REPRESENTATION LEARNING FOR EXACT PREIMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern neural predictors can model highly nonlinear maps, but many scientific and engineering tasks require reasoning in the opposite direction: given a performance or safety level, the goal is to characterize the preimage, that is, the complete set of inputs which meet the desired target level and optimize over that set. For expressive neural predictors, however, such preimages typically have no explicit representation and are expensive to recover or optimize over. This creates a fundamental three-way challenge between expressive forward prediction, accurate preimage approximation, and tractable optimization over the preimage for downstream tasks. We introduce TRIO (**t**ractable **r**epresentations for pre**i**mage learning and inverse **o**ptimization), a framework for learning representations that make these objectives compatible by construction. Our key contribution is a *preimage factorization*: the forward model remains expressive through nonlinear radial transformations (including neural networks), while, under inversion, each transformation reduces to a single scalar radius, which yields simple geometric level sets. This yields an explicit geometric representation that is reusable for downstream optimization over the preimage, and, for linear objectives, we show that this admits a closed-form global solution. We finally prove a universal approximation theorem which shows that TRIO can approximate any continuous forward map and its entire family of potentially disconnected, nonconvex preimages arbitrarily well. Hence, TRIO combines expressive forward modeling, exact preimage recovery, and tractable global downstream optimization over preimages by design.

1 INTRODUCTION

Many problems in science and engineering require not only predicting the outcome of a given input, but instead identifying the full set of inputs that satisfy a desired performance or safety requirement. For example, in inverse design, safety, and backward reachability, the goal is to find all designs that meet a prescribed target level under a learned surrogate, and then optimize over the resulting preimage (Bhosekar & Ierapetritou, 2018; Kotha et al., 2023; Lee et al., 2023).

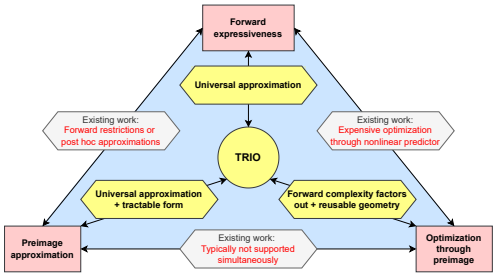


Figure 1: **Filling the gap:** TRIO combines forward expressiveness with exact preimage approximation and tractable inverse optimization.

However, this type of backward reasoning is difficult for expressive predictors: the forward map may be easy to evaluate, while its preimages are typically not available in explicit form and costly to recover or optimize over (Ceccon et al., 2022; Hoang et al., 2024; Kotha et al., 2023). This creates a *fundamental three-way challenge between expressive forward prediction, accurate preimage approximation, and tractable downstream optimization over the preimage*. Existing work has largely addressed these challenges along three separate directions: (i) recovering preimages post hoc from trained predictors (e.g., Dathathri et al., 2019; Kotha et al., 2023; Björklund et al., 2026); (ii) expensive inverse optimization through learned predictors (e.g., Bergman et al., 2022; Ceccon et al., 2022; Schweidtmann & Mitsos, 2019); and (iii) designing restricted model architectures that make inverse reasoning more tractable (e.g., Amos et al., 2017; Behrmann et al., 2019; Papamakarios et al., 2021). Yet, an approach that addresses all challenges simultaneously is still missing (see Figure 1).

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.

We introduce TRIO (**t**ractable **r**epresentations for pre**i**mage learning and inverse **o**ptimization), a framework that combines expressive forward prediction with exact preimage representation and tractable downstream optimization. For any target chosen after training, TRIO represents the complete learned preimage as an explicit finite union of ellipsoidal regions, even when the predictor is non-injective and the preimage is disconnected or nonconvex. Unlike approaches that recover preimages post hoc, repeatedly optimize over the predictor, or impose convexity or invertibility for tractability, TRIO provides an explicit preimage representation that can be reused for set queries and downstream optimization.

The key idea behind TRIO is a *preimage factorization*: TRIO uses expressive radial transformations for the forward prediction, while their complexity factors out when computing the preimage. Specifically, in the backward pass, TRIO combines radial components through a union operation, while each radial transformation collapses into a single scalar radius. The result is an exact finite geometric preimage even when the radial transformations are parametrized by neural networks. Importantly, this construction separates forward expressiveness from the complexity of preimage computation.

Our contributions are fivefold[1]: **(1)** We introduce TRIO, a framework that unifies expressive forward prediction, accurate preimage approximation, and tractable downstream optimization over the preimage. **(2)** We prove that the preimage factorization is exact: every target level yields a finite union of ellipsoidal regions, while the complexity of the forward radial models factors out of subsequent preimage reasoning. **(3)** We show that this geometry enables globally exact downstream optimization, and, for linear objectives, this even admits a closed-form global solution. **(4)** We establish a universal approximation theorem to show that TRIO can approximate any continuous forward map and the complete family of target-dependent preimages arbitrarily well. **(5)** We show empirically that TRIO learns accurate forward predictors and preimages while substantially accelerating exact inverse optimization over baselines.

2 Related Work

Prior work has largely studied the tension between expressive forward prediction, preimage approximation, and tractable downstream optimization along three three separate directions that we review below: (i) recovering preimages from trained predictors, (ii) solving inverse optimization problems through learned predictors, and (iii) designing architectures that make inverse reasoning more tractable.

