	2.230×10^{-16}	0	0.36
SOC	stress	1.716	1.089×10^{-8}	353	25.02
unconstrained	regular	0.000	3.527×10^{-16}	0	0.31
unconstrained	stress	0.000	3.329×10^{-8}	442	36.73
unconstrained	structural boundary	0.000	1.038×10^{-9}	117	4.94

The projected-gradient references use up to 45 trial steps with contraction 0.5. If local search fails, Full approximately solves

$$\min_{z \in \mathcal{X}} F(z) + \frac{1}{2\gamma} \|z - x_k\|_{M_k}^2, \qquad \gamma = 8,$$

using at most 5000 primal–dual iterations and tolerance 5×10^{-7}; the final primal iterate is used for the update. The local methods allow 300 outer iterations and Prox-Only allows 40. Accepted local and proximal candidates are adjusted by the same numerical boundary-identification procedure, using a snap tolerance of 2×10^{-4}.

Numerical evaluation and termination. The algorithmic structural activity tolerance is 10^{-7}; final residuals and directional derivatives use 10^{-11}. ReLU preactivations with absolute value at most the applicable tolerance and norm residuals with norm at most it retain their full multiplier intervals and balls. Constraint activity uses tolerance 10^{-9}. Final residuals are recomputed with $M = I$ for all methods. DCI and Prox-Only stop using this Euclidean joint residual, while AD-PGD and Readout-PGD use their projected-gradient mapping; the stopping threshold is 2×10^{-5}. For the returned point x_{out}, define the normalized gap $\mathrm{Gap}(x_{\mathrm{out}}) = \max\{F(x_{\mathrm{out}}) - F(x^\star), 0\}/(1 + |F(x^\star)|)$. Success requires feasibility within 10^{-9}, normalized gap at most 10^{-8}, decision error $\|x_{\mathrm{out}} - x^\star\|_2 \le 2 \times 10^{-4}$, and numerical joint residual at most 2×10^{-5}, independently of the internal stopping flag. Declared records an internal convergence declaration; Certified requires a declaration and a passing final residual test, and Wrong records a declaration that fails this residual test. All rates use the total number of runs as the denominator. Solver runtimes exclude the final external residual recomputation.

Matched safeguard-budget ablation. We test every $J_{\max} \in \{0, 1, \ldots, 10\}$. For each objective, Local and Full share the initialization and metric construction, which are held fixed across budgets. Three seeds give 12 joint-boundary, 15 stress, and three structural-boundary objectives per budget, for 330 Local/Full pairs and 660 solver runs in total. Rescue is the proportion of pairs for which Local fails and Full succeeds. In this sweep it equals the Full safeguard invocation rate, so Table 26 and Figure 2 reports the common value as SG.

At $J_{\max} = 10$, Local succeeds on all three structural-boundary objectives, 11 of 12 joint-boundary objectives, and 13 of 15 stress objectives. Across the tested budgets, larger budgets reduce safeguard use, while Full succeeds throughout.

Table 18: **Overall reliability on 95 objectives per method.** SG is the percentage of runs invoking at least one safeguard step. Prox-Only denotes Metric-Prox-Only.

Method	Success	Declared	Certified	Wrong stop	SG
AD-PGD	49.5%	47.4%	36.8%	10.5%	0.0%
DCI-C-Full	100.0%	100.0%	100.0%	0.0%	5.3%
DCI-C-Local	94.7%	94.7%	94.7%	0.0%	0.0%
DCI-I-Local	63.2%	63.2%	63.2%	0.0%	0.0%
Prox-Only	100.0%	100.0%	100.0%	0.0%	100.0%
Readout-PGD	60.0%	57.9%	47.4%	10.5%	0.0%

Table 19: **Overall accuracy and computation.** Iterations and gaps are medians over all 95 runs; residuals and runtimes are summarized by their medians and 95th percentiles, denoted by $Q_{.95}$. Runtimes are in seconds.

