FEASIBLE BY TOLERANCE OR BY GEOMETRY? RADIAL RETRACTIONS IN NEURAL SOLVERS FOR CONSTRAINED OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many neural solvers for parametric constrained programs correct the network's prediction with a differentiable projection or feasibility iteration, which approaches the feasible set from outside and stops at a small residual. Whether the corrected output passes a fixed feasibility check then depends on how this iteration was stopped. Evaluating all methods under one strict rule, we find that the strict feasibility rate of two recent solvers moves from 0% to 100% as the iteration budget, the floating-point precision or the stopping tolerance changes, while the trained network stays fixed. We propose RRA (radial retraction from an anchor), which reaches the boundary from inside instead. A differentiable Gauss–Newton interior map moves the network's proposal to a strictly feasible anchor in the basin the network selected, and a radial retraction stops at the first boundary contact on the ray from this anchor toward the proposal. The exit parameter of this ray is a closed-form root for affine, convex-quadratic, box and second-order-cone constraints, and a scan brackets it from below for trigonometric constraints. The returned point is then feasible for any strictly feasible anchor, whatever the tolerance, depth or convergence of the map that placed it. RRA passes the strict rule on every test instance and training seed of ten families of convex and nonconvex QP, QCQP and SOCP, and its optimality gap is lower than that of FSNet on all ten. It is on par with the most accurate correction baseline while training 5 to 14 times faster. Longer training, which still takes at most 55% of that baseline's training time, moves RRA ahead of it. RRA also solves the two nonconvex second-order-cone families on which that baseline returns almost no feasible point.

1 INTRODUCTION

A neural solver for a family of constrained programs $\min_y f_x(y)$ subject to $y \in \mathcal{C}_x$ maps the parameter x to a solution, which is useful only if it is feasible. Recent methods such as FSNet (Nguyen & Donti, 2025), ENFORCE (Lastrucci & Schweidtmann, 2025) and ΠNet (Grontas et al., 2026) enforce the constraints in the same way. A network predicts a point, an iterative correction moves this point toward $\mathcal{C}_x$ until a stopping rule is met, and training differentiates through the correction. This design reaches small constraint violations under the stopping rule each method is run with.

We compare all methods under one strict rule (Section 2) and, for two of them, also hold the trained network fixed while its stopping rule changes. The strict feasibility rate of ΠNet then rises from 0% to 100% as its iteration budget grows and its arithmetic changes from single to double precision, and the rate of ENFORCE rises from 9.7% to 100% as its stopping tolerance tightens (Figure 1b,c). Both rates change because the correction reaches the feasible set from outside. An iteration that starts at an infeasible prediction stops once its residual drops below ε, and its output can then lie outside $\mathcal{C}_x$ by up to ε. Whether this output passes a given rule depends on ε, on the iteration budget and on the arithmetic. *When feasibility comes from an iteration that approaches the set from outside, it is a property of the stopping rule, not of the learned model.*

Feasibility that does not depend on a stopping rule requires a construction. HardNet (Min & Azizan, 2024) enforces affine constraints in closed form, optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019) project exactly onto convex sets, and RAYEN (Tordesillas et al., 2023) maps the network

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.

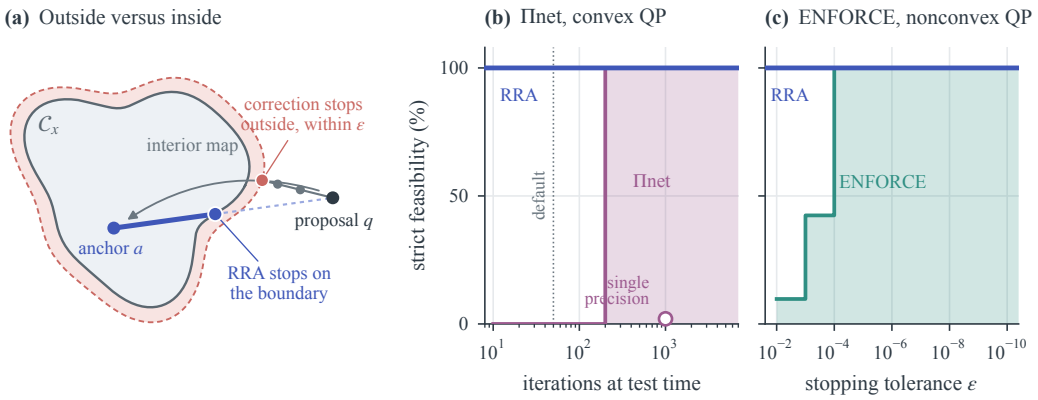


Figure 1: **Where a correction stops determines whether its output is feasible.** (a) An iteration that corrects the proposal q approaches the feasible set $\mathcal{C}_x$ from outside and stops on the dashed contour, where its residual falls below ε. RRA starts at a strictly feasible anchor a and stops where the ray toward q first crosses the boundary. (b) Strict feasibility rate of one trained ΠNet network on convex QP as its projection iterations at test time vary. (c) The same for one trained ENFORCE network on nonconvex QP as its stopping tolerance varies. The horizontal line is RRA, which has no stopping rule.

output along a ray from a fixed interior point of a convex set. Homeomorphic projection (Liang et al., 2023) reaches nonconvex sets homeomorphic to a ball through an invertible network trained on the set. These constructions either restrict the class of constraints or learn a global map of the set. A correction iteration, in contrast, works locally from the prediction and serves two purposes at once. It moves the prediction into the basin of $\mathcal{C}_x$ that the network has selected (the part of the set that a local iteration from the prediction reaches), and its derivative provides the training signal for the network. Returning a feasible point is its second purpose, which it can meet only up to its stopping tolerance.

RRA, for radial retraction from an anchor, keeps the first purpose and serves the second by reaching the boundary from inside (Figure 1a). Its *interior map* moves the network's proposal q to a strictly feasible anchor a in the selected basin, using Gauss–Newton steps that drive violated and nearly active constraints to a small margin inside the set. A *radial retraction* then follows the ray from this anchor toward q and stops where a constraint first becomes active. A null-space parametrization eliminates the equality constraints, and the exit parameter of the ray is then the smallest positive root of one scalar function per constraint. This root has a closed form for affine, convex-quadratic, box and second-order-cone constraints, and a scan with a second-order Taylor bound brackets it from below for trigonometric constraints. Because the retraction starts inside and stops at the first crossing, the feasibility of its output is a property of this geometry and holds for any strictly feasible anchor, whatever the tolerance, depth or convergence of the map that placed it.

The paper makes four contributions.

1. Under one strict rule, the feasibility of neural solvers built on correction iterations changes with the iteration budget, the precision and the stopping tolerance while the trained network stays fixed. We trace this dependence to the side from which the correction approaches the feasible set.
2. RRA reaches the boundary from inside through a differentiable interior map and a radial retraction whose exit parameter is computed in closed form or bracketed from below. Its output is feasible for any strictly feasible anchor (Proposition 1), and the construction covers trigonometric and nonconvex cone constraints without a learned map of the feasible set.
3. The evaluation covers ten families and nine baselines, with three training seeds for every method. RRA passes the strict rule on every test instance and seed, improves on FSNet on every family and is on par with ENFORCE while training 5 to 14 times faster.

4. Ablations attribute the accuracy of RRA to the Gauss–Newton interior map. Replacing this map by a fixed anchor, a shallower map or an Adam feasibility descent loses accuracy, while sliding along the boundary after the retraction becomes unnecessary once the anchor comes from the map. A proximity term between proposal and output matters on the families whose optima have many active constraints.

2 Problem Setting and the Strict Rule

We use the parametric-program benchmark of Donti et al. (2021) as extended by Nguyen & Donti (2025). Each instance is the program $\min_y f_x(y)$ subject to $Ay = x$, $g_x(y) \le 0$ and $-5 \le y \le 5$ for its own parameter x, with n=100 variables and 50 equality and 50 inequality constraints. The ten families are convex QP, QCQP and SOCP, convex QP at n=200 with 100 equality and 100 inequality constraints, nonconvex QP with $G\sin(y) \le h\cos(x)$, nonconvex QCQP with $\frac{1}{2}y^\top H_i y + G_i^\top \cos(y) \le h_i$, nonconvex SOCP with $\cos(y)$ inside the cone, and a variant of each of these three families with a term $0.1\|y\|_2$ added to the objective. All methods share one split of each family into 7000 training, 1000 validation and 2000 test instances, with 17,000 training instances at n=200 (Appendix A.1).

An output y passes the strict rule when its normalized residual

$$R_\infty(y) = \max\left\{\|Ay - x\|_\infty/\tau_{\text{eq}},\ \max_i\,(g_{x,i}(y))_+/\tau_{\text{ineq}},\ \max_j\,(|y_j| - 5)_+/\tau_{\text{box}}\right\}$$

is at most one. The tolerances are $\tau_{\text{eq}} = \tau_{\text{ineq}} = 10^{-4}$ for the equality and inequality rows and the tighter $\tau_{\text{box}} = 10^{-9}$ for the box, which every method can satisfy exactly. The strict feasibility rate is the fraction of all outputs that pass this rule, and the optimality gap $(f_x(y) - f_x^\star)/\max(|f_x^\star|, 1)$, in percent, is averaged over the outputs that pass it. One evaluator computes both quantities for all methods.

Clarabel (Goulart & Chen, 2026) computes the reference optimum $f_x^\star$ on the convex families. For the nonconvex families the benchmark provides a single local solution per instance, and even the spatial branch-and-bound of Gurobi (Gurobi Optimization, LLC, 2025) leaves a gap of 98 to 113% on nonconvex QP after 600 s per instance (Appendix D.2). Consequently, we replace this solution on the four nonconvex QCQP and SOCP families by a best-known reference, marked †, which keeps for each instance the best point that passes the strict rule among those SLSQP reaches from the benchmark solution and from the outputs of RRA and FSNet (Appendix D.1). Its mean objective on nonconvex SOCP is -5.72 against -0.62 for the benchmark solution, and an output better than every polished candidate has a negative gap. The two nonconvex QP families keep the benchmark solution, and on nonconvex QP the incumbents of Gurobi match it (Appendix D.2). Gap differences between two methods on one instance keep their sign under every reference, because the reference appears only in the normalization.

3 RRA: Reaching the Boundary from Inside

RRA computes its output in three steps (Algorithm 1). A network proposes a point $q_\theta(x)$, the interior map $\mathcal{I}$ moves the proposal to a strictly feasible anchor, and the radial retraction $\mathcal{R}$ stops at the first boundary contact on the ray from the anchor toward the proposal. The solver output is then $y(x) = y\big(x, \mathcal{R}(x, \mathcal{I}(x, q_\theta(x)), q_\theta(x))\big)$, where $y(x, p)$ maps null-space coordinates to the solution.

3.1 An interior anchor in the basin the network selects

Let $A \in \mathbb{R}^{m\times n}$ have full row rank, A^+ its pseudoinverse and $N \in \mathbb{R}^{n\times k}$, $k = n-m$, an orthonormal basis of its null space. Every y with $Ay = x$ is $y(x, p) = A^+x + Np$ for a unique $p \in \mathbb{R}^k$. Because the solver works in p, the equality constraints hold to machine precision at every point it evaluates (Lemma C.1). We write $c_x(p) \in \mathbb{R}^r$ for the inequality and box rows in these coordinates. The exact feasible set is $\mathcal{C}_x = \{p : c_x(p) \le 0\}$, and the strict rule enlarges it by its tolerances. A point with $c_x(p) < 0$ is strictly feasible, which is stronger than passing the strict rule. A multilayer perceptron with four hidden layers of width 1024, as in FSNet, maps x and a precomputed anchor $a_0(x)$ to the proposal $q_\theta(x) = a_0(x) + s\,\text{MLP}([\,x,\ a_0(x)/s\,])$ with scale s. This proposal need not be feasible.