(i) Preimage approximation: *How can we recover the whole inverse set of a trained network?* For a non-injective predictor, a given target level generally corresponds *not* to a single input but to an entire, potentially disconnected set of inputs. Recovering this set from a generic neural network is computationally challenging. Early methods construct under- and over-approximations by propagating output constraints backward through the network (Dathathri et al., 2019). More recent approaches refine these approximations through branch-and-bound (Kotha et al., 2023), relaxations (Zhang et al., 2025; Björklund et al., 2026), probabilistic approximation (Marzari et al., 2026), or targeted refinement procedures (Koller et al., 2026). $\Longrightarrow$ *These methods aim to approximate the preimage post hoc at substantial computational cost; TRIO has an explicit representation of the learned preimage by construction.*

(ii) Inverse optimization subject to learned predictors: *How can we optimize over inputs when a trained predictor determines the constraint?* A direct approach is to embed the learned forward model into the downstream optimization problem and search over its inputs (Bergman et al., 2022; Ceccon et al., 2022). Depending on the model class, this may involve gradient-based nonlinear optimization (Wächter & Biegler, 2006), mixed-integer formulations for piecewise-linear networks (Anderson et al., 2020; Tsay et al., 2021), or deterministic global optimization through relaxations and spatial branch-and-bound (Schweidtmann & Mitsos, 2019; Bestuzheva et al., 2025). Across these approaches, the feasible set remains defined implicitly by the learned predictor, so each new downstream objective requires solving a new complex optimization problem through the nonlinear model. $\Longrightarrow$ *These methods optimize over a feasible set defined implicitly by the predictor; TRIO instead represents this set explicitly, which removes the nonlinear predictor from subsequent downstream optimization.*

[1] Code is available at `https://anonymous.4open.science/r/75A7`.

(iii) Customized architectures for tractable inverse reasoning: *Can the predictor itself be designed to make backward reasoning tractable?* Input convex neural networks (ICNNs) impose convexity to enable efficient downstream optimization (Amos et al., 2017; Pfrommer et al., 2023), but this restricts their sublevel sets to convex geometry. Invertible neural networks (Jacobsen et al., 2018; Behrmann et al., 2019) impose bijectivity, so each output corresponds to a single input, which is inapplicable in settings where many inputs can attain the same output. Normalizing flows similarly rely on invertible transformations (Dinh et al., 2017; Papamakarios et al., 2021); when used for inverse reasoning, they can represent distributions over possible inputs, but do not provide the complete feasible preimage as an explicit set. $\Longrightarrow$ *These methods enforce convexity or bijectivity for tractability reasons; TRIO allows for non-injective, non-convex mappings and disconnected preimages.*

Research gap: Methods that jointly address (i)–(iii) are missing. To fill this gap, TRIO learns expressive predictors with exact, explicit, and reusable learned preimages, while preserving universal forward and backward approximation, and reducing downstream global optimization to computationally tractable optimization over ellipsoidal regions.

3 SETUP

Notation: Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the admissible input domain, and let $f : \mathcal{X} \to \mathbb{R}$ denote an unknown, potentially nonlinear and non-injective function. We observe training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, with $y_i = f(x_i)$.

For a target level $g \in \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ and any scalar map $h : \mathcal{X} \to \mathbb{R}$, we define its sublevel set

$$\mathcal{P}_h(g) \coloneqq \{x \in \mathcal{X} : h(x) \leq g\} = h^{-1}((-\infty, g]). \quad (1)$$

Thus, $\mathcal{P}_h(g)$ gives the preimage of the lower output set $(-\infty, g]$. We refer to $\mathcal{P}_h(g)$ simply as the *preimage* at level g throughout the paper. The sublevel sets are nested by definition; i.e., $g_1 \leq g_2$ implies $\mathcal{P}_h(g_1) \subseteq \mathcal{P}_h(g_2)$. Any such preimage mapping induces a Galois connection, which has a natural order-theoretic interpretation (see Supplement A.4).

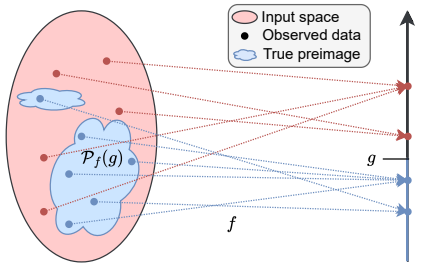


Figure 2: **Computing the preimage:** TRIO allows for non-injective, non-convex mappings and disconnected preimages.

Problem statement: We aim to learn a predictor F_θ that supports three tasks: (i) accurate forward prediction of f, (ii) approximation of the corresponding preimages $P_f(g)$, and (iii) tractable downstream optimization over these preimages. Importantly, we make *no injectivity or convexity assumption*; hence, a preimage may be nonconvex and disconnected.