Method	Iter.	Gap	Median res.	$Q_{.95}$ res.	Median s	$Q_{.95}$ s
AD-PGD	38	1.604×10^{-11}	2.568×10^{-5}	5.127×10^{-1}	0.0869	0.5065
DCI-C-Full	11	1.448×10^{-15}	3.594×10^{-9}	1.860×10^{-5}	0.1356	1.3989
DCI-C-Local	11	3.830×10^{-14}	4.052×10^{-9}	9.636×10^{-2}	0.0934	0.9610
DCI-I-Local	40	1.208×10^{-11}	1.853×10^{-5}	1.509×10^{0}	0.1681	1.3205
Prox-Only	5	2.520×10^{-13}	4.785×10^{-6}	1.653×10^{-5}	0.6554	4.9452
Readout-PGD	40	1.101×10^{-11}	1.823×10^{-5}	3.924×10^{-1}	0.0715	1.1882

D.6 Experiment 5: Learned and Explicit Battery-Scheduling Objectives

Application and data. We study day-ahead battery scheduling using context-conditioned convex neural objectives. For each day r, the pre-dispatch context c_r defines a convex objective over the battery coefficient $z \in [-1,1]^{12}$:

$$c_r \longmapsto \widehat{F}_\theta(\,\cdot\,; c_r), \quad z_r^\theta \in \operatorname*{arg\,min}_{z \in [-1,1]^{12}} \widehat{F}_\theta(z; c_r).$$

The model therefore predicts an objective rather than a schedule directly. We evaluate this pipeline at two distinct levels. Experiment 5A freezes a SOC-ICNN objective and asks how accurately and efficiently different algorithms minimize that same function; it therefore isolates inference error. Experiment 5B evaluates the resulting schedules under the realized engineering cost and compares learned ICNN objectives with explicit engineering objectives. Its operational regret includes surrogate error as well as any residual inference error. Reporting the complete certificate separately allows these two sources to be distinguished.

We use the versioned 2020-10-06 Open Power System Data hourly series for the Swedish SE3 bidding zone: day-ahead price, actual load, official day-ahead load forecast, and actual onshore wind generation. Timestamps are converted to Europe/Stockholm, and incomplete, duplicate, 23-hour, and 25-hour local days are removed. A causal wind forecast uses the same-hour mean of the previous seven available days. The chronological split contains 1068 training days (2015–2017), 363 validation days (2018), and 363 test days (2019).

The context $c_r \in \mathbb{R}^{77}$ contains 24 prices, 24 normalized load forecasts, 24 causal wind forecasts, and five calendar features. Net load is normalized by the training-set 95th percentile of its absolute value. Realized test-day load and wind are not included in the context.

Battery model and training objective. For $T = 24$, the 12-dimensional decision uses the basis

$$\overline{\Phi}_{t,j} = \sin(\pi j t/T), \quad j = 1, \ldots, 12.$$

The normalized battery has capacity one, initial energy 0.5, energy bounds $[0.1, 0.9]$, and four-hour duration, so $P_{\max} = 0.25$. We set

$$\eta = 0.98 \min\left\{ \frac{\min(e_0 - e_{\min}, e_{\max} - e_0)}{\max_t \|\overline{\phi}_t\|_1}, \frac{P_{\max}}{\max_t \|\overline{\phi}_{t+1} - \overline{\phi}_t\|_1} \right\}, \quad \Phi = \eta \overline{\Phi}.$$

Thus every $z \in [-1,1]^{12}$ satisfies the energy, power, and terminal-energy constraints without post-processing. Let $A \in \mathbb{R}^{24 \times 12}$ be the linear map from z to the hourly battery injection induced by this

Table 20: **Success/certified termination by difficulty regime.** Each entry is Success/Certified (%) over n objectives per method. AD and Readout denote the corresponding PGD methods; L and F denote Local and Full.