The interior map performs D Gauss–Newton steps on at most w rows that are violated or lie within a margin μ of their boundary. With S the selected rows and $J_S = \nabla_p c_{x,S}(p)$, each step is

$$p \leftarrow p - J_S^\top \left(J_S J_S^\top + \rho I\right)^{-1} \left(c_{x,S}(p) + \mu \mathbf{1}\right), \tag{1}$$

which drives the selected rows to $-\mu$ at the cost of one linear solve of size at most w. We use $D = 20$ steps (10 on the nonconvex SOC families) on at most $w = 16$ rows at the margin $\mu = 10^{-4}$. Feasibility-seeking methods such as ENFORCE apply a step of the same kind to the prediction itself and aim at the boundary of $\mathcal{C}_x$, whereas the interior map aims at the margin μ inside it. Consequently, an early stop still leaves a strictly feasible anchor whenever the remaining error is smaller than μ. The solver checks the output of the map after D steps and takes the global anchor $a_0(x)$ when the check fails. A feasibility search computes this anchor once per instance and accepts it only when every row is strictly satisfied there.

3.2 A RETRACTION THAT STOPS AT THE FIRST CROSSING

The retraction follows the ray $p(t) = a + t(q - a)$, $t \geq 0$, from a strictly feasible anchor a toward the proposal q up to its first exit $t^\star = \inf\{t > 0 : c_x(p(t)) \not< 0\}$ and returns

$$\mathcal{R}(x, a, q) = a + \min(1, t^\star)\,(q - a). \tag{2}$$

This exit is the smallest first positive root among the rows $c_{x,i}(p(t))$, and for most rows it has a closed form. An affine or box row is linear in t along the ray and a convex-quadratic row is quadratic, while a second-order-cone row $\|G_i y + h_i\| \leq c_i^\top y + d_i$ becomes a quadratic after squaring and keeps the sign of the original up to its first root (Lemma C.2). Trigonometric rows, which occur in the nonconvex QP and QCQP families, have no closed-form root, and a scan brackets it instead by evaluating each such row on a grid and bisecting the first cell that contains a sign change. With M_i a bound on the second derivative along the ray, a second-order Taylor bound excludes an earlier crossing in every preceding cell (Lemma C.3), and the retraction returns the lower end of the final bracket, which lies below the root. The cone rows of the nonconvex SOC families contain $\cos(y)$ inside the norm, and the scan runs on them without this bound. It still returns the lower end of a bracket at which every row has been evaluated and found nonpositive.

Proposition 1 (Feasibility for any strictly feasible anchor)**.** *Let c_x be continuous and $c_x(a) < 0$. For every $q \in \mathbb{R}^k$ the point $\mathcal{R}(x, a, q)$ of Eq. (2) satisfies $c_x \leq 0$, and so does every point of the segment from a to $\mathcal{R}(x, a, q)$.*

The proof needs only the definition of the first exit and the continuity of c_x, since every row is negative before $t^\star$ and continuity makes it nonpositive at $t^\star$ (Appendix C.2). Because the argument uses neither the proposal nor the way the anchor was obtained, a few Gauss–Newton steps suffice for the interior map, whose tolerance, depth and convergence affect where the output lands but not whether it is feasible. This map, or the global anchor when its check failed, supplied a strictly feasible anchor on every test instance of the ten families and every seed. The work per instance is bounded in advance as well, by D linear systems of size at most w and one closed-form or bracketed exit per row, while an iteration stopped by a tolerance runs until its residual falls below it.

The two nonconvex SOC families need more than one anchor, because their reference optima have about 15 active rows and a single random start finds a strictly feasible anchor for only 13% of their instances (Appendix G.4). The solver keeps four anchors from sixteen starts on these families, follows each retraction with four sliding stages along the rows that stopped the ray and returns the best of the resulting points (Appendix E.1). A stage is kept only when every row satisfies a check well inside the tolerances of the strict rule, so that every output still passes this rule (Lemma C.5).

3.3 TRAINING THROUGH THE INTERIOR MAP

Gradients reach the network through both operators. The retraction differentiates its exit through the closed-form root, or through the implicit function theorem on $c_{x,i}(p(t^\star)) = 0$ when the exit comes from the scan (Lemma C.4). The interior map passes them through the linear solves of Eq. (1), and its derivative has a simple form.

Proposition 2 (Derivative of one step of the interior map)**.** *On an open set where the selected rows S are fixed, $c_{x,S}$ is C^2 and J_S has full row rank, one step of Eq. (1) at a point where the selected rows*

Algorithm 1 RRA inference for one instance x

1: $q \leftarrow a_0(x) + s\,\mathrm{MLP}([x, a_0(x)/s])$ proposal
2: $a \leftarrow \mathcal{I}(x, q)$ by D steps of Eq. (1); **if** $c_x(a) \not\leq 0$ **then** $a \leftarrow a_0(x)$ anchor
3: $t^\star \leftarrow$ first exit of the ray $a + t(q - a)$, in closed form or bracketed by the scan
4: $p \leftarrow a + \min(1, t^\star)(q - a)$ retraction
5: **return** $y = A^+x + Np$

equal $-\mu$ has the Jacobian $I - J_S^\top (J_S J_S^\top + \rho I)^{-1} J_S$, which tends to the orthogonal projector onto their tangent space as $\rho \to 0$. Elsewhere the Jacobian differs from this matrix by $O(\|c_{x,S}(p) + \mu \mathbf{1}\|)$.

The interior map composes D such steps into its Jacobian $J_{\mathcal{I}}$, and the retraction projects $t^\star I + (1 - t^\star) J_{\mathcal{I}}$ along the ray onto the tangent plane of the row that stops it (Lemma C.6). The ablations in Section 4.3 keep this projection and replace $J_{\mathcal{I}}$ by the derivative of other interior maps.

With $p(x) = \mathcal{R}(x, \mathcal{I}(x, q_\theta(x)), q_\theta(x))$ the retracted point, training minimizes

$$\mathcal{L}(\theta) = \mathbb{E}_x \Big[f_x\big(y(x, p(x))\big) + \lambda \left\| \big(q_\theta(x) - \mathrm{sg}[\,p(x)\,]\big)/s \right\|^2 \Big], \tag{3}$$

with $\lambda = 1$ and the stop-gradient sg. The second term keeps the proposal close to the point the retraction returns, so that the network controls where the output lands and not only the direction of the ray. This proximity term leaves feasibility untouched, and training uses no reference solution. We train with AdamW for 100 epochs (40 on the SOC families) and keep the checkpoint with the best validation objective (Appendix E.1).

4 Experiments

We train every learned method with seeds 2025, 2026 and 2027 on the fixed split, and every result below covers the three seeds. Four of the nine baselines correct the prediction from outside the feasible set, namely FSNet (Nguyen & Donti, 2025) at 500 epochs, ENFORCE (Lastrucci & Schweidtmann, 2025) with its adaptive-depth Gauss–Newton projection, ΠNet (Grontas et al., 2026) in double precision with 1000 projection iterations at test time, and DC3 (Donti et al., 2021). The remaining five are HardNet-Aff (Min & Azizan, 2024) and an exact projection with an OptNet layer (Amos & Kolter, 2017) on convex QP, a supervised network followed by an SLSQP projection, and penalty and adaptive-penalty training (Pan et al., 2019; Fioretto et al., 2020). Each baseline uses a backbone of the size RRA uses (Appendices E.3 to E.6). All methods were trained and timed on one machine with an NVIDIA L40S GPU (48 GB), an AMD EPYC 7R13 processor with 16 virtual cores and 128 GB of memory.

4.1 RRA passes the strict rule on every family and seed

Table 1 reports the lowest rate over the three seeds, and each iterative baseline falls below 100% on at least one family. FSNet does so on 5 families and reaches 87.8% on convex QP with n=200. ΠNet falls below 100% on 2 of 3 convex-QCQP seeds, with 95.8% at worst. ENFORCE passes the strict rule on at most 0.05% of its outputs on the two nonconvex SOC families, where its training does not converge (Appendix E.4). The exact projections, OptNet on convex QP and SLSQP on four families, stay at 100%, but the SLSQP projection takes up to 2.9 h per 2000 test instances (Table 16).

Figure 2 shows how far below the bar the outputs lie. Most outputs of RRA lie near 10^{-10}, the rounding error of the equality rows, and none reaches 10^{-6}. FSNet's outputs lie between 10^{-3} and the bar, with a tail beyond it. ENFORCE and ΠNet reach 10^{-10} where their iterations converge, and their other outputs spread up to the bar and beyond it.

The sweeps of Figure 1(b,c) hold one trained network of ΠNet and one of ENFORCE fixed and vary only how their iterations stop. ΠNet on convex QP stays at 0% up to 100 test iterations and in single precision, and it reaches 100% from 200 double-precision iterations on. ENFORCE on nonconvex QP goes from 9.7% at $\varepsilon = 10^{-2}$ to 100% from $\varepsilon = 10^{-4}$ on. Once the rate reaches 100%, the optimality gap is the same at every budget and tolerance of the two sweeps (Table 11). Hence the stopping rule changes whether an output counts, not what the network learned. The radial retraction stops at the first crossing and has no such rule to set.

Table 1: **Strict feasibility rate (%)**, the lowest over the training seeds 2025–2027, on the 2000 test instances of each family. The baselines are FSNet (Nguyen & Donti, 2025), ENFORCE (Lastrucci & Schweidtmann, 2025), ΠNet (Grontas et al., 2026), DC3 (Donti et al., 2021) and a supervised network followed by an SLSQP projection. / not applicable (ΠNet projects onto convex sets), – not run, † best-known references. Penalty and adaptive-penalty training reach 0% on every family they were run on. Per-seed rates, with those of HardNet-Aff and the OptNet projection, are in Appendix F.1.

	convex				nonconvex, + adds $0.1\|y\|_2$					
Method	QP	QCQP	SOCP	QP, n=200	QP	QP+	QCQP†	QCQP+†	SOCP†	SOCP+†
RRA (ours)	100	100	100	100	100	100	100	100	100	100
FSNet	100	100	100	87.8	100	99.9	99.9	100	99.7	99.7
ENFORCE	100	100	100	–	100	100	100	100	0.0	0.0
ΠNet	100	95.8	100	100	/	/	/	/	/	/
DC3	37.1	25.2	0.0	–	90.0	71.6	53.9	39.7	0.0	0.0
supervised + SLSQP	100	100	–	–	100	–	100	–	–	–

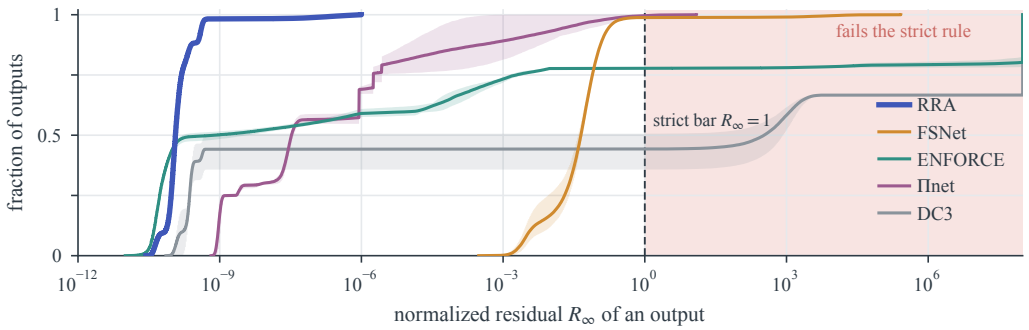


Figure 2: **Normalized residual of every output under the strict rule.** Each line is the empirical distribution of R_∞ over all outputs of one method, pooled over the families and seeds on which it was run, and its band gives the range over the three training seeds. Outputs right of the dashed bar fail the strict rule, and the height of a curve at the bar is the strict feasibility rate of its method.