Given an objective $J : \mathcal{X} \to \mathbb{R}$ (e.g., cost or risk), we aim to learn the following tasks:

$$\textbf{1}\ \underbrace{f : \mathcal{X} \to \mathbb{R},}_{\text{forward model}} \qquad \textbf{2}\ \underbrace{\mathcal{P}_f(g) \text{ for } g \in \mathbb{R}}_{\text{preimage}} \qquad \textbf{3}\ \underbrace{\min_{x \in \mathcal{X}} J(x) \text{ s.t. } f(x) \leq g\,.}_{\text{downstream optimization}} \quad (2)$$

An equivalent objective for the latter is $\min\{J(x) : x \in \mathcal{P}_f(g)\}$. These tasks are common in inverse design, safety, and surrogate-based optimization (Kotha et al., 2023; Ceccon et al., 2022).

Our aim is to learn a predictor that combines *(i) expressive forward modeling* with *(ii) exact preimage representation*, and *(iii) tractable downstream optimization over the preimage*. Here, F_θ captures the nonlinear structure of f, while its induced preimages $\mathcal{P}_{F_\theta}(g)$ admit exact[2], explicit, and reusable representations that support efficient querying and globally exact downstream optimization. *Hence, unlike prior work, we aim to do so (i)* ***without*** *post-hoc preimage recovery or separate inverse models, (ii)* ***without*** *repeatedly optimizing through the learned predictor, and (iii)* ***without*** *imposing invertibility or convexity on the forward map.*

[2]Throughout, *exact* refers to the learned predictor $\widehat{\mathcal{P}}(g) = \mathcal{P}_{F_\theta}(g)$. This structural guarantee is different from agreement with the unknown ground-truth preimage $\mathcal{P}_f(g)$.

4 TRIO

We now introduce TRIO and provide our main theoretical properties. **Section 4.1** defines the predictor class, and shows that, for every target level, the corresponding preimage has an exact finite representation in which expressive forward transformations factor out, and therefore removes the complexity during subsequent backward reasoning. **Section 4.2** uses the preimage representation to obtain tractable global downstream optimization over the learned preimage, and shows that this yields closed-form global solutions for linear objectives. Finally, **Section 4.3** establishes a universal approximation theorem, thus showing TRIO can approximate any continuous forward function and the entire resulting family of preimages arbitrarily well.

4.1 Exact preimage factorization with TRIO

Key idea: The key to TRIO is a *preimage factorization*: We parameterize forward expressiveness through radial transformations, such that its complexity factors out when computing the preimage.

Instead of using an unrestricted predictor for which preimages must later be recovered post hoc, we parameterize the forward model through expressive nonlinear transformations of learned radial coordinates with explicitly controlled geometry. Here, monotonicity along each radial coordinate ensures that, for a fixed target level, each transformation reduces to a simple scalar radius. Therefore, they completely factor out of the spatial preimage representation.

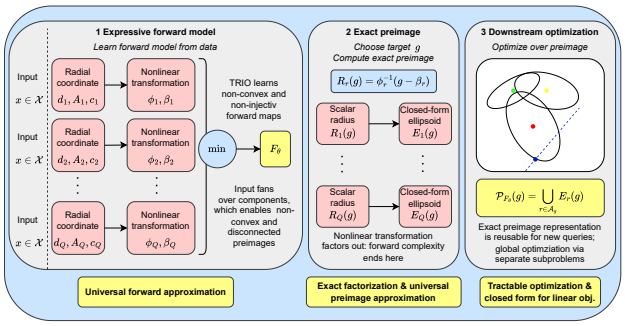


Figure 3: **TRIO:** The forward predictor is learned from data; for a chosen target, the preimage can be obtained exactly, which enables downstream optimziation over independent subproblems.

For a fixed target level, each radial component induces an ellipsoidal sublevel set. In the backward operation, the minimum envelope over these components then turns into a union of the corresponding ellipsoids, which allows TRIO to represent disconnected and nonconvex preimages. This construction preserves flexibility in the forward model while providing an exact and reusable representation of its preimages.

Tractable Representations for Preimage Learning and Inverse Optimization (TRIO)

TRIO defines

$$F_\theta(x) = \min_{r=1,\dots,Q} \left[\beta_r + \phi_r(d_r(x))\right], \tag{3}$$

for Q radial components, where each component applies a radial transformation ϕ_r along the learned radial coordinate

$$d_r(x) := \sqrt{(x - c_r)^\top A_r (x - c_r)}. \tag{4}$$

The bias terms are given by $\beta_r \in \mathbb{R}$; each radial transformation $\phi_r : [0, \infty) \to [0, \infty)$ satisfies $\phi_r(0) = 0$ and is continuous, unbounded, and strictly increasing; the locations are $c_r \in \mathbb{R}^d$; and the spatial geometries are controlled by $A_r \succ 0$. The outer minimum couples the transformations into a nonconvex and non-injective predictor which allows for disconnected preimages.

Intuitively, the construction provides flexibility along two complementary axes. First, each component learns its own center and anisotropic geometry, and the outer minimum allows different components to dominate in different parts of the input space. Their target-dependent sublevel sets therefore combine into expressive unions of ellipsoidal regions, which allows for disconnected and nonconvex preimages. Second, along each learned radial coordinate ϕ_r, each radial component can use a flexible nonlinear transformation ϕ_r, including neural networks, without complicating the backward geometry.