Regime	n	AD	Readout	DCI-I	DCI-C-L	DCI-C-F	Prox
Regular	25	100/100	100/100	100/100	100/100	100/100	100/100
Constraint boundary	20	50/50	50/50	100/100	100/100	100/100	100/100
Structural boundary	5	0/0	100/100	20/20	80/80	100/100	100/100
Joint boundary	20	10/0	5/0	45/45	90/90	100/100	100/100
Stress	25	40/0	64/20	20/20	92/92	100/100	100/100

Table 21: **Unconstrained results by regime.** Success and Certified are percentages; residuals and runtimes are medians over all five runs per row.

Regime	Method	Success	Certified	Residual	Time (s)
regular	AD-PGD	100.0%	100.0%	1.310×10^{-5}	0.0279
regular	DCI-C-Full	100.0%	100.0%	6.619×10^{-7}	0.0028
regular	DCI-C-Local	100.0%	100.0%	6.619×10^{-7}	0.0027
regular	DCI-I-Local	100.0%	100.0%	1.310×10^{-5}	0.0424
regular	Prox-Only	100.0%	100.0%	6.386×10^{-6}	0.0477
regular	Readout-PGD	100.0%	100.0%	1.310×10^{-5}	0.0185
stress	AD-PGD	0.0%	0.0%	3.614×10^{-1}	0.1821
stress	DCI-C-Full	100.0%	100.0%	9.947×10^{-10}	0.5595
stress	DCI-C-Local	100.0%	100.0%	9.947×10^{-10}	0.5123
stress	DCI-I-Local	20.0%	20.0%	1.044×10^{0}	0.2470
stress	Prox-Only	100.0%	100.0%	9.947×10^{-10}	2.0616
stress	Readout-PGD	100.0%	100.0%	9.947×10^{-10}	0.3400
structural boundary	AD-PGD	0.0%	0.0%	2.950×10^{-1}	0.0400
structural boundary	DCI-C-Full	100.0%	100.0%	3.608×10^{-9}	0.1356
structural boundary	DCI-C-Local	80.0%	80.0%	3.608×10^{-9}	0.0934
structural boundary	DCI-I-Local	20.0%	20.0%	5.574×10^{-1}	0.1308
structural boundary	Prox-Only	100.0%	100.0%	1.537×10^{-5}	0.4855
structural boundary	Readout-PGD	100.0%	100.0%	1.594×10^{-5}	0.2204

scaled basis, $g(z) = n + Az$, and R the hourly first-difference operator. The realized training target is

$$C(z; n, \lambda) = (\lambda - \tfrac{\Delta}{2}\mathbf{1})^\top g(z) + \Delta \mathbf{1}^\top [g(z)]_+ + \kappa_{\mathrm{tp}} \|Az\|_1 + \kappa_g \|g(z)\|_2 + \kappa_r \|Rg(z)\|_2 + \tfrac{\rho}{2}\|z\|_2^2, \tag{61}$$

where $\Delta = 20$, $\kappa_{\mathrm{tp}} = 2$, $\kappa_g = 8$, $\kappa_r = 5$, and $\rho = 0.1$.

For each sampled schedule, the labels are the normalized cost and its exact decision gradient. Each training day contributes 12 schedules and each validation day eight. The zero schedule is always included, 20% of the remaining schedules place random coordinates on box faces, and the rest lie in the box interior. Cost values and gradients use the same affine normalization estimated from the training set. The loss is

$$\mathcal{L} = \frac{1}{N}\sum_i (\widehat{F}_\theta - \widetilde{C}_i)^2 + 0.05 \frac{1}{N}\sum_i \|\nabla_z \widehat{F}_\theta - \nabla_z \widetilde{C}_i\|_2^2.$$

We train three context-conditioned convex architectures under an approximately matched 19k-parameter budget and context width 64. ReLU-ICNN uses the ReLU backbone and a strongly convex ridge; Quad-ICNN adds ten quadratic modules; SOC-ICNN uses six quadratic modules and two norm modules of residual dimension 24. All models use ridge floor 10^{-3}. Their configurations are listed in Table 27.

Adam uses at most 80 epochs, batch size 256, learning rate 2×10^{-3}, weight decay 10^{-6}, gradient clipping at 10, validation every five epochs, and early-stopping patience eight. Each architecture is trained with seeds $\{0, 1, 2\}$.