4.2 Optimality gap relative to FSNet and ENFORCE

Table 2 compares the optimality gap of the outputs that pass the strict rule. RRA has a lower seed-mean gap than FSNet on all 10 families, and on the instances where both methods pass the strict rule it has the lower gap 82% of the time. The paired bootstrap interval of the mean difference lies below zero in 28 of the 30 comparisons of family and seed (Figure 3a). Both exceptions, one seed on each nonconvex QCQP family, have an interval above zero and a median that also favors FSNet.

Figure 3(b) shows how far the training seed moves the gap of each method. The seed standard deviation of RRA is the smallest of the four methods on all 10 families and stays at or below 10^{-3} percentage points on the convex ones, where FSNet reaches 0.16 points and ΠNet reaches 2.7. RRA with longer training also keeps a smaller seed standard deviation than ENFORCE on all seven families where ENFORCE has a strict feasibility rate of 100% on every seed (Table 12).

ENFORCE, the most accurate baseline, has a lower seed-mean gap than RRA on four of these seven families and a higher one on three, and the paired difference changes sign across seeds on five of them (Figure 3a). Its largest lead, 0.010 percentage points on nonconvex QP, disappears when RRA trains longer, still in a fraction of the training time of ENFORCE (Section 4.4). The remaining baselines, including the two exact projections on convex QP, are less accurate than RRA on every family they solve (Table 13). On convex QP, for example, the OptNet projection reaches 0.033% and HardNet-Aff reaches 0.123%, against 0.0021% for RRA.

Table 2: **Optimality gap (%)** on the outputs that pass the strict rule, mean over the training seeds 2025–2027, for the baselines of Table 1. Bold marks the lowest gap among RRA and the baselines with a strict feasibility rate of 100%. The second row is RRA with longer training (five times its epochs), which still takes at most 55% of the training time of ENFORCE (Figure 5). $^\circ$ gap on the outputs that pass the strict rule, for a method below 100% on some seed, $\times$ fewer than half of the outputs pass the strict rule, other marks as in Table 1. Per-seed values and standard deviations are in Table 12.

	convex				nonconvex, + adds $0.1\|y\|_2$					
Method	QP	QCQP	SOCP	QP, n=200	QP	QP+	QCQP†	QCQP+†	SOCP†	SOCP+†
RRA (ours)	0.0021	0.0059	**0.0055**	**0.0069**	0.128	0.091	**0.369**	**−0.0053**	**8.62**	**14.2**
RRA, longer training	0.0015	0.0034	0.0025	–	0.103	0.071	0.249	−0.134	–	–
FSNet	0.036	0.079	0.730	0.070°	0.145	0.125°	4.67°	0.682	24.8°	56.7°
ENFORCE	**0.0018**	**0.0053**	0.0056	–	**0.118**	**0.083**	0.518	0.016	$\times$	$\times$
ΠNet	0.051	3.12°	0.299	0.550	/	/	/	/	/	/
DC3	$\times$	$\times$	$\times$	–	2.48°	1.81°	15.7°	$\times$	$\times$	$\times$
supervised + SLSQP	0.073	0.219	–	–	0.250	–	6.25	–	–	–

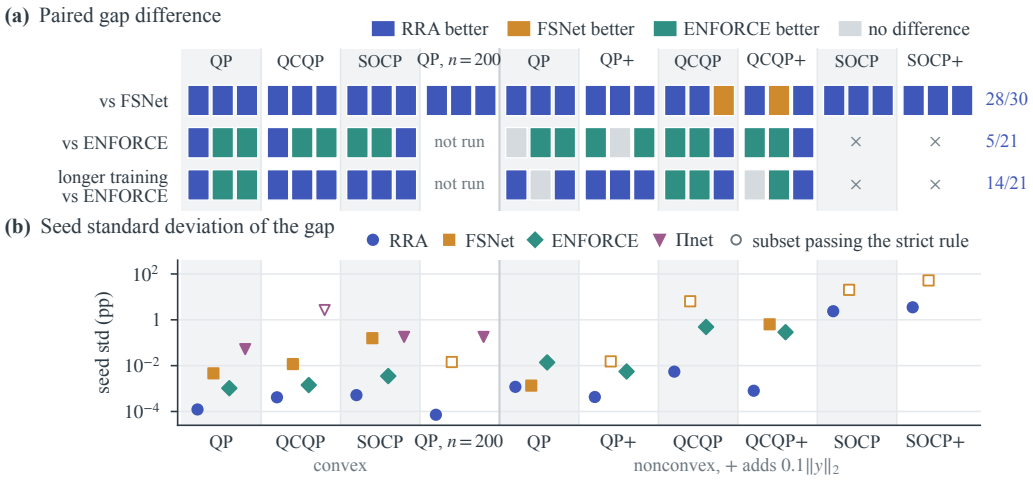


Figure 3: **Paired gap differences and the spread of the gap over training seeds.** Each column is one family, labeled above (a) and below (b). (a) Paired difference of the mean gap on the instances where both methods pass the strict rule, one tile per training seed (2025 to 2027 from left to right). A tile is gray when the 95% bootstrap interval over 10,000 replicates contains zero, and the number on the right counts the tiles that favor RRA. $\times$ fewer than half of the outputs of ENFORCE pass the strict rule. (b) Standard deviation of the gap over the three seeds, with hollow markers for a baseline whose strict feasibility rate is below 100% on some seed.

4.3 Ablations of the interior map and the proximity term

Figure 4(a) compares RRA on nonconvex QP with variants that change the interior map or the retraction. Replacing the interior map by a fixed global anchor multiplies the gap by 7.3, even with sliding stages that move the output along the boundary. A shallower interior map loses accuracy as well, and with the same sliding stages the gap rises from 0.124% at depth 20 to 0.176% for the best run at depth 10 and 0.345% at depth 5. Replacing the Gauss–Newton map by an Adam feasibility descent raises the gap 4.6 to 6.0 times, whether the gradient of that descent is detached, passed straight through, or replaced by the tangent projection of Proposition 2 around the Adam anchor. This last variant uses the projector of a converged Gauss–Newton step without the derivative of the map that produced its anchor.

(a) Variants on nonconvex QP

	Gauss–Newton map	exact derivative	depth D	sliding stages	gap (%)
RRA	●	●	20	0	0.127
with sliding stages	●	●	20	8	0.124 ×1.0
interior map depth 10	●	●	10	8	0.176 ×1.4
interior map depth 5	●	●	5	8	0.345 ×2.7
Adam map, straight-through	○	○	–	8	0.589 ×4.6
Adam map, tangent projection	○	○	–	8	0.621 ×4.9
Adam map, detached	○	○	–	8	0.767 ×6.0
fixed anchor, no interior map	○	○	–	8	0.928 ×7.3

(b) Proximity term

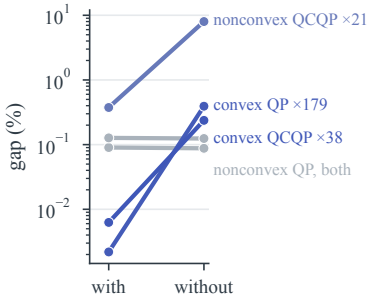
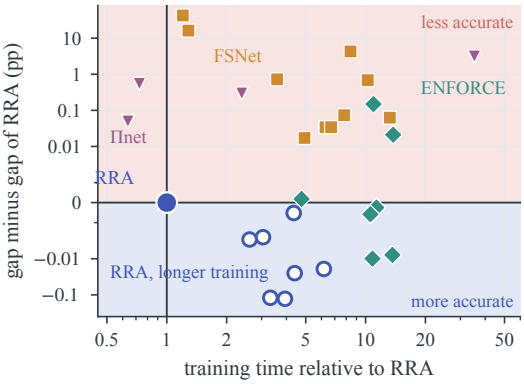


Figure 4: **Variants of the interior map, and runs without the proximity term.** (a) Variants of RRA on nonconvex QP (seed 2025, 100 epochs, every output feasible under the strict rule). Filled dots mark a Gauss–Newton interior map and its exact derivative, and the columns give the depth D and the number of sliding stages. The shading and the factor give the gap relative to RRA, whose seed standard deviation is 0.0012 percentage points, and each variant below the second row shows its best run over the settings tried (Table 17). (b) Gap with and without the proximity term on the five families where both runs exist, with the ratio in the label. The line color gives the number of active rows at the reference optimum, from 7 in gray to 16 in blue (Table 19).

(a) Accuracy against training time



(b) Inference time on one GPU

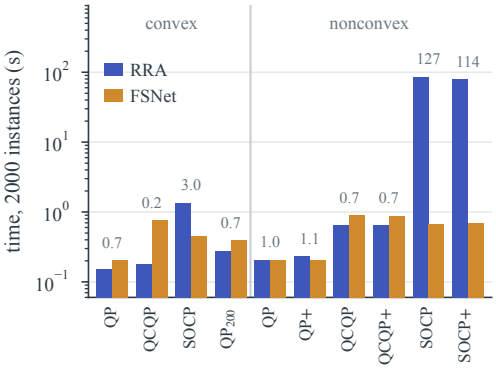


Figure 5: **Accuracy against training time, and inference time on one GPU.** (a) Each baseline on each family, placed by its training time relative to RRA and by its seed-mean gap minus that of RRA on a symmetric logarithmic axis. Points above zero are less accurate than RRA, and hollow circles are RRA with longer training. (b) Time for the 2000 test instances, measured back to back on the same GPU as the best of five repeats, with batch 2000 except for RRA on the three SOCP families (batch 500). The number above each pair is the ratio of RRA to FSNet. QP_{200} is convex QP with n=200, and Appendix F.4 lists the times of every method.

Sliding stages help a retraction from a fixed anchor reach optima with many active constraints, but even sixteen stages leave its gap several times that of RRA on convex QP (0.0098%) and convex QCQP (0.037%, Table 18). With the anchor from the interior map, eight sliding stages change the gap on nonconvex QP only from 0.127% to 0.124% and more than double the inference time (Figure 6). Consequently, RRA uses a single ray outside the two nonconvex SOC families.

Figure 4(b) compares runs with and without the proximity term of Eq. (3). Without the term the gap grows 21 to 179 times on the families whose reference optima have 12 to 16 active rows. By contrast, it does not change on the two nonconvex QP families, whose optima have about 7. An optimum with many active rows lies where many faces meet, so a small error in the ray direction

moves the exit point to another face. The proximity term lets the network place the output directly instead of steering it through the direction of the ray alone.

4.4 Training and inference time

Every FSNet point in Figure 5(a) lies above zero and to the right of RRA, because FSNet at 500 epochs is less accurate than RRA and trains longer on all 10 families. The ENFORCE points straddle zero, and RRA trains 5 to 14 times faster than ENFORCE on the seven families where ENFORCE has a strict feasibility rate of 100% on every seed. Both methods train for the same number of epochs on all of these families but convex SOCP, where RRA trains for fewer. RRA with longer training still uses at most 55% of the training time of ENFORCE, and its seed-mean gap is then lower on all 7 of these families (Table 2). The paired difference favors it in 14 of the 21 seed comparisons (Figure 3a). Hence the difference at the reported epochs reflects training length.