The above specification has direct implications for the inversion step: The preimage structure of TRIO is determined entirely by the predictor, **not** by a specialized training procedure. All parameters are learned jointly from the input data $\{(x_i, y_i)\}_{i=1}^N$ using standard empirical risk minimization such as mean squared error. Hence, *no inverse model, target-specific training, or expensive post-hoc preimage computation is required.*

Factoring out forward complexity in preimage computation: For a fixed target level g, the minimum envelope in Equation 3 turns the preimage into a union over components, while monotonicity reduces each radial transformation ϕ_r to a single target-level-dependent radius. The following theorem shows that this factorization is exact.

Theorem 4.1 (Exact preimage factorization)**.** *For any target level g, we define the active components $\mathcal{A}(g) \coloneqq \{r : g \geq \beta_r\}$, and, for each $r \in \mathcal{A}(g)$, the radius $R_r(g) \coloneqq \phi_r^{-1}(g - \beta_r)$. Then, the learned preimage is exactly*

$$\mathcal{P}_{F_\theta}(g) = \bigcup_{r \in \mathcal{A}(g)} E_r(g) \tag{5}$$

with $E_r(g) \coloneqq \left\{x \in \mathcal{X} : (x - c_r)^\top A_r (x - c_r) \leq R_r(g)^2\right\}$. Hence, every target level induces g an exact finite union of at most Q ellipsoidal regions.

Proof. See Supplement B. □

Theorem 4.1 holds for any parameter values satisfying the conditions above, and is therefore independent of training accuracy or generalization. It shows is that the complexity of the forward radial transformations does not carry over to computing the preimage. Each active radial transformation enters the preimage only through the scalar

$$R_r(g) = \phi_r^{-1}(g - \beta_r). \tag{6}$$

Hence, once the radius $R_r(g)$ is computed, ϕ_r no longer appears in the preimage representation. The learned preimage is therefore fully specified by the geometric parameters $\big(c_r, A_r, R_r(g)\big)$.

Computation of radii: The only required step in constructing the preimage is therefore the computation of the radii $R_r(g)$ through simple scalar inversion of $\phi_r(g)$. *The entire preimage is then available analytically.*

Radial model ϕ_r	Forward parameterization	Evaluation of ϕ_r^{-1}
(Broken) Power	(Piecewise) power law	(Piecewise) closed form
Monotone spline	Rational-quadratic spline	Analytic within a spline bin
Monotone neural	Strictly monotone neural network	One-dimensional root solve

Table 1: **Radial instantiations.** Different choices of ϕ_r change the forward parameterization while preserving the same ellipsoidal preimage representation.

For neural parameterizations of $\phi_r^{-1}(g)$, strict monotonicity allows $R_r(g)$ to be computed by simple one-dimensional root-finding problem with a unique solution. Once the radii have been computed for a given target level, the resulting preimage representation can be reused across arbitrarily many downstream queries without reevaluating the radial transformations; we formalize this as an *amortization property* in Supplement A.1. Importantly, more expressive choices of ϕ_r (see Table 1) may change the forward model and the one-time radius computation, but not the form or spatial complexity of the resulting preimage representation.

$\Longrightarrow$ *TRIO represents each learned preimage exactly as a finite union of ellipsoidal geometries that can be reused for subsequent downstream optimization over the preimage. The radial transformations ϕ_r may be arbitrarily expressive, but, after a single scalar inversion, their complexity factors out, and subsequent preimage queries operate only on the resulting geometric regions.*

4.2 Efficient optimization over learned preimages

The preimage factorization of TRIO also has implications for the downstream optimization task

$$\min \{J(x) : \ x \in \mathcal{P}_{F_\theta}(g)\}. \tag{7}$$

For a generic neural predictor, the feasible set $F_\theta(x) \leq g$ is defined only implicitly by the learned model and is generally nonconvex. For TRIO, Theorem 4.1 replaces this implicit feasible set with

the exact finite union

$$\mathcal{P}_{F_\theta}(g) = \bigcup_{r \in \mathcal{A}(g)} E_r(g), \tag{8}$$

and therefore

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x) = \min_{r \in \mathcal{A}(g)} \inf_{x \in E_r(g)} J(x). \tag{9}$$

Thus, global optimization over the generally nonconvex learned preimage is solved by optimizing separately over regions $E_r(g)$, and selecting the best solution. This is useful when the optimization problems over each region are tractable. For example, if the input space $\mathcal{X}$ is convex, every active ellipsoid $E_r(g)$ is convex as well since $A_r \succ 0$. If, in addition, J is convex, then optimization over each region in Equation 9 is a convex optimization problem. Therefore, global optimization over the full nonconvex preimage reduces to finitely many convex optimization tasks.

The above reduction is especially relevant for linear objectives. For a generic nonlinear predictor F_θ (e.g., an MLP), even a linear downstream objective J can still lead to a highly difficult nonconvex inverse-constrained problem as the predictor imposes the feasibility constraint. Under TRIO, *complexity of F_θ has already been factored out*, and, when $\mathcal{X} = \mathbb{R}^d$, linear optimization over each ellipsoidal region has a closed-form solution. Evaluating these solutions across all active regions yields the global optimum over the complete, potentially disconnected preimage.