Table 22: **Box results by regime.** Success and Certified are percentages; residuals and runtimes are medians over all five runs per row.

Regime	Method	Success	Certified	Residual	Time (s)
constraint boundary	AD-PGD	100.0%	100.0%	1.806×10^{-5}	0.1502
constraint boundary	DCI-C-Full	100.0%	100.0%	1.395×10^{-10}	0.0062
constraint boundary	DCI-C-Local	100.0%	100.0%	1.395×10^{-10}	0.0050
constraint boundary	DCI-I-Local	100.0%	100.0%	1.806×10^{-5}	0.1826
constraint boundary	Prox-Only	100.0%	100.0%	4.082×10^{-6}	0.1793
constraint boundary	Readout-PGD	100.0%	100.0%	1.806×10^{-5}	0.0751
joint boundary	AD-PGD	0.0%	0.0%	2.239×10^{-1}	0.0880
joint boundary	DCI-C-Full	100.0%	100.0%	1.025×10^{-9}	0.2139
joint boundary	DCI-C-Local	100.0%	100.0%	1.025×10^{-9}	0.2845
joint boundary	DCI-I-Local	60.0%	60.0%	1.960×10^{-5}	0.4427
joint boundary	Prox-Only	100.0%	100.0%	5.256×10^{-6}	2.4852
joint boundary	Readout-PGD	0.0%	0.0%	2.239×10^{-1}	0.1401
regular	AD-PGD	100.0%	100.0%	1.512×10^{-5}	0.0669
regular	DCI-C-Full	100.0%	100.0%	1.337×10^{-9}	0.0037
regular	DCI-C-Local	100.0%	100.0%	1.337×10^{-9}	0.0034
regular	DCI-I-Local	100.0%	100.0%	1.512×10^{-5}	0.0540
regular	Prox-Only	100.0%	100.0%	1.400×10^{-5}	0.0843
regular	Readout-PGD	100.0%	100.0%	1.512×10^{-5}	0.0228
stress	AD-PGD	100.0%	0.0%	5.757×10^{-10}	0.1423
stress	DCI-C-Full	100.0%	100.0%	5.757×10^{-10}	0.2568
stress	DCI-C-Local	100.0%	100.0%	5.757×10^{-10}	0.2451
stress	DCI-I-Local	0.0%	0.0%	1.080×10^{0}	0.2469
stress	Prox-Only	100.0%	100.0%	5.757×10^{-10}	1.4811
stress	Readout-PGD	100.0%	0.0%	5.757×10^{-10}	0.1811

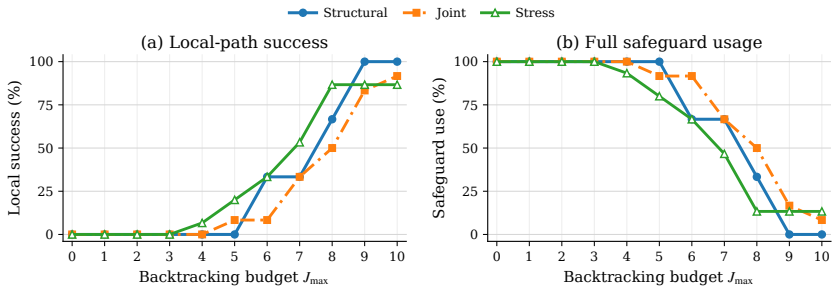


Figure 2: **Finite line-search budget and safeguard.** Increasing $J_{\max}$ resolves more instances locally and reduces safeguard use; the Full method remains successful throughout.