Inference time on the same GPU separates the two nonconvex SOC families from the rest (Figure 5b). On the other eight families RRA takes 0.2 to 3.0 times the time of FSNet and is faster on 5 of them. Its largest ratio is on convex SOCP, where cone rows appear in every Gauss–Newton step and RRA was timed with batch 500 instead of 2000 (Appendix E.2). RRA takes 79.9 to 86.0 s per 2000 instances on the two nonconvex SOC families, against 0.68 to 0.70 s for FSNet. Most of this time, 59 to 60%, goes to the multi-start anchor search that lets RRA solve these two families.

5 Related Work

Neural solvers for constrained programs, surveyed by Kotary et al. (2021), differ in how they reach the feasible set and from which side. Penalty and Lagrangian terms in the training loss (Pan et al., 2019; Fioretto et al., 2020) do not enforce the constraints, whereas feasibility-seeking corrections approach the set from outside and stop at a tolerance. These corrections are equality completion and gradient steps in DC3 (Donti et al., 2021), a differentiated quasi-Newton minimization of the violation in FSNet (Nguyen & Donti, 2025), an adaptive-depth Gauss–Newton projection on a Fischer–Burmeister reformulation (Fischer, 1992) in ENFORCE (Lastrucci & Schweidtmann, 2025), and operator splitting in ΠNet (Grontas et al., 2026). The interior map of RRA is an iteration of the same kind, but it aims at a target inside the set and its output serves only as an anchor. Structural constructions, in contrast, enforce feasibility without a stopping rule. Differentiable optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019) project exactly onto convex sets, and HardNet (Min & Azizan, 2024) enforces input-dependent affine constraints in closed form. Gauge maps (Tabas & Zhang, 2022) scale the output of a policy onto a polytope along rays from an interior point, and RAYEN (Tordesillas et al., 2023) does the same on a convex set, with a closed-form step for linear, quadratic, second-order-cone and matrix inequality constraints. RAYEN computes this point once, for constraints that do not depend on the input. The radial retraction builds on the same ray construction but takes its anchor per instance from a differentiated map in the basin the network selects. The retraction also covers trigonometric rows through the bracketing scan, and its guarantee holds when the anchor comes from a local iteration. Homeomorphic projection (Liang et al., 2023) also bisects along rays, but in the latent ball of a learned invertible map rather than on the constraints (Table 4).

6 Conclusion

Because a correction iteration reaches the feasible set from outside and stops at a tolerance, the feasibility of a neural solver built on it depends on the stopping rule. RRA reaches the boundary from inside instead, with a differentiated interior map that places a strictly feasible anchor in the selected basin and a radial retraction that computes or brackets the exit of the ray toward the proposal. Its output is feasible for any strictly feasible anchor, at a cost per instance that is bounded in advance. RRA passes the strict rule on every instance and seed of the ten families, including the two on which ENFORCE returns almost no feasible point. It improves on FSNet on every family and is on par with ENFORCE in a fraction of its training time. Its accuracy comes from the differentiated interior map. Our results suggest reporting the stopping rule together with the feasibility rate of neural solvers that rely on an iteration. Feasibility by geometry extends to every constraint class whose first exit along a ray can be computed or bracketed from below.

AI USE STATEMENT

We used large language models to modify experiment and plotting code under the authors' direction and to polish the language of the manuscript. The authors reviewed every code change and checked all experiments, numbers and claims against the evaluation records. We take responsibility for the final content of this work, including the code and text edited with these tools.

ETHICS STATEMENT

This work concerns numerical methods for parametric optimization on synthetic benchmark families and involves no human subjects, personal data or deployment. We see no direct ethical risk beyond the general one that a feasibility guarantee holds for the constraint model as written and not for a mis-specified model.

REPRODUCIBILITY STATEMENT

Code, the split indices, the reference solutions with their sources and the per-instance evaluation records of every method will be released. Every number in the paper comes from these records. Appendix E.1 lists the settings of RRA, Appendix E.2 the hardware and software, Appendices E.3 to E.6 the baseline implementations, and Appendix D.1 the construction of the references.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane T. Barratt, Stephen P. Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9558–9570, 2019.

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

Priya L. Donti, David Rolnick, and J. Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations (ICLR)*, 2021.

Ferdinando Fioretto, Terrence W. K. Mak, and Pascal Van Hentenryck. Predicting AC optimal power flows: Combining deep learning and Lagrangian dual methods. In *AAAI Conference on Artificial Intelligence*, pp. 630–637, 2020. doi: 10.1609/AAAI.V34I01.5403.

Andreas Fischer. A special Newton-type optimization method. *Optimization*, 24(3–4):269–284, 1992. doi: 10.1080/02331939208843795.

Paul J. Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. *Mathematical Programming Computation*, 2026. doi: 10.1007/s12532-026-00320-7.

Panagiotis D. Grontas, Antonio Terpin, Efe C. Balta, Raffaello D'Andrea, and John Lygeros. Pinet: Optimizing hard-constrained neural networks with orthogonal projection layers. In *International Conference on Learning Representations (ICLR)*, 2026.

Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2025. URL `https://www.gurobi.com`.

James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder. End-to-end constrained optimization learning: A survey. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4475–4482, 2021. doi: 10.24963/IJCAI.2021/610.

Giacomo Lastrucci and Artur M. Schweidtmann. ENFORCE: Nonlinear constrained learning with adaptive-depth neural projection. *arXiv preprint arXiv:2502.06774*, 2025. doi: 10.48550/arXiv.2502.06774.

Enming Liang, Minghua Chen, and Steven H. Low. Low complexity homeomorphic projection to ensure neural-network solution feasibility for optimization over (Non-)Convex set. In *International Conference on Machine Learning (ICML)*, pp. 20623–20649, 2023.

Youngjae Min and Navid Azizan. HardNet: Hard-constrained neural networks with universal approximation guarantees. *arXiv preprint arXiv:2410.10807*, 2024. doi: 10.48550/arXiv.2410.10807.

Hoang T. Nguyen and Priya L. Donti. FSNet: Feasibility-seeking neural network for constrained optimization with guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xiang Pan, Tianyu Zhao, and Minghua Chen. DeepOPF: Deep neural network for DC optimal power flow. In *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2019. doi: 10.1109/SMARTGRIDCOMM.2019.8909795.

Daniel Tabas and Baosen Zhang. Computationally efficient safe reinforcement learning for power systems. In *American Control Conference (ACC)*, pp. 3303–3310, 2022. doi: 10.23919/ACC53348.2022.9867652.

Jesus Tordesillas, Victor Klemm, Jonathan P. How, and Marco Hutter. RAYEN: Imposition of hard convex constraints on neural networks. *arXiv preprint arXiv:2307.08336*, 2023. doi: 10.48550/arXiv.2307.08336.

A THE BENCHMARK AND THE EVALUATION RULE

A.1 FAMILIES AND SPLIT

All families follow the generator of Donti et al. (2021) as extended by Nguyen & Donti (2025). Each instance is the program $\min_y f_x(y)$ subject to $Ay = x$, $g_x(y) \le 0$ and $-5 \le y \le 5$, with the parameter $x \in \mathbb{R}^m$ drawn per instance and the matrices fixed per family. There are n=100 variables, m=50 equality constraints and 50 inequality rows, except for convex QP at n=200 with 100 equality and 100 inequality constraints. Table 3 lists the objective and the inequality rows. Three further families add $0.1\|y\|_2$ to the objective of the nonconvex QP, QCQP and SOCP families. In the null-space coordinates this term is $0.1\sqrt{\|A^+x\|^2 + \|p\|^2}$.

Table 3: Objective and inequality rows of the families (trigonometric functions act elementwise).

Family	objective $f_x(y)$	inequality rows $g_{x,i}(y)$
convex QP	$\frac{1}{2}y^\top Qy + p^\top y, Q \succeq 0$	$G_i^\top y - h_i$
convex QCQP	$\frac{1}{2}y^\top Qy + p^\top y$	$\frac{1}{2}y^\top H_i y + G_i^\top y - h_i, H_i \succeq 0$
convex SOCP	$\frac{1}{2}y^\top Qy + p^\top y$	$\|G_i y + h_i\|_2 - (c_i^\top y + d_i)$
nonconvex QP	$\frac{1}{2}y^\top Qy + p^\top \sin(y)$	$G_i^\top \sin(y) - h_i \cos(x_i)$
nonconvex QCQP	$\frac{1}{2}y^\top Qy + p^\top \sin(y)$	$\frac{1}{2}y^\top H_i y + G_i^\top \cos(y) - h_i$
nonconvex SOCP	$\frac{1}{2}y^\top Qy + p^\top \sin(y)$	$\|G_i \cos(y) + h_i\|_2 - (c_i^\top y + d_i)$

Every dataset holds 10,000 instances (20,000 at n=200). As in the training code of FSNet, the split is a random permutation with seed 2025 into 7000 training, 1000 validation and 2000 test instances (17,000/1000/2000 at n=200), and every method is evaluated on the same test instances.

A.2 THE STRICT RULE

One evaluator produces every number in the paper. For a returned point y on instance x it forms

$$R_\infty(y) = \max\left\{ \frac{\|Ay - x\|_\infty}{\tau_{\text{eq}}},\ \frac{\max_i (g_{x,i}(y))_+}{\tau_{\text{ineq}}},\ \frac{\max_j (|y_j| - 5)_+}{\tau_{\text{box}}} \right\} \tag{4}$$

and counts the output as passing the strict rule when $R_\infty \le 1$. The gap is the signed relative difference between the objective and the reference, averaged over the outputs that pass the rule. A method that passes the rule on only a subset of the instances is compared on that subset, which the tables mark.

B CONSTRUCTIONS THAT RETURN A FEASIBLE OUTPUT

The constructions of Section 5 differ in what they ask of the feasible set and of the point they start from, and Table 4 places these differences side by side. The first row is the design that Section 4.1 evaluates under the strict rule, and the last row is the one this paper proposes.

Table 4: How each construction obtains a feasible output, which constraints it covers, and which interior point it needs. The rows cite, in order, DC3, FSNet, ENFORCE and ΠInet (Donti et al., 2021; Nguyen & Donti, 2025; Lastrucci & Schweidtmann, 2025; Grontas et al., 2026), the optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019), HardNet (Min & Azizan, 2024), the gauge map (Tabas & Zhang, 2022) and RAYEN (Tordesillas et al., 2023), and homeomorphic projection (Liang et al., 2023).

Construction	Where feasibility comes from	Constraint classes	Interior reference point
Correction iteration	an iteration from the prediction that stops at a residual ε and can end outside by up to ε	differentiable rows, convex sets for ΠInet	none
Optimization layer	the projection is solved exactly	convex	none
HardNet	a closed-form correction	affine rows that may depend on the input	none
Gauge map, RAYEN	the output is scaled along a ray from the reference point	affine, convex quadratic, second-order cone, matrix inequality, all convex	one interior point per constraint set, computed once
Homeomorphic projection	a ray is bisected in the latent ball of a learned invertible map	sets homeomorphic to a ball	one interior point and an invertible map trained per family
RRA (this paper)	the ray from the anchor stops at the first boundary contact	affine, convex quadratic, box, second-order cone, rows with a curvature bound along the ray, and nonconvex cone rows checked at the returned point	one strictly feasible anchor per instance, from a differentiated map of the proposal

C PROOFS

Throughout, N is an orthonormal basis of $\ker A$, the null-space parametrization is $y = A^{+}x + Np$ with $p \in \mathbb{R}^k$ and $k = n - \operatorname{rank} A$, and $c_x : \mathbb{R}^k \to \mathbb{R}^r$ collects the inequality and box rows expressed in these coordinates. The exact feasible set $\mathcal{C}_x = \{p : c_x(p) \le 0\}$ is contained in the set that the strict rule accepts. We write a for a strictly feasible anchor, q for the proposal and $\phi_i(t) = c_{x,i}(a + t(q - a))$ for row i restricted to the ray.