Theorem 4.2 (Closed-form linear optimization)**.** *Let $\mathcal{X} = \mathbb{R}^d$, let $a \in \mathbb{R}^d$ with $a \neq 0$, and fix a target level g with $\mathcal{A}(g) \neq \varnothing$. Let $J(x) = a^\top x$ be a linear objective function. Then*

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x) = \min_{r \in \mathcal{A}(g)} \left[a^\top c_r - R_r(g) \sqrt{a^\top A_r^{-1} a} \right]. \tag{10}$$

If $r^\star$ attains the minimum above, then a globally optimal solution is

$$x^\star = c_{r^\star} - \frac{R_{r^\star}(g)}{\sqrt{a^\top A_{r^\star}^{-1} a}} A_{r^\star}^{-1} a. \tag{11}$$

Thus, global linear optimization over the generally nonconvex, disconnected preimage reduces to evaluating one scalar objective value for each active radial component.

Proof. See Supplement B. □

Once the output-level-dependent radii are available, Theorem 4.2 removes the inverse-constrained optimization entirely. In other words, no iterative nonlinear solver or mixed-integer representation of the predictor is required. Instead, global optimality follows from evaluating the analytic solution associated with each active region, and then selecting the best one.

4.3 Universal approximation theorem

A natural question is whether the exact preimage structure of TRIO limits its approximation power. The following result shows that it does not: a single finite TRIO predictor can approximate any continuous forward function and the corresponding preimages arbitrarily well.

Theorem 4.3 (Universal approximation for forward predictor and preimage representation)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and let $f \in C(\mathcal{X})$. Then, for every $\varepsilon > 0$, there exists a finite TRIO predictor F_θ such that*

$$\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| < \varepsilon. \tag{12}$$

and, simultaneously for every target level $g \in \mathbb{R}$,

$$\mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + \varepsilon). \tag{13}$$

The result already holds for the restricted subclass $F_\theta(x) = \min_{r=1,\dots,Q} \left[\beta_r + \lambda \|x - c_r\|_2^2 \right]$, where $\lambda > 0$ is a common quadratic coefficient shared by all experts.

Proof. See Supplement B. □

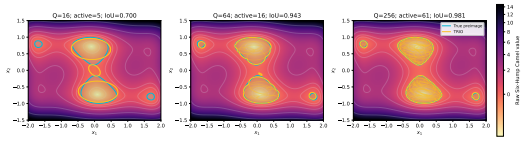
Theorems 4.1 and 4.3 establish two complementary guarantees. Theorem 4.1 shows that every TRIO preimage is represented *exactly* by a finite union of ellipsoidal regions. Theorem 4.3 shows that imposing this exact structure does *not* restrict what the model can ultimately learn. Instead, a single finite predictor can approximate any continuous forward map arbitrarily well, and, at the same time, its preimage representation learns the entire family of ground-truth preimages to arbitrarily fine resolution. Thus, the learned preimages are both *exact for the learned predictor* F_θ (Theorem 4.1) and *arbitrarily faithful to the ground-truth preimages* of the underlying function (Theorem 4.3).

Surprisingly, the above universal approximation theorem requires neither expressive radial transformations nor learned anisotropic metrics; it already holds for the restricted subclass with $A_r = I$ and $\phi_r(d) = \lambda d^2$. Richer choices of A_r and ϕ_r therefore provide additional flexibility for finite Q, while preserving the same exact preimage factorization. In particular, the target level g need not be specified during training. Here, Theorem 4.3 shows that the same learned predictor supports preimage queries over a continuum of target levels *without* changing the model or the form of its preimage representation.

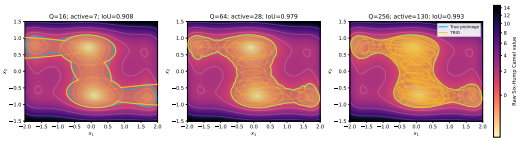
Approximation rate: For Lipschitz targets, we further show that TRIO achieves a constructive uniform approximation rate of $\mathcal{O}(Q^{-1/d})$ in the number of radial components Q; see Supplement A.2.

5 Experiments

The aim of our experiments is to validate the theoretical properties of TRIO empirically. Specifically, we analyze whether *(i)* TRIO retains sufficient expressiveness to achieve accurate forward prediction; *(ii)* TRIO learns accurate preimages while preserving their exact and explicit representation; and *(iii)* TRIO allows for efficient downstream optimization over the preimage. For a fair comparison, all neural baselines have the same number of parameters.



(a) Target level $g = -0.1$.



(b) Target level $g = 0.9$.

Figure 4: **Preimage reconstruction.** For fixed neural backbone, larger Q yields increasingly accurate geometric reconstruction, while the number of active experts varies across target levels.

We compare TRIO against baselines corresponding to the above aims. *(i)* For forward prediction, we use standard unrestricted MLPs, together with an input convex neural network (ICNN) (Amos et al., 2017) as a baseline with tractable convex sublevel sets. *(ii)* For preimage recovery, we compare against PREMAP2 (Björklund et al., 2026; Zhang et al., 2025) as a state-of-the-art post-hoc reconstruction method. *(iii)* For downstream optimization, we compare against standard approaches that optimize directly through the learned predictor: IPOPT (Wächter & Biegler, 2006) and SCIP (Bestuzheva et al., 2025) for smooth neural predictors (smooth MLPs), Gurobi (Gurobi Optimization, LLC, 2026) for continuous piecewise-linear networks (CPWL MLPs), and CLARABEL (Goulart & Chen, 2026) for ICNNs. All results are reported over ten seeds. Details about the datasets, training, and solver are in **Supplement C**.