D.6.1 Experiment 5A: Inference on Frozen SOC-ICNN Objectives

Frozen-objective inference protocol. For a selected test day and seed, the context and trained parameters are fixed, so every solver minimizes the same box-constrained function

$$z \mapsto \widehat{F}_{\theta_s}(z; c_r), \quad z \in [-1, 1]^{12}.$$

Thirty evenly spaced 2019 days for each of three seeds give 90 objectives, and every method starts from $z = 0$. We compare AD-GD, Readout-GD, AD-L-BFGS-B, Canonical-Newton, DCI-I, DCI-C, and the high-accuracy Lifted-PD reference. Local methods receive 120 iterations and use the line search in the main algorithm, with slope coefficient 0.1, Armijo coefficient 10^{-4}, contraction 0.5, and at most 30 reductions. The structural tolerance is 10^{-7}. DCI-C uses damping 10^{-5} and metric eigenvalue bounds $[10^{-3}, 10^{3}]$; Lifted-PD uses 30,000 iterations and tolerance 10^{-9}. Success requires learned-objective gap at most 10^{-6} and an independently recomputed complete residual at most 2×10^{-5}.

Table 23: **Euclidean ball results by regime.** Success and Certified are percentages; residuals and runtimes are medians over all five runs per row.

Regime	Method	Success	Certified	Residual	Time (s)
constraint boundary	AD-PGD	0.0%	0.0%	3.217×10^{-5}	0.0812
constraint boundary	DCI-C-Full	100.0%	100.0%	1.839×10^{-5}	0.0420
constraint boundary	DCI-C-Local	100.0%	100.0%	1.839×10^{-5}	0.0414
constraint boundary	DCI-I-Local	100.0%	100.0%	1.760×10^{-5}	0.1056
constraint boundary	Prox-Only	100.0%	100.0%	1.030×10^{-5}	0.1185
constraint boundary	Readout-PGD	0.0%	0.0%	3.217×10^{-5}	0.0419
joint boundary	AD-PGD	0.0%	0.0%	2.526×10^{-1}	0.0787
joint boundary	DCI-C-Full	100.0%	100.0%	1.516×10^{-5}	0.5507
joint boundary	DCI-C-Local	100.0%	100.0%	1.516×10^{-5}	0.5443
joint boundary	DCI-I-Local	20.0%	20.0%	5.822×10^{-1}	0.1685
joint boundary	Prox-Only	100.0%	100.0%	7.338×10^{-6}	1.6459
joint boundary	Readout-PGD	0.0%	0.0%	1.885×10^{-1}	0.0341
regular	AD-PGD	100.0%	100.0%	1.365×10^{-5}	0.0484
regular	DCI-C-Full	100.0%	100.0%	3.612×10^{-8}	0.0028
regular	DCI-C-Local	100.0%	100.0%	3.612×10^{-8}	0.0028
regular	DCI-I-Local	100.0%	100.0%	1.365×10^{-5}	0.0390
regular	Prox-Only	100.0%	100.0%	5.834×10^{-6}	0.0709
regular	Readout-PGD	100.0%	100.0%	1.365×10^{-5}	0.0205
stress	AD-PGD	0.0%	0.0%	5.059×10^{-1}	0.1154
stress	DCI-C-Full	100.0%	100.0%	3.977×10^{-10}	0.5326
stress	DCI-C-Local	60.0%	60.0%	4.335×10^{-9}	0.4125
stress	DCI-I-Local	20.0%	20.0%	4.647×10^{-1}	0.3246
stress	Prox-Only	100.0%	100.0%	3.977×10^{-10}	1.9995
stress	Readout-PGD	20.0%	0.0%	3.697×10^{-1}	0.0929

D.6.2 Experiment 5B: Operational Scheduling

Operational comparison. Unlike Experiment 5A, this experiment compares complete decision pipelines. Every returned schedule is evaluated under the realized engineering cost $C(z; n_r, \lambda_r)$ in equation 61. The learned pipelines minimize their ReLU-, Quad-, or SOC-ICNN surrogate using DCI-C. Point Forecast and Historical SAA instead construct explicit convex engineering objectives: Point Forecast substitutes the forecasts available before dispatch, whereas Historical SAA averages the realized objectives associated with the seven nearest standardized training contexts. The Perfect Information (PI) oracle uses the realized test-day net load and is therefore not deployable; it defines the within-basis reference decision. No Storage uses $z = 0$.