C.1 THE NULL-SPACE PARAMETRIZATION REMOVES THE EQUALITY CONSTRAINTS

Lemma C.1 (Exactness of the null-space parametrization)**.** *For every $p \in \mathbb{R}^k$ the point $y = A^{+}x + Np$ satisfies $Ay = x$, and $p \mapsto y$ is an affine bijection onto $\{y : Ay = x\}$.*

Proof. $AN = 0$ because the columns of N span $\ker A$, so $Ay = AA^{+}x$. The right-hand sides of these families lie in $\operatorname{range}(A)$, on which AA^{+} is the identity, hence $Ay = x$. The map is injective because N has full column rank, and surjective because any two solutions of $Ay = x$ differ by an element of $\ker A = \operatorname{range}(N)$. □

Lemma C.1 leaves only rounding error in the equality rows, and most returned points of RRA have a normalized residual near 10^{-10} (Figure 2). The basis is computed once and stored with the network, because null-space bases computed on different devices can differ by an orthogonal rotation.

C.2 Feasibility of the radial retraction

Proposition 1 (Feasibility for any strictly feasible anchor, restated). *Let c_x be continuous and $c_x(a) < 0$, and let $t_i^\star = \inf\{t > 0 : \phi_i(t) \geq 0\} \in (0, \infty]$ and $t^\star = \min_i t_i^\star$. For every $q \in \mathbb{R}^k$ the point $p = a + \min(1, t^\star)(q - a)$ satisfies $c_x(p) \leq 0$, and so does every point of the segment from a to p.*

Proof. Fix a row i and let $\bar{t} = \min(1, t^\star) \leq t_i^\star$. For $t \in [0, \bar{t})$ we have $t < t_i^\star$, so t does not belong to the set whose infimum defines $t_i^\star$, and $\phi_i(t) < 0$. By continuity $\phi_i(\bar{t}) = \lim_{t \uparrow \bar{t}} \phi_i(t) \leq 0$ when $\bar{t} > 0$, and $\phi_i(0) < 0$ by hypothesis. Every row is nonpositive on $[0, \bar{t}]$, which is the claim. □

The proof uses no property of a beyond $c_x(a) < 0$ and no property of q. Consequently, the conclusion survives an interior map that stopped early, was given a loose tolerance or did not converge. An untrained or adversarial proposal changes the objective of the output but not its feasibility. When the interior map returns a point that is not strictly feasible, the solver takes the global anchor, which satisfies the hypothesis. An implementation inherits the proposition to the accuracy with which it computes $t^\star$, which Lemmas C.2 and C.3 address.

C.3 Exit parameters in closed form

Lemma C.2 (Closed-form first exit)**.** *Along the ray write $y(t) = y_a + t\,w$ with $y_a = A^+x + Na$ and $w = N(q - a)$, and suppose that the row is strictly satisfied at the anchor. Each of the following rows reduces to a scalar quadratic $\psi_i(t) = \alpha t^2 + \beta t + \gamma$ with $\gamma < 0$ whose first positive root is $t_i^\star$, and this root is*

$$t_i^\star = \begin{cases} (\sqrt{\Delta} - \beta)/(2\alpha), & \alpha > 0 \text{ and } \beta < 0, \\ -2\gamma/(\beta + \sqrt{\Delta}), & \beta + \sqrt{\Delta} > 0 \text{ and } \Delta \geq 0, \\ \infty, & \text{otherwise}, \end{cases} \qquad \Delta = \beta^2 - 4\alpha\gamma. \tag{5}$$

1. Affine row $G_i^\top y - h_i$ *and* box row $\pm y_j - 5$*: $\alpha = 0$, $\beta = G_i^\top w$, $\gamma = G_i^\top y_a - h_i$.*
2. Convex-quadratic row $\frac{1}{2} y^\top H_i y + G_i^\top y - h_i$ *with $H_i \succeq 0$: $\alpha = \frac{1}{2} w^\top H_i w \geq 0$, $\beta = w^\top H_i y_a + G_i^\top w$, $\gamma = \phi_i(0)$.*
3. Cone row $\|G_i y + h_i\| - (c_i^\top y + d_i)$*: with $u_0 = G_i y_a + h_i$, $v = G_i w$, $\pi_0 = c_i^\top y_a + d_i$ and $\kappa = c_i^\top w$, take $\alpha = \|v\|^2 - \kappa^2$, $\beta = 2(u_0^\top v - \pi_0 \kappa)$ and $\gamma = \|u_0\|^2 - \pi_0^2$.*

Proof. In cases 1 and 2, substituting $y(t)$ into the row gives a polynomial of degree at most two in t with the listed coefficients, and $\gamma = \phi_i(0) < 0$ because the anchor is strictly feasible. Hence $\phi_i = \psi_i$, and the first sign change of ϕ_i is the first positive root of ψ_i.

Case 3 needs an argument because ψ_i is the squared form. Write $\pi(t) = \pi_0 + t\kappa$ for the affine part, so that $\phi_i(t) = \|u_0 + tv\| - \pi(t)$ and $\psi_i(t) = \|u_0 + tv\|^2 - \pi(t)^2$. Strict feasibility at the anchor gives $\pi_0 > \|u_0\| \geq 0$. Wherever $\pi(t) \geq 0$, both sides of $\|u_0 + tv\| \geq \pi(t)$ are nonnegative, squaring is an equivalence, and $\phi_i(t)$ and $\psi_i(t)$ have the same sign. It remains to show that the first positive root of ψ_i occurs while π is still nonnegative. If $\kappa \geq 0$, then $\pi(t) \geq \pi_0 > 0$ for all $t \geq 0$. If $\kappa < 0$, let $t_\pi = \pi_0/|\kappa| > 0$ be the zero of π. Then $\psi_i(0) = \gamma < 0$ and $\psi_i(t_\pi) = \|u_0 + t_\pi v\|^2 \geq 0$, so ψ_i has a root in $(0, t_\pi]$ by the intermediate value theorem. Before that first root, $\pi > 0$ and $\psi_i < 0$ imply $\phi_i < 0$, and at the root $\phi_i = 0$. In both cases the first positive root of ψ_i is $t_i^\star$.

For (5), suppose first that $\alpha > 0$. Then $\gamma < 0$ gives $\Delta > \beta^2 \geq 0$, and the two real roots have the product $\gamma/\alpha < 0$. Hence exactly one root is positive, and it equals $(\sqrt{\Delta} - \beta)/(2\alpha)$. Multiplying numerator and denominator by $\beta + \sqrt{\Delta} > 0$ turns it into $-2\gamma/(\beta + \sqrt{\Delta})$. The two forms agree in exact arithmetic and cancel in opposite regimes, so (5) takes the first when $\beta < 0$ and the second otherwise. If $\alpha = 0$, the row is affine, $\sqrt{\Delta} = |\beta|$, and the second form returns $-\gamma/\beta$ for $\beta > 0$ and ∞ for $\beta \leq 0$, which is correct because a row whose value decreases along the ray is never reached. If $\alpha < 0$, which occurs only for cone rows with $|\kappa| > \|v\|$, then ψ_i is concave with $\psi_i(0) < 0$, and it has a positive root only when $\Delta \geq 0$ and $\beta > 0$. Writing $\alpha = -|\alpha|$ then shows that the smaller root

$(\beta - \sqrt{\Delta})/(2|\alpha|)$ equals $-2\gamma/(\beta + \sqrt{\Delta})$, which is the form that the case condition selects. When $\Delta < 0$ or $\beta \le 0$ the row is never reached, and (5) returns ∞. □

A row that is already active at the anchor within rounding, $\gamma \ge -10^{-10}$, is handled separately, and the ray crosses it again at $-\beta/\alpha$ when $\alpha > 0 > \beta$.

C.4 The bracketing scan

The trigonometric rows have no closed-form first root. Along the ray, a row of the nonconvex QP family is $\phi(t) = \sum_j G_{ij} \sin(y_{a,j} + t w_j) - h_i \cos(x_i)$, and a row of the nonconvex QCQP family is $\phi(t) = \frac{1}{2} y(t)^\top H_i y(t) + \sum_j G_{ij} \cos(y_{a,j} + t w_j) - h_i$. Both have a second derivative bounded along the ray by

$$M_i = \sum_j |G_{ij}|\, w_j^2 + \left| w^\top H_i w \right|. \tag{6}$$

The second term is present only for QCQP rows. The bound holds because $\frac{d^2}{dt^2} \sin(y_{a,j} + t w_j) = -w_j^2 \sin(y_{a,j} + t w_j)$, with the same identity for the cosine, and because the quadratic part has the constant second derivative $w^\top H_i w$.

Lemma C.3 (Bracketing). *Let $|\phi''| \le M_i$ along the ray and let $[t_0, t_1]$ be a cell of length $h = t_1 - t_0$. If*

$$\min\left\{ \max\left(\phi(t_0),\, \phi(t_0) + \phi'(t_0)h + \tfrac{1}{2} M_i h^2\right),\, \max\left(\phi(t_1),\, \phi(t_1) - \phi'(t_1)h + \tfrac{1}{2} M_i h^2\right) \right\} \le 0, \tag{7}$$

then $\phi(t) \le 0$ for every $t \in [t_0, t_1]$.

Proof. Taylor's theorem with the Lagrange remainder at t_0 gives, for every t in the cell, some ξ between t_0 and t with $\phi(t) = \phi(t_0) + \phi'(t_0)(t - t_0) + \frac{1}{2}\phi''(\xi)(t - t_0)^2 \le u(t)$, where $u(t) = \phi(t_0) + \phi'(t_0)(t - t_0) + \frac{1}{2} M_i (t - t_0)^2$. The bound u is convex in t, so its maximum over the cell is attained at an endpoint and equals the first maximum in (7). The expansion at t_1 gives the second maximum in the same way. Each maximum bounds ϕ on the whole cell, so the smaller one does as well. □

The scan evaluates the scanned rows on a grid of 32 cells over $[0, \min(1, t^\star_{\mathrm{cf}})]$, where $t^\star_{\mathrm{cf}}$ is the closed-form exit of the remaining rows. It takes the first grid point with a positive row, bisects the cell before it for 30 steps while keeping the lower end nonpositive, and checks (7) on every earlier cell and on a sub-grid of 64 cells between the last grid point and the lower end. An instance whose check fails is scanned again on a four times finer grid, at most twice. The retraction returns the lower end of the final bracket. Any remaining error of the scan then places the returned point inside $\mathcal{C}_x$, because this end lies below the first crossing. On the eight families without nonconvex cone rows, every output of every reported seed leaves through a closed-form exit or through a scanned exit whose checks passed.

The cone rows of the two nonconvex SOC families contain $\cos(y)$ inside the norm, and the scan runs on them without the check (7). It still returns the lower end of a bracket at which every row has been evaluated and found nonpositive. Hence every output on these families satisfies $c_x \le 0$, and the evaluator finds that all of them pass the strict rule. Because the check is absent, the scan does not exclude a crossing between grid points before the returned point, and the segment statement of Proposition 1 is not established for these rows.