Six-Hump Camel dataset. The Six-Hump Camel data provides a two-dimensional benchmark with strongly nonconvex and disconnected sublevel sets (details in Supplement C.1).

• *What do the learned preimages look like?* We use a neural radial backbone and vary only the number of radial components Q, thereby allowing us to isolate the effect of geometric capacity. Figure 4 shows how the learned preimage changes as Q increases from 16 to 256 for a fixed radial backbone. Here, we see that learned preimage progressively approximates the true boundary and can recover finer nonconvex structures. This is consistent with the intended behavior of TRIO: increasing Q provides a finer preimage representation, while different target levels g activate different components of the same trained predictor.

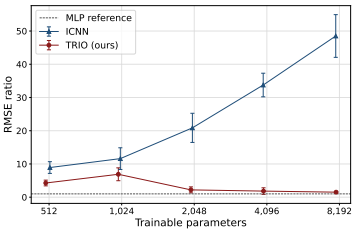


Figure 5: **Forward RMSE:** For increasing parameters, TRIO converges quickly to the forward RMSE of the MLP, while the ICNN completely deteriorates.

• *How does forward expressiveness improve as the model grows?* In Figure 5, we increase Q and show how forward RMSE of TRIO compares against that of an unrestricted MLP using the same number of parameters. For this, we report the ratio of the respective RMSEs. As the parameter count increases, the RMSE of TRIO approaches quickly approaches the MLP, while a same-sized ICNN leads to a large performance gap. Thus, increasing Q improves not only the preimage representation but also the forward predictive performance of TRIO, without imposing global convexity. This behavior is consistent with our universal approximation theorem.

• *Exact versus post-hoc preimage recovery.* We now assess the advantage of an exact preimage representation over post-hoc recovery. To do so, we compare the explicit preimage from TRIO against post-hoc recovery using PREMAP2 for an unrestricted MLP. Table 2 shows the benefits. TRIO provides the exact preimage by construction, thus achieves almost perfect coverage (i.e., itersection over union [IoU] of 0.96 compared to the true preimage) and requires only 0.01 s. For larger refinement budget, PREMAP2 reaches only 0.80 IoU, and, at $B = 2048$, the post-hoc recovery requires even 228.29 s, (=3.6×10^4 times the cost of TRIO). This shows the benefit of TRIO: the exact preimages *by construction* lead to almost perfect coverage of the learned preimage and computational speedups of several orders of magnitude.

Method	Budget	Certified coverage ↑	Truth IoU ↑	Time (s) ↓	Rel. time increase
PREMAP2	64	0.42 ± 0.03	0.42 ± 0.03	7.42 ± 0.24	1 178×
	128	0.63 ± 0.02	0.63 ± 0.02	14.51 ± 0.99	2 303×
	256	0.74 ± 0.02	0.74 ± 0.02	28.12 ± 1.90	4 463×
	512	0.78 ± 0.03	0.77 ± 0.03	53.83 ± 3.54	8 545×
	1 024	0.80 ± 0.03	0.79 ± 0.03	116.86 ± 29.96	18 549×
	2 048	0.80 ± 0.03	0.80 ± 0.03	228.29 ± 35.57	36 237×
TRIO (ours)	exact	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.96 \pm 0.02}$	$\mathbf{0.01 \pm 0.00}$	1×

Table 2: **Exact preimage vs. post-hoc recovery.** TRIO represents the learned preimage by construction, achieves a higher ground-truth IoU, and is up to 10^5 times faster.

Increasing nonlinearity. We next analyze whether the properties of TRIO hold when the underlying data becomes increasingly nonlinear. Here, we use a dataset based on complex powers and thus highly non-injective maps indexed by p, where larger p induce increasingly oscillatory forward dynamics and increasingly multimodal, disconnected preimages. We instantiate TRIO with the Broken-Power radial backbone and compare against a smooth MLPs, CPWL MLPs, and ICNNs. Experimental details are provided in the Supplement C.3.

• *Can TRIO maintain (i) forward accuracy and (ii) accurate preimage approximation as nonlinearity increases?* We measure (i) forward accuracy by test RMSE and (ii) backward accuracy by preimage IoU across target levels. The neural baselines do **not** provide an explicit preimage natively; we approximate their preimage using a grid-based forward evaluation and output thresholding.