For method m on test day r, operational regret is

$$\mathrm{Regret}_r(m) = C(z_r^m; n_r, \lambda_r) - C(z_r^{\mathrm{PI}}; n_r, \lambda_r).$$

Thus, Point Forecast and Historical SAA compare learned objectives with explicit engineering models under the same information timing, while PI quantifies the remaining value of unavailable future information. All methods use the same 12-dimensional feasible battery basis, and all 363 days of 2019 are evaluated. The learned-model results aggregate three seeds, giving 1089 seed–day solves per architecture. "Certified" means that the returned decision satisfies the complete optimality certificate for the objective that the corresponding pipeline actually minimizes; it does not certify agreement between a learned surrogate and the realized cost. Cost and regret are daily means, peak exchange is the mean daily value of $\max_t |g_t(z)|$, and ramp norm is the mean daily value of $\|Rg(z)\|_2$.

Paired uncertainty and error attribution. We use a seven-day circular moving-block bootstrap with 2000 repetitions and seed 2027. In Table 32, a positive difference means that the first pipeline has higher realized cost.

Table 24: **Simplex results by regime.** Success and Certified are percentages; residuals and runtimes are medians over all five runs per row.

Regime	Method	Success	Certified	Residual	Time (s)
constraint boundary	AD-PGD	100.0%	100.0%	1.741×10^{-5}	0.1649
constraint boundary	DCI-C-Full	100.0%	100.0%	1.510×10^{-9}	0.0399
constraint boundary	DCI-C-Local	100.0%	100.0%	1.510×10^{-9}	0.0409
constraint boundary	DCI-I-Local	100.0%	100.0%	1.514×10^{-5}	1.3149
constraint boundary	Prox-Only	100.0%	100.0%	5.041×10^{-6}	0.6324
constraint boundary	Readout-PGD	100.0%	100.0%	1.741×10^{-5}	0.0937
joint boundary	AD-PGD	40.0%	0.0%	1.124×10^{-1}	0.1806
joint boundary	DCI-C-Full	100.0%	100.0%	1.265×10^{-9}	1.3852
joint boundary	DCI-C-Local	60.0%	60.0%	3.154×10^{-9}	0.7801
joint boundary	DCI-I-Local	60.0%	60.0%	3.154×10^{-9}	0.5540
joint boundary	Prox-Only	100.0%	100.0%	1.265×10^{-9}	6.3220
joint boundary	Readout-PGD	20.0%	0.0%	1.261×10^{-1}	0.1028
regular	AD-PGD	100.0%	100.0%	1.387×10^{-5}	0.0515
regular	DCI-C-Full	100.0%	100.0%	7.056×10^{-11}	0.0044
regular	DCI-C-Local	100.0%	100.0%	7.056×10^{-11}	0.0043
regular	DCI-I-Local	100.0%	100.0%	1.387×10^{-5}	0.0496
regular	Prox-Only	100.0%	100.0%	1.029×10^{-5}	0.0879
regular	Readout-PGD	100.0%	100.0%	1.387×10^{-5}	0.0251
stress	AD-PGD	100.0%	0.0%	4.168×10^{-10}	0.2917
stress	DCI-C-Full	100.0%	100.0%	4.168×10^{-10}	0.9822
stress	DCI-C-Local	100.0%	100.0%	4.168×10^{-10}	0.9567
stress	DCI-I-Local	60.0%	60.0%	1.519×10^{-9}	0.5613
stress	Prox-Only	100.0%	100.0%	4.168×10^{-10}	4.8295
stress	Readout-PGD	80.0%	0.0%	1.224×10^{-9}	0.2797

Among the 1089 SOC-ICNN operational solves, 1079 (99.1%) satisfy the complete certificate. Their mean operational regret is 2.1654, nearly unchanged from 2.1733 over all solves, and the noncertified runs contribute only 0.0278 to the overall mean. The operational gap is therefore primarily attributable to approximation of the realized engineering objective, rather than failure of DCI to minimize the learned SOC-ICNN objective.