C.5 Differentiability

Lemma C.4 (Where the retraction is differentiable). *Suppose that at (a, q) the minimum defining $t^\star$ is attained by a single row i, that this row is continuously differentiable near the contact point, that the crossing is transversal, $\phi_i'(t^\star) > 0$, and that $t^\star < 1$. Then the map $(a, q) \mapsto \mathcal{R}(x, a, q)$ is continuously differentiable in a neighborhood of (a, q), with*

$$\nabla_{(a,q)}\, t^\star = -\frac{\nabla_{(a,q)}\, \phi_i(t^\star)}{\phi_i'(t^\star)}. \tag{8}$$

Proof. Let $F(a, q, t) = c_{x,i}(a + t(q - a))$, which is C^1 in all arguments by hypothesis. Affine, quadratic and box rows are C^1 everywhere, and a cone row is C^1 wherever the vector inside its norm is nonzero. At the point in question $F = 0$ and $\partial F/\partial t = \phi_i'(t^\star) > 0$, so the implicit function theorem gives a C^1 function $t^\star(a, q)$ near (a, q) with the stated gradient. Because the minimizing row is unique, $t^\star$ coincides with this branch on a neighborhood, and $\min(1, t^\star) = t^\star$ there since $t^\star < 1$. Composition with the affine map $(a, q, t) \mapsto a + t(q - a)$ preserves differentiability. □

The hypotheses fail at a tie between two rows, at a tangential crossing, at the vertex of a cone row and at $t^\star = 1$, where the retraction switches between returning q and returning the exit point. On each of these sets the solver differentiates the branch it selected, as a rectifier or a maximum does where its pieces meet. Its derivative comes from the closed-form root of (5) or from the implicit derivative above, not from the bisection steps.

C.6 Sliding stages and several anchors

Lemma C.5 (Sliding stages and the choice among anchors keep the rows bounded)**.** *Let $c_x(p_0) \le 0$, and let $p_1, \dots, p_L$ be produced by sliding stages, each accepted only if a check confirms $c_x(p_\ell) \le \delta$ for a fixed $\delta \ge 0$, with the previous point retained otherwise. Then every retained point satisfies $c_x \le \delta$, and so does the best of the points returned from several anchors.*

Proof. The base point satisfies $c_x \le 0 \le \delta$ by Proposition 1. At stage ℓ the retained point is either p_ℓ, which was checked before acceptance, or the retained point of stage $\ell - 1$, which satisfies the bound by induction. Selecting among points that satisfy the bound returns such a point. □

Consequently, a stage may use any heuristic without putting the bound at risk. Each stage used here takes a radial step along the tangent space of the rows that stopped the ray and then applies Newton corrections on curved rows. A stage is accepted at $\delta = 10^{-10}$ in the units of the rows, well inside the tolerances of the strict rule. This check applies to one point and does not control the length of an iteration. Sliding stages run only on the two nonconvex SOC families.

C.7 Differentiating the interior map

Proposition 2 (Derivative of one step of the interior map, restated). *Fix an open set on which the selected rows S are constant, $c_{x,S}$ is C^2 and $J_S(p) = \partial c_{x,S}/\partial p$ has full row rank, and let $T_\rho(p) = p - J_S^\top (J_S J_S^\top + \rho I)^{-1} r(p)$ with $r(p) = c_{x,S}(p) + \mu \mathbf{1}$ be one Gauss–Newton step. At any p with $r(p) = 0$,*

$$\frac{\partial T_\rho}{\partial p}(p) = I - J_S^\top (J_S J_S^\top + \rho I)^{-1} J_S \xrightarrow[\rho \to 0]{} I - J_S^\top (J_S J_S^\top)^{-1} J_S,$$

the orthogonal projector onto $\ker J_S$*, the tangent space of the face $\{c_{x,S} = -\mu\}$. At a general p the Jacobian differs from the first matrix by $O(\|r(p)\|)$.*

Proof. Write $M(p) = J_S(p)^\top (J_S(p) J_S(p)^\top + \rho I)^{-1}$, so that $T_\rho(p) = p - M(p) r(p)$ and, by the product rule,

$$\frac{\partial T_\rho}{\partial p} = I - M(p)\frac{\partial r}{\partial p} - \Big(\frac{\partial M}{\partial p}\Big)[\,\cdot\,, r(p)] = I - M(p) J_S(p) - \Big(\frac{\partial M}{\partial p}\Big)[\,\cdot\,, r(p)], \tag{9}$$

where the last term is the third-order tensor $\partial M/\partial p$ contracted with $r(p)$ in its second slot. Each entry of M is a rational function of the entries of J_S whose denominator is bounded away from zero by ρ, and J_S is C^1 on the set considered, so $\partial M/\partial p$ is bounded there by a constant C that depends on $\sup \|\partial J_S\|$ and on ρ. Hence the last term of (9) has norm at most $C\|r(p)\|$ and vanishes when $r(p) = 0$, which proves the first display and the final claim. As $\rho \to 0$, $J_S^\top (J_S J_S^\top + \rho I)^{-1} J_S \to J_S^\top (J_S J_S^\top)^{-1} J_S$ because $J_S J_S^\top$ is invertible, and $I - J_S^\top (J_S J_S^\top)^{-1} J_S$ is symmetric, idempotent, annihilates $\operatorname{range}(J_S^\top)$ and fixes $\ker J_S$, so it is the orthogonal projector onto $\ker J_S$. The face $\{c_{x,S} = -\mu\}$ has tangent space $\ker J_S$ at each of its points. □

C.8 THE DERIVATIVE OF THE SOLVER

The interior map applies D steps of Eq. (1), each with its own selected rows, and its Jacobian at the proposal is the product $J_{\mathcal{I}} = \partial T_{D-1} \cdots \partial T_0$ of the step Jacobians of Proposition 2. The retraction then differentiates the exit it found.

Lemma C.6 (Derivative of the composed solver)**.** *Let $\mathcal{I}$ be differentiable at q with Jacobian $J_{\mathcal{I}}$, let $a = \mathcal{I}(q)$ satisfy $c_x(a) < 0$, and let the hypotheses of Lemma C.4 hold at (a, q) with stopping row i. Write $p = a + t^\star(q - a)$, $d = q - a$, $n = \nabla c_{x,i}(p)$ and $\eta = n^\top d > 0$. Then $q \mapsto p$ is differentiable at q with*

$$\frac{\partial p}{\partial q} = \Big(I - \frac{d\, n^\top}{\eta}\Big)\big[\, t^\star I + (1 - t^\star) J_{\mathcal{I}}\,\big]. \tag{10}$$

Proof. Differentiating $p = a + t^\star(q - a)$ with $a = \mathcal{I}(q)$ and collecting the terms in $\partial a/\partial q = J_{\mathcal{I}}$ gives

$$\frac{\partial p}{\partial q} = t^\star I + (1 - t^\star) J_{\mathcal{I}} + d\,(\nabla_q t^\star)^\top. \tag{11}$$

Lemma C.4 keeps row i the unique stopping row in a neighborhood of q, so $c_{x,i}(p) = 0$ there and $n^\top \partial p/\partial q = 0$. Multiplying (11) by $n^\top$ and using $n^\top d = \eta > 0$ gives $(\nabla_q t^\star)^\top = -\eta^{-1} n^\top [t^\star I + (1 - t^\star) J_{\mathcal{I}}]$, and substituting it back into (11) gives (10). □

The first factor of (10) is an oblique projection onto the tangent plane of the stopping row along the direction of the ray, and it agrees with the orthogonal projection only when the ray meets the boundary along its normal. The derivative of the interior map acts in the second factor, and Proposition 2 gives it for a converged Gauss–Newton map. Section 4.3 varies this factor.

D REFERENCE SOLUTIONS

D.1 BEST-KNOWN REFERENCES

The candidates on every instance of the four nonconvex QCQP and SOCP families are the points that pass the strict rule among the benchmark reference, the seed-2025 outputs of RRA with four anchors and sliding stages, and the seed-2025 outputs of FSNet, from both implementations on the two QCQP families (Appendix E.3). SLSQP (at most 200 iterations, function tolerance 10^{-12}) is warm-started at each candidate, a polished point is kept only if it passes the strict rule, and the best objective is recorded with its source. Table 5 shows how far the references moved and which source supplied the best point. The two nonconvex QP families keep the benchmark reference, and on the nonconvex QP instances of Appendix D.2 the incumbents of Gurobi agree with it. The gap difference between two methods on a common instance is $(f_x(y_A) - f_x(y_B))/\max(|f_x^\star|, 1)$, which depends on the reference only through the normalization.

Table 5: Instances on which a polished candidate improves on the benchmark reference, mean objective of the benchmark and the best-known references, and the source of the best point.

		mean objective		source of the best point		
Family	improved of 2000	benchmark	best-known	RRA	FSNet	benchmark
nonconvex QCQP	952	−17.789	−18.090	609	346	1045
nonconvex QCQP $+\|y\|$	1019	−16.375	−16.990	719	310	971
nonconvex SOCP	1943	−0.620	−5.720	1139	809	52
nonconvex SOCP $+\|y\|$	1690	−2.431	−4.488	1395	301	304

D.2 GLOBAL BOUNDS WITH GUROBI

Gurobi 12 solves nonconvex QP instances by spatial branch-and-bound, with the trigonometric terms modeled by its general function constraints. After 600 s per instance the MIP gap remains between 98% and 113%, and every incumbent agrees with the benchmark reference to within 10^{-6} (Table 6).

Table 6: Gurobi spatial branch-and-bound on nonconvex QP instances, with the objective of the benchmark reference.

time limit (s)	instance	lower bound	incumbent	benchmark	MIP gap (%)	proven optimal
45	3714	−41.555	−19.485	−19.485	113.3	no
45	4507	−41.289	−19.771	−19.771	108.8	no
45	7591	−41.668	−19.629	−19.629	112.3	no
600	1940	−38.657	−19.247	−19.247	100.8	no
600	3158	−38.628	−19.339	−19.339	99.7	no
600	3294	−39.018	−18.307	−18.307	113.1	no
600	108	−39.255	−19.816	−19.816	98.1	no

E IMPLEMENTATION

E.1 RRA SETTINGS

All learned methods use a multilayer perceptron with four hidden layers of width 1024, with SiLU activations for RRA and ENFORCE, the defaults of the FSNet code for FSNet and DC3, and the settings of Appendix E.5 for ΠInet. RRA is trained with AdamW at learning rate 5×10^{-4} and weight decay 10^{-3}, batch size 512 (256 on convex SOCP and 128 on the two nonconvex SOC families), 100 epochs (40 on the three SOC families), a cosine learning-rate schedule and gradient-norm clipping at 1.0. We keep the checkpoint with the best validation objective among evaluations every ten epochs. The interior map uses depth D=20 (10 on the two nonconvex SOC families), at most 16 selected rows and margin μ=10^{-4}. Its linear solve adds the ridge $\rho\,\sigma(p)I$ with $\rho = 10^{-10}$ and $\sigma(p)$ the mean diagonal of $J_S J_S^\top$ clamped below at one. An output of the map is accepted as an anchor when every row is at most -10^{-9}, and the search behind the global anchor accepts a candidate whose smallest slack, relative to the scale of its row, is above 10^{-6}. Training uses the proximity weight λ=1 and the proposal scale s=5. Global anchors come from one start, and the solver takes one ray step, a single retraction along the ray toward the proposal. The two nonconvex SOC families instead use the four anchors and the sliding stages of Section 3.2, and the computation from each anchor, called a branch below, takes a second ray step with its retracted point as the new anchor. There the objective term of Eq. (3) is the smallest objective among the branches that return one, and the proximity term is the mean of the four branch terms. The settings of the interior map, the retraction and the proximity weight come from the variants of Appendix G.1 at seed 2025 and are shared by every family except the two nonconvex SOC families. Seeds 2026 and 2027 use these settings unchanged.