Complex power	$p = 4$		$p = 6$		$p = 8$		$p = 10$		$p = 12$	
Model	RMSE [↓]	IoU [↑]	RMSE [↓]	IoU [↑]	RMSE [↓]	IoU [↑]	RMSE [↓]	IoU [↑]	RMSE [↓]	IoU [↑]
Smooth MLP	$\mathbf{0.17 \pm 0.05}$	$\mathbf{0.99 \pm 0.00}^\dagger$	0.56 ± 0.18	$0.95 \pm 0.01^\dagger$	0.83 ± 0.21	$0.90 \pm 0.02^\dagger$	3.15 ± 6.23	$0.79 \pm 0.15^\dagger$	3.18 ± 5.94	$0.75 \pm 0.12^\dagger$
CPWL MLP	0.68 ± 0.09	$0.98 \pm 0.00^\dagger$	1.37 ± 0.18	$0.92 \pm 0.01^\dagger$	2.26 ± 0.30	$0.85 \pm 0.01^\dagger$	3.89 ± 1.54	$0.76 \pm 0.02^\dagger$	4.61 ± 0.60	$0.68 \pm 0.01^\dagger$
ICNN	31.49 ± 0.24	$0.38 \pm 0.02^\dagger$	26.67 ± 0.34	$0.39 \pm 0.02^\dagger$	23.59 ± 0.27	$0.40 \pm 0.03^\dagger$	21.27 ± 0.20	$0.40 \pm 0.03^\dagger$	19.68 ± 0.36	$0.37 \pm 0.05^\dagger$
TRIO (ours)	0.19 ± 0.01	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.31 \pm 0.02}$	$\mathbf{0.97 \pm 0.00}$	$\mathbf{0.51 \pm 0.03}$	$\mathbf{0.94 \pm 0.00}$	$\mathbf{0.78 \pm 0.10}$	$\mathbf{0.88 \pm 0.01}$	$\mathbf{1.01 \pm 0.12}$	$\mathbf{0.84 \pm 0.01}$

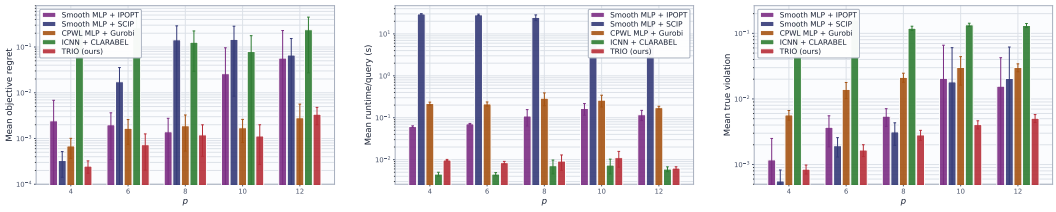
†: The preimages are **not** native to the baseline models, but are instead approximated by computationally expensive dense grid evaluation and output thresholding.

Table 3: **Forward accuracy and preimage approximation:** Forward RMSE ($\times 10^{-2}$; ↓ lower is better) and preimage IoU (↑ higher is better), reported as mean ± std. dev. over ten seeds.

Table 3 reports the results: (i) TRIO achieves the best forward accuracy from $p = 6$ onward. (ii) In terms of preimage approximation, TRIO achieves the highest IoU across all values of p, with a more pronounced advantage for larger p. The MLPs and ICNN are not a benchmark in itself but a performance comparison for predictors without the ability of generating preimages. Overall, this confirms the desired behavior by TRIO: it maintains both accurate forward prediction and accurate preimage approximation even for highly nonlinear settings.

• *Does the explicit preimage representation translate into more efficient downstream optimization?* Here, we solve, for each learned predictor and target level g, the projection problem $\min_x \frac{1}{2}\|x - x_0\|_2^2$ s.t. $F_\theta(x) \leq g$, which is the smallest perturbation required to enter the learned preimage. For the neural baselines, the constraint remains implicit in the predictor and is handled using global smooth nonlinear optimziation (SCIP for smooth MLP), global mixed integer optimization (Gurobi for CPWL MLP), local smooth nonlinear optimization (IPOPT for smooth MLP), or convex optimization (CLARABEL for ICNN). In contrast, TRIO *optimizes directly over the explicit preimage without additional solvers* (see Supplement C.5).

Figure 6 shows the results: (i) TRIO achieves the lowest regret wrt. objective; (ii) it achieves substantially lower per-query runtimes than optimization through the unrestricted neural predictors; and



(a) [↓] Objective regret (b) [↓] Runtime/query (s) (c) [↓] Constraint violation

Figure 6: **Downstream optimization. (a)** Regret wrt. objective (↓ lower is better); **(b)** Runtime per projection query (↓ lower is better); and **(c)** Constraint violation (↓ lower is better).

(iii) it achieves the lowest constraint violation. Overall, these results show that the explicit preimages from TRIO translates into accurate and computationally efficient downstream optimization.

• *Sensitivity to forward-model complexity.* Finally, we examine whether downstream optimization cost depends on the complexity of the forward radial model. Holding $Q = 128$ fixed, we increase radial backbone size from 1.2K to 500K parameters (Table 4). More expressive backbones increase only slightly the one-time scalar inversion cost, while subsequent optimization remains constant at 14 ms/query, with less than 3% variation between the smallest to largest model. Hence, once the preimage has been constructed, downstream optimization cost is independent backbone complexity.

Radial backbone	Parameters	One-time inversion (ms)	Time/query (ms)	Time/ellipse (μs)
Broken-Power	1 152	0.04 ± 0.02	13.68 ± 5.40	106.84 ± 42.21
Spline	3 200	6.32 ± 2.52	13.92 ± 5.84	108.78 ± 45.63
Neural-small	5 120	13.08 ± 5.08	13.82 ± 5.65	107.93 ± 44.11
Neural-mid	50 048	21.49 ± 8.95	13.90 ± 5.74	108.62 ± 44.82
Neural-large	500 096	72.16 ± 30.68	14.03 ± 5.94	109.59 ± 46.42

Table 4: **Sensitivity to forward complexity.** One-time inversion and downstream optimization time for increasingly complex radial backbones with fixed $Q = 128$. ⇒ The backbone complexity only affects scalar inversion cost, while optimization time remains unchanged.