Table 25: **Second-order cone results by regime.** Success and Certified are percentages; residuals and runtimes are medians over all five runs per row.

Regime	Method	Success	Certified	Residual	Time (s)
constraint boundary	AD-PGD	0.0%	0.0%	2.897×10^{-5}	0.0899
constraint boundary	DCI-C-Full	100.0%	100.0%	1.773×10^{-5}	0.0226
constraint boundary	DCI-C-Local	100.0%	100.0%	1.773×10^{-5}	0.0243
constraint boundary	DCI-I-Local	100.0%	100.0%	1.911×10^{-5}	0.1290
constraint boundary	Prox-Only	100.0%	100.0%	3.631×10^{-6}	0.1446
constraint boundary	Readout-PGD	0.0%	0.0%	2.897×10^{-5}	0.0479
joint boundary	AD-PGD	0.0%	0.0%	2.870×10^{-1}	0.0569
joint boundary	DCI-C-Full	100.0%	100.0%	1.317×10^{-5}	0.4419
joint boundary	DCI-C-Local	100.0%	100.0%	1.317×10^{-5}	0.4420
joint boundary	DCI-I-Local	40.0%	40.0%	5.425×10^{-1}	0.5898
joint boundary	Prox-Only	100.0%	100.0%	4.078×10^{-6}	1.2838
joint boundary	Readout-PGD	0.0%	0.0%	2.761×10^{-1}	0.2140
regular	AD-PGD	100.0%	100.0%	1.251×10^{-5}	0.0473
regular	DCI-C-Full	100.0%	100.0%	1.360×10^{-8}	0.0029
regular	DCI-C-Local	100.0%	100.0%	1.360×10^{-8}	0.0028
regular	DCI-I-Local	100.0%	100.0%	1.251×10^{-5}	0.0423
regular	Prox-Only	100.0%	100.0%	6.203×10^{-6}	0.0681
regular	Readout-PGD	100.0%	100.0%	1.251×10^{-5}	0.0194
stress	AD-PGD	0.0%	0.0%	3.948×10^{-1}	0.1193
stress	DCI-C-Full	100.0%	100.0%	1.452×10^{-9}	0.6874
stress	DCI-C-Local	100.0%	100.0%	1.452×10^{-9}	0.6986
stress	DCI-I-Local	0.0%	0.0%	1.600×10^{0}	0.1581
stress	Prox-Only	100.0%	100.0%	1.452×10^{-9}	1.6843
stress	Readout-PGD	20.0%	0.0%	3.182×10^{-1}	0.1657

Table 26: **Matched Local/Full safeguard ablation.** Local is success rate (%), SG is Full safeguard use (%), and $Q_{.95}$ s is the 95th percentile of Full runtime in seconds. Full success and certified termination are 100% in every cell. The number n is the number of paired objectives per budget.

	Structural boundary ($n = 3$)			Joint boundary ($n = 12$)			Stress ($n = 15$)		
$J_{\max}$	Local	SG	$Q_{.95}$ s	Local	SG	$Q_{.95}$ s	Local	SG	$Q_{.95}$ s
0	0.0	100.0	0.277	0.0	100.0	3.971	0.0	100.0	1.970
1	0.0	100.0	0.236	0.0	100.0	3.903	0.0	100.0	1.924
2	0.0	100.0	0.243	0.0	100.0	4.555	0.0	100.0	2.135
3	0.0	100.0	0.260	0.0	100.0	5.268	0.0	100.0	2.339
4	0.0	100.0	0.696	0.0	100.0	4.117	6.7	93.3	3.037
5	0.0	100.0	0.722	8.3	91.7	4.298	20.0	80.0	2.251
6	33.3	66.7	0.772	8.3	91.7	4.432	33.3	66.7	1.881
7	33.3	66.7	1.196	33.3	66.7	3.095	53.3	46.7	1.571
8	66.7	33.3	0.653	50.0	50.0	3.178	86.7	13.3	1.754
9	100.0	0.0	0.300	83.3	16.7	3.098	86.7	13.3	1.756
10	100.0	0.0	0.291	91.7	8.3	3.431	86.7	13.3	1.790