E.2 HARDWARE AND SOFTWARE

All experiments ran on one machine with an NVIDIA L40S GPU (48 GB), an AMD EPYC 7R13 processor with 16 virtual cores and 128 GB of memory. RRA, FSNet, DC3, ENFORCE and the penalty baselines use Python 3.13 and PyTorch 2.13 with CUDA 13.0, and ΠInet uses the authors' JAX package. The convex references use Clarabel, the best-known references use SLSQP from SciPy, and the global bounds use Gurobi 12. One other long-running job shared the GPU throughout, and every training time in the paper was measured under this same load. The inference times of Figure 5(b) were measured back to back on the same GPU with two CPU threads, and each is the best of five repeats. FSNet ran with batch 2000 on every family and RRA with the same batch except on the three SOCP families, where it ran with batch 500. On the two nonconvex SOC families the multi-start anchor search accounts for 46.9 to 51.8 s of the best repeat, and the network, the interior map and the retraction with sliding stages of the four branches account for the rest. Table 16 also lists single-pass times on the shared GPU.

E.3 FSNET

FSNet runs for 500 epochs, the length its authors report, in our implementation. The authors' public code, run on the same split, gives the same gap on every family where both were run except two (Table 7). The two implementations are close on nonconvex QP. On nonconvex QCQP the authors'

code reaches 1.61% on seed 2025 against 12.1% for our implementation, and both values lie within the seed range of our implementation (Table 12) and above the gap of RRA.

Table 7: Optimality gap (%) of FSNet in our implementation and in the authors' code, seed 2025. ° gap on the outputs that pass the strict rule, – not run, † best-known references.

Family	500 epochs	500 epochs, authors' code	100 epochs
convex QP	0.031	0.031	0.091
convex QCQP	0.092	0.092	0.303
convex SOCP	0.554	–	–
convex QP, n=200	0.077°	–	–
nonconvex QP	0.145	0.162	0.259
nonconvex QP $+\|y\|$	0.143°	0.143°	–
nonconvex QCQP†	12.1°	1.61	12.5
nonconvex QCQP $+\|y\|$†	1.04	1.04	–
nonconvex SOCP†	16.4°	–	–
nonconvex SOCP $+\|y\|$†	28.8°	–	–

E.4 ENFORCE

The public code of ENFORCE (Lastrucci & Schweidtmann, 2025) targets function fitting, and we implement the method for the parametric programs of this benchmark, with the same instances and the same evaluator as every other method. The backbone predicts an extended output (y, s) with one slack per inequality row, and each row $g_i(y) \le 0$ becomes the Fischer–Burmeister equation $\varphi(s_i, -g_i(y)) = 0$ with $\varphi(a, b) = a + b - \sqrt{a^2 + b^2}$ (Fischer, 1992). The adaptive projection repeats a Gauss–Newton correction on the stacked residual $[Ay - x; \varphi]$ until $\|\cdot\|_\infty < \varepsilon$ or a depth cap is reached, and training differentiates through it. Box bounds are ordinary rows, and the set the method enforces is exactly the set the evaluator checks.

The Gauss–Newton step is damped by a backtracking search over $\{1, 0.5, 0.25, 0.1\}$ that accepts the first decrease of the residual, because the undamped step diverges on convex QP. Slacks are initialized with a rectifier rather than a softplus, because a strictly positive slack forces its row onto the boundary. The tolerance is $\varepsilon = 10^{-10}$ with depth 12, which meets the box tolerance of 10^{-9} on every family where training converges. ENFORCE trains for 100 epochs, as RRA does outside the SOC families. Objective-only training is unstable on convex QP (Table 8). Adding the distance term of the FSNet baseline (weight 5) repairs it, and the convex families use this term. On the two nonconvex SOC families training diverges, and the returned points violate the inequality and box constraints far beyond the tolerances. The same settings make ENFORCE the most accurate baseline on the other nonconvex families.

Table 8: ENFORCE with and without the distance term in the training objective, seed 2025. × fewer than half of the outputs pass the strict rule, – not run, † best-known references.

	objective only		with distance term	
Family	strict (%)	gap (%)	strict (%)	gap (%)
convex QP	100	41.7	100	0.0030
convex QCQP	–	–	100	0.0070
convex SOCP	–	–	100	0.0036
nonconvex QP	100	0.132	–	–
nonconvex QP $+\|y\|$	100	0.083	–	–
nonconvex QCQP†	100	0.234	–	–
nonconvex QCQP $+\|y\|$†	100	−0.105	–	–
nonconvex SOCP†	0.0	×	–	–
nonconvex SOCP $+\|y\|$†	0.05	×	–	–

E.5 ΠNET

We use the authors' JAX package and the constraint objects of their QP benchmark, an equality constraint with a right-hand side per instance, an affine inequality constraint and a box constraint, composed by the package's projection. The convex SOCP and QCQP rows use the package's nonlinear form $g(Ay + a) \le f^\top y + b$. The cone row $\|G_i y + h_i\| \le c_i^\top y + d_i$ maps to it directly, and the quadratic row $\frac{1}{2} y^\top H_i y + G_i^\top y \le h_i$ maps to the rotated cone $\|(2L_i y,\, 2v_i - 1)\| \le 2v_i + 1$ with $v_i = h_i - G_i^\top y$ and $H_i = L_i^\top L_i$. All reported runs use double precision and 1000 projection iterations at test time. Double precision is needed because the box tolerance of 10^{-9} cannot be met in the single precision that JAX uses by default. Table 9 compares the authors' defaults, their tuned step size and equilibration, a backbone of the size RRA uses and the combination of the last two. The backbone of the size RRA uses gives the lowest gap. ΠNet projects onto convex sets and does not apply to the six nonconvex families.

Table 9: ΠNet settings on the convex families, seed 2025. $\times$ fewer than half of the outputs pass the strict rule.

Family	setting	strict (%)	gap (%)
convex QP	authors' defaults	100	0.351
	authors' tuned σ and equilibration	100	0.214
	backbone of the size RRA uses	100	0.110
	tuned, with that backbone	100	0.111
	single precision	0.0	$\times$
convex QCQP	backbone of the size RRA uses	100	2.42
convex SOCP	backbone of the size RRA uses	100	0.092
convex QP, n=200	backbone of the size RRA uses	100	0.726

E.6 OTHER BASELINES

DC3 and the penalty and adaptive-penalty baselines use the settings of the FSNet code. HardNet-Aff (Min & Azizan, 2024) uses its closed-form affine layer, and the OptNet projection (Amos & Kolter, 2017) solves the exact projection as a differentiable QP layer and trains for 500 epochs. Both methods apply only to affine constraints and run on convex QP. The supervised baseline trains the network on reference solutions and projects its output with SLSQP at test time. Every baseline was run on every family its method supports, with two exceptions. ENFORCE and DC3 were not run on convex QP with n=200. The supervised baseline was run only on convex QP, convex QCQP, nonconvex QP and nonconvex QCQP, because its SLSQP projection takes up to 2.9 h per 2000 test instances (Table 16).

F RESULTS BY METRIC

F.1 STRICT FEASIBILITY RATE PER SEED

Table 10 gives the rate of each iterative baseline for every training seed, and it leaves out RRA, which passes the strict rule on every family and seed. The OptNet and SLSQP projections pass the strict rule on every family and seed they were run on. HardNet-Aff reaches 100, 99.9 and 100% on the three seeds of convex QP, and penalty and adaptive-penalty training return no output that passes the strict rule on any family or seed.

Table 10: Strict feasibility rate (%) of the iterative baselines per training seed. / not applicable, – not run, † best-known references.

	convex				nonconvex, + adds $0.1\|\|y\|\|_2$					
Seed	QP	QCQP	SOCP	QP, n=200	QP	QP+	QCQP†	QCQP+†	SOCP†	SOCP+†
FSNet										
2025	100	100	100	89.3	100	99.9	99.9	100	99.7	99.7
2026	100	100	100	90.1	100	100	99.9	100	99.9	99.9
2027	100	100	100	87.8	100	100	100	100	99.9	100
ENFORCE										
2025	100	100	100	–	100	100	100	100	0.0	0.05
2026	100	100	100	–	100	100	100	100	0.0	0.0
2027	100	100	100	–	100	100	100	100	0.05	0.05
Π*net*										
2025	100	100	100	100	/	/	/	/	/	/
2026	100	95.8	100	100	/	/	/	/	/	/
2027	100	99.5	100	100	/	/	/	/	/	/
DC3										
2025	59.8	61.9	0.0	–	92.0	75.3	81.7	46.0	0.0	0.0
2026	37.1	78.6	0.0	–	90.0	74.7	93.3	82.0	0.0	0.0
2027	39.4	25.2	0.0	–	91.9	71.6	53.9	39.7	0.0	0.0

F.2 One network under several stopping rules

Table 11 gives the strict feasibility rate and the gap behind Figure 1(b,c). The Πnet network is the one trained with seed 2025 on convex QP. The ENFORCE network was trained on nonconvex QP with tolerance 10^{-10} and depth 12 and is separate from the three seeds of Table 2. Only the stopping rule at test time changes along each row, and the gap stays the same once every output passes the strict rule.

Table 11: One trained network each of Πnet and ENFORCE as the stopping rule at test time varies. × fewer than half of the outputs pass the strict rule, so no gap is reported.

Π*net on convex QP, iterations at test time*									
	10	25	50	100	200	500	1000	2000	5000
strict (%)	0.0	0.0	0.0	0.0	100	100	100	100	100
gap (%)	×	×	×	×	0.111	0.111	0.111	0.111	0.111
ENFORCE on nonconvex QP, stopping tolerance ε									
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
strict (%)	9.7	42.4	100	100	100	100	100	100	100
gap (%)	×	×	0.218	0.218	0.218	0.218	0.218	0.218	0.218

F.3 Optimality gap per seed

Tables 12 and 13 list the gap of every method for each training seed together with its standard deviation, and Table 14 adds five further RRA seeds on three convex families.

Table 12: Optimality gap (%) per training seed and its standard deviation over the seeds 2025–2027. $^\circ$ gap on the outputs that pass the strict rule, for a seed below 100% (Table 10), $\times$ a seed on which fewer than half of the outputs pass the rule, left out of the standard deviation, / not applicable, – not run, † best-known references.

	convex				nonconvex, + adds $0.1\|y\|_2$					
Seed	QP	QCQP	SOCP	QP, n=200	QP	QP+	QCQP†	QCQP+†	SOCP†	SOCP+†
RRA (ours)										
2025	0.0022	0.0063	0.0060	0.0069	0.127	0.090	0.375	−0.0058	11.0	11.9
2026	0.0019	0.0060	0.0053	0.0068	0.129	0.091	0.364	−0.0044	8.65	12.4
2027	0.0020	0.0055	0.0051	0.0069	0.128	0.091	0.368	−0.0058	6.23	18.3
std	0.0001	0.0004	0.0005	0.0001	0.0012	0.0004	0.0055	0.0008	2.37	3.52
RRA, longer training										
2025	0.0015	0.0033	0.0025	–	0.103	0.071	0.248	−0.134	–	–
2026	0.0015	0.0033	0.0024	–	0.102	0.072	0.248	−0.134	–	–
2027	0.0015	0.0036	0.0027	–	0.105	0.071	0.249	−0.134	–	–
std	<0.0001	0.0002	0.0001	–	0.0016	0.0004	0.0008	0.0003	–	–
FSNet										
2025	0.031	0.092	0.554	0.077°	0.145	0.143°	12.1°	1.04	16.4°	28.8°
2026	0.037	0.078	0.860	0.078°	0.147	0.116	1.66°	−0.054	47.9°	116.9°
2027	0.040	0.069	0.775	0.053°	0.144	0.116	0.276	1.06	10.1°	24.4
std	0.0046	0.012	0.158	0.014	0.0013	0.015	6.44	0.637	20.3	52.2
ENFORCE										
2025	0.0030	0.0070	0.0036	–	0.132	0.083	0.234	−0.105	×	×
2026	0.0013	0.0047	0.0037	–	0.104	0.088	0.240	−0.193	×	×
2027	0.0011	0.0043	0.0097	–	0.120	0.077	1.08	0.345	×	×
std	0.0010	0.0014	0.0035	–	0.014	0.0056	0.487	0.289	–	–
Π*net*										
2025	0.110	2.42	0.092	0.726	/	/	/	/	/	/
2026	0.036	6.07°	0.406	0.558	/	/	/	/	/	/
2027	0.0081	0.882°	0.400	0.367	/	/	/	/	/	/
std	0.052	2.66	0.180	0.179	/	/	/	/	/	/
DC3										
2025	4.08°	3.71°	×	–	2.34°	1.96°	11.4°	×	×	×
2026	×	4.35°	×	–	2.42°	1.61°	17.8°	19.4°	×	×
2027	×	×	×	–	2.69°	1.86°	18.1°	×	×	×
std	–	0.454	–	–	0.182	0.181	3.77	–	–	–

Table 13: Optimality gap (%) per training seed of the methods run on few families.