Real-world application (AC power-flow): Finally, we test whether the same advantage of TRIO carry over to real-world application. For this, we use the IEEE 30-bus system from `pandapower` (Thurner et al., 2018) and maximize total renewable power injection across five buses, subject to the learned surrogate of the grid-security constraint. Each returned solution is then checked against the true AC power-flow. This yields a linear objective problem in which both efficient optimization over the learned model and feasibility under the real-world dynamics are important (details in Supplement C.1).

Table 5 reports the results across three criteria relevant in practice: the achieved objective value, feasibility under the AC power-flow simulator, and whether global optimality is established for the optimization problem defined by the learned predictor. TRIO is the only approach for which all runs are both AC-feasible and globally optimal with respect to the learned model, while requiring only 7 ms per solve. The baselines fail along at least one criterion: Gurobi solves the neural CPWL optimization problem globally, but several returned solutions violate the AC constraint; SCIP (solver time-out) and IPOPT (local solver) return AC-feasible solutions but without global optimality across all runs; and the ICNN fails in both. At the same time, TRIO is $32\times$–$3\,855\times$ faster than optimization through the neural baselines. Overall, TRIO achieves the desired behavior for application in practice: it offers fast global optimization while producing solutions that remain feasible under the underlying AC simulator.

Method	Returned MW	All AC-feasible	Global optimality	Solver time ↓	Rel. time increase
ICNN + CLARABEL	109.92 ± 0.21	✗*	✗*	0.301 ± 0.017	$43\times$
Smooth MLP + SCIP	111.95 ± 0.54	✓	✗	26.985 ± 8.380	$3\,855\times$
Smooth MLP + IPOPT	111.95 ± 0.54	✓	✗	0.221 ± 0.066	$32\times$
CPWL MLP + Gurobi	113.51 ± 0.43	✗	✓	11.468 ± 7.388	$1\,638\times$
TRIO (ours)	109.54 ± 0.92	✓	✓	$\mathbf{0.007 \pm 0.001}$	$1\times$

∗: ICNN had solver failures in 2 runs, and only 6 of 8 remaining runs had clean optimal status.

Table 5: **Downstream optimization using the AC power-flow data.** We report the returned objective value, whether all runs are AC-feasible, global optimality with respect to the learned model, and mean solve time. Runtime increase is measured relative to TRIO. ⇒ TRIO is the only approach that is both AC-feasible and globally optimal across all runs, while solving the problem in milliseconds.

Conclusion: To our knowledge, TRIO is the first framework for learning (i) expressive forward predictors with (ii) exact and explicit preimage representations by construction, and (iii) tractable global downstream optimization. We see broad applications in engineering and science, where inverse reasoning is central, but also in ML model inspection and auditing, where explicit preimages can help identify decision boundaries and safety-relevant regions of the input space.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing the proposed methods, with formulating and proving mathematical claims. We did not use generative AI tools to generate synthetic datasets, develop the theoretical framework, propose or refine hypotheses, design the research methodology or experiments, or interpret experimental results. Translation, dataset cleaning and reformatting, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to aid paper writing and to cross-check our related-work coverage for potentially overlooked relevant literature; all suggested references were independently verified by the authors. We reviewed all AI-assisted work, including the mathematical arguments and generated experimental artifacts. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide complete proofs for the theoretical results and detailed descriptions of the experimental setup, including the data-generating processes, model configurations, training procedures, and evaluation protocols. Code for reproducing the experiments will be made available with the paper. Synthetic datasets are generated from fully specified procedures, and the reported results can be reproduced using the provided implementation.

REFERENCES

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *ICML*, 2017.

Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183:3–39, 2020.

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Joern-Henrik Jacobsen. Invertible residual networks. In *ICML*, 2019.

David Bergman, Teng Huang, Philip Brooks, Andrea Lodi, and Arvind U. Raghunathan. JANOS: An integrated predictive and prescriptive modeling framework. *INFORMS Journal on Computing*, 34(2):807–816, 2022.

Ksenia Bestuzheva, Antonia Chmiela, Benjamin Müller, Felipe Serrano, Stefan Vigerske, and Fabian Wegscheider. Global optimization of mixed-integer nonlinear programs with SCIP 8. *Journal of Global Optimization*, 91:287–310, 2025.

Atharv Bhosekar and Marianthi Ierapetritou. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, 108:250–267, 2018.

Anton Björklund, Mykola Zaitsev, Paolo Morettin, and Marta Kwiatkowska. Efficient preimage approximation for neural network certification. *arXiv preprint*, arXiv:2505.22798, 2026.

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

Francesco Ceccon, Jordan Jalving, Joshua Haddad, Alexander Thebelt, Calvin Tsay, Carl D. Laird, and Ruth Misener. OMLT: Optimization & machine learning toolkit. *JMLR*, 23:1–8, 2022.

Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages*, pp. 238–252. ACM Press, 1977.

Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pp. 269–282. ACM Press, 1979.

Sumanth Dathathri, Sicun Gao, and Richard M. Murray. Inverse abstraction of neural networks using symbolic interpolation. In *AAAI*, 2019.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017.

Peter M. Dower. Coverage control and function approximation via