Table 27: **Matched learned convex model configurations.**

Model	Width	Params.	Quad	Norm	Ridge
ReLU-ICNN	117	19,135	–	–	10^{-3}
Quad-ICNN	107	19,135	10	–	10^{-3}
SOC-ICNN	64	19,167	6	2×24	10^{-3}

Table 28: **Frozen-objective inference reliability on 90 SOC-ICNN objectives.**

Method	Success	Wrong stop	Gap	Median res.	$Q_{.95}$ res.
AD-GD	92.2%	0.0%	6.271×10^{-15}	1.348×10^{-5}	1.050×10^{0}
Readout-GD	94.4%	0.0%	6.190×10^{-15}	1.339×10^{-5}	4.890×10^{-4}
AD-L-BFGS-B	94.4%	5.6%	0	9.802×10^{-8}	4.103×10^{-2}
Canonical-Newton	93.3%	0.0%	8.575×10^{-16}	4.559×10^{-6}	3.945×10^{-1}
DCI-I	96.7%	0.0%	6.081×10^{-15}	1.316×10^{-5}	1.936×10^{-5}
DCI-C	100.0%	0.0%	2.101×10^{-16}	1.183×10^{-6}	1.677×10^{-5}
Lifted-PD	100.0%	0.0%	0	1.384×10^{-9}	2.789×10^{-9}

Table 29: **Frozen-objective inference computation.** SG is safeguard invocation.

Method	Median iter.	Median s	$Q_{.95}$ s	SG
AD-GD	17.5	0.0204	0.0449	0.0%
Readout-GD	18.0	0.0149	0.0468	0.0%
AD-L-BFGS-B	7.0	0.0070	0.0756	0.0%
Canonical-Newton	3.0	0.0258	0.1835	0.0%
DCI-I	18.0	0.0210	0.0896	2.2%
DCI-C	2.0	0.0031	0.0250	3.3%
Lifted-PD	537.5	0.2414	0.7436	0.0%

Table 30: **Out-of-time operational decision quality.** Cost and regret are evaluated using the realized engineering objective.

Pipeline	Cost	Regret	Peak exchange	Ramp norm
No Storage	865.4857	1.1567	0.7959	0.1342
Point Forecast	864.3345	0.0054	0.7904	0.1272
Historical SAA	864.3394	0.0104	0.7918	0.1303
ReLU-ICNN	866.9870	2.6580	0.8258	0.2158
Quad-ICNN	866.3496	2.0206	0.8082	0.1746
SOC-ICNN	866.5023	2.1733	0.8135	0.1889
PI oracle	864.3290	0.0000	0.7901	0.1266

Table 31: **Operational certification and computation.** SG denotes safeguard invocation.

Pipeline	Certified	Median s	$Q_{.95}$ s	Mean s	SG
No Storage	–	0.00000	0.0000	0.0000	0.0%
Point Forecast	100.0%	0.96179	8.1375	1.9420	0.0%
Historical SAA	100.0%	6.77422	33.5591	10.7374	0.0%
ReLU-ICNN	97.8%	0.00224	0.1225	0.0217	0.0%
Quad-ICNN	99.6%	0.00189	0.0624	0.0121	0.0%
SOC-ICNN	99.1%	0.00305	0.0738	0.1762	1.6%
PI oracle	100.0%	0.99607	5.8912	1.8159	0.0%

Table 32: **Paired moving-block bootstrap for realized operating cost.**

Difference	Mean	95% CI	$P(A < B)$
SOC-ICNN − ReLU-ICNN	-0.4847	$[-0.6659, -0.3272]$	1.000
SOC-ICNN − Quad-ICNN	0.1526	$[0.0276, 0.2623]$	0.009
SOC-ICNN − Historical SAA	2.1629	$[1.9397, 2.3885]$	0.000
SOC-ICNN − Point Forecast	2.1678	$[1.9502, 2.4059]$	0.000