Method	Family	2025	2026	2027	mean	std
HardNet-Aff	convex QP	0.051	0.165°	0.152	0.123	0.063
OptNet projection	convex QP	0.0028	0.050	0.046	0.033	0.026
supervised + SLSQP	convex QP	0.060	0.075	0.083	0.073	0.011
	convex QCQP	0.243	0.091	0.323	0.219	0.118
	nonconvex QP	0.232	0.246	0.272	0.250	0.020
	nonconvex QCQP	6.36	6.12	6.27	6.25	0.120

Table 14: Optimality gap (%) of RRA for five further seeds on three convex families, every output feasible under the strict rule. Mean and standard deviation are over all eight seeds.

Family	2028	2029	2030	2031	2032	mean	std
convex QP	0.0021	0.0020	0.0019	0.0018	0.0019	0.0020	0.0001
convex QCQP	0.0060	0.0055	0.0065	0.0058	0.0067	0.0060	0.0005
convex QP, n=200	0.0069	0.0070	0.0069	0.0068	0.0070	0.0069	0.0001

F.4 Training and inference time

Table 15: Training time per run (hours), mean over the seeds 2025–2027. / not applicable, – not run.

	convex				nonconvex, + adds $0.1\|y\|_2$					
Method	QP	QCQP	SOCP	QP, n=200	QP	QP+	QCQP†	QCQP+†	SOCP†	SOCP+†
RRA (ours)	0.28	0.29	0.58	0.59	0.34	0.28	0.30	0.27	2.6	2.3
RRA, longer training	1.2	0.90	1.5	–	1.5	1.7	1.0	1.0	–	–
FSNet	1.8	2.3	2.1	7.8	1.7	1.9	2.5	2.7	3.4	2.8
ENFORCE	3.2	3.1	2.7	–	3.7	3.9	3.3	3.7	4.3	4.0
ΠNet	0.18	10.4	1.4	0.43	/	/	/	/	/	/
DC3	0.17	0.18	0.19	–	0.17	0.20	0.19	0.22	0.20	0.21
HardNet-Aff	0.04	–	–	–	–	–	–	–	–	–
OptNet projection	16.8	–	–	–	–	–	–	–	–	–
supervised + SLSQP	0.08	0.09	–	–	0.05	–	0.05	–	–	–
penalty	0.06	0.07	–	–	0.07	–	0.07	–	–	–
adaptive penalty	0.06	0.07	–	–	0.06	–	0.07	–	–	–

Table 16: Inference time for the 2000 test instances (seconds), back to back as the best of five repeats and in a single pass on the shared GPU. / not applicable, – not run.

	convex				nonconvex, + adds $0.1\|y\|_2$					
Method	QP	QCQP	SOCP	QP, n=200	QP	QP+	QCQP†	QCQP+†	SOCP†	SOCP+†
Back to back on the same GPU, best of five repeats										
RRA (ours)	0.15	0.18	1.36	0.27	0.21	0.23	0.64	0.64	86.0	79.9
FSNet	0.20	0.76	0.45	0.39	0.20	0.20	0.90	0.88	0.68	0.70
Single pass on the shared GPU										
RRA (ours)	1.64	1.61	6.59	2.07	1.90	2.15	9.71	6.03	80.2	56.4
FSNet	0.61	0.85	1.03	3.94	0.61	2.91	0.99	1.26	3.28	1.47
ENFORCE	10.2	6.86	21.3	–	5.85	8.61	5.95	19.1	20.3	15.6
ΠNet	0.83	664	221	3.19	/	/	/	/	/	/
DC3	0.66	0.54	0.22	–	0.64	0.85	0.70	0.71	0.69	0.59
HardNet-Aff	0.06	–	–	–	–	–	–	–	–	–
OptNet projection	11.3	–	–	–	–	–	–	–	–	–
supervised + SLSQP	39.3	10476	–	–	20.9	–	99.3	–	–	–

G Ablations and geometry

G.1 Variants on nonconvex QP

Table 17 lists every RRA variant trained on nonconvex QP with the optimizer settings of RRA, grouped by interior map and retraction. Figure 6 places the same runs in the plane of gap and inference time, and RRA itself, a Gauss–Newton map of depth 20 with a single ray, lies at its lower

left. On the convex families, a retraction from a fixed anchor with sixteen sliding stages reaches several times the gap of RRA (Table 18).

Table 17: RRA variants on nonconvex QP (seed 2025, 100 epochs, every output feasible under the strict rule). Runs of one variant differ in the proximity weight λ and in whether the solver returns the better of the anchor and the retracted point. The last column gives the range of their gaps.

interior map	retraction	λ tried	runs	gap (%)
Gauss–Newton interior map				
depth 30	8 sliding stages	1	1	0.123
depth 20	single ray	0, 1	3	0.124–0.127
depth 20	2 sliding stages	1	1	0.128
depth 20	8 sliding stages	0, 1	3	0.123–0.125
depth 20	8 sliding stages, without the global anchor	1	1	0.123
depth 10	8 sliding stages	1	1	0.176
depth 10, 32 rows	8 sliding stages	1	1	0.178
depth 5	4 sliding stages	1	1	0.503
depth 5	8 sliding stages	0, 1	2	0.358–0.477
depth 5	8 sliding stages, 2 starts	0	1	0.345
Adam interior map				
30 steps, straight-through	4 sliding stages	0	1	0.746
30 steps, straight-through	8 sliding stages	0, 0.3, 1, 5	4	0.589–0.839
30 steps, tangent projection	4 sliding stages	1	1	0.621
30 steps, tangent projection	8 sliding stages	0	1	0.881
30 steps, detached	8 sliding stages	0	2	0.767–0.768
60 steps, detached	4 sliding stages	0	1	1.000
Fixed anchor, no interior map				
–	8 sliding stages	1	1	0.928
–	8 sliding stages, 2 ray steps	0	1	1.20
–	16 sliding stages	0	1	1.10

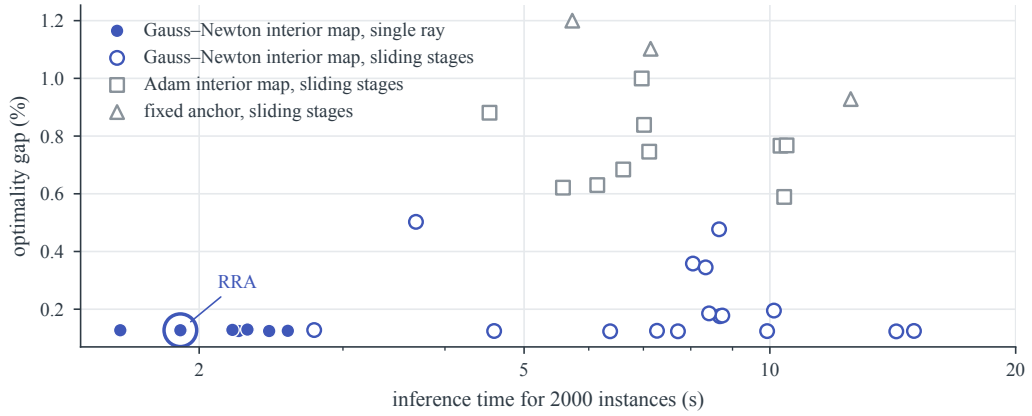


Figure 6: **Gap against inference time for every variant on nonconvex QP.** Filled markers use a single ray and hollow markers add sliding stages. The marker shape gives the interior map. Inference times are single passes on the shared GPU.

Table 18: Gap (%) of RRA and of a retraction from a fixed anchor with sixteen sliding stages on the convex families (seed 2025, 100 epochs, same optimizer settings).

Family	RRA	fixed anchor, sixteen sliding stages
convex QP	0.0022	0.0098
convex QCQP	0.0063	0.037

G.2 Proximity term

Table 19: Gap (%) with and without the proximity term, at the reported interior map and a single ray (seed 2025, 100 epochs), and the mean number of active rows at the reference optimum. † best-known references.

Family	active rows at the optimum	gap with (%)	gap without (%)	ratio
convex QP	15.6	0.0022	0.392	179
convex QCQP	16.0	0.0063	0.237	38
nonconvex QP	6.7	0.127	0.124	0.98
nonconvex QP $+\|\|y\|\|$	6.6	0.090	0.088	0.97
nonconvex QCQP†	12.5	0.375	8.00	21

G.3 Settings on the nonconvex SOC families

Table 20 lists the runs on nonconvex SOCP that differ in the anchor search and in the number of sliding stages. Four sliding stages reduce the gap substantially with a single anchor and change it little with four anchors kept from sixteen starts. The runs with four anchors also use two ray steps, an interior map of depth 10 and batch size 128.

Table 20: RRA settings on nonconvex SOCP (seed 2025, best-known references).

starts S	anchors B	stages L	ray steps K	depth D	batch	epochs	gap (%)
1	1	0	1	20	256	40	46.7
1	1	4	1	20	256	40	20.1
16	4	0	2	10	128	40	11.2
16	4	4	2	10	128	40	11.0

G.4 ACTIVE CONSTRAINTS AND ANCHOR COVERAGE

A search that uses only the constraints finds a strictly feasible anchor from a single random start on every instance of seven families. On the two nonconvex SOC families it succeeds on 13% of the instances, and eight starts do not improve on this. RRA adds a Gauss–Newton polish at the end of the anchor search on these families and keeps four of sixteen starts. This search raises their inference time two orders of magnitude above that of the other families. Table 21 also counts the rows active at the reference optimum.

Table 21: Active rows at the reference optimum (mean over the 2000 test instances) and anchor coverage of the search without the Gauss–Newton polish, on the nine families with $n = 100$.

		anchor coverage (%)	
Family	active rows at the optimum	one start	eight starts
convex QP	15.6	100	100
convex QCQP	16.0	100	100
convex SOCP	9.0	100	100
nonconvex QP	6.7	100	100
nonconvex QP $+\|y\|$	6.6	100	100
nonconvex QCQP	12.5	100	100
nonconvex QCQP $+\|y\|$	12.2	100	100
nonconvex SOCP	14.9	13	13
nonconvex SOCP $+\|y\|$	15.2	13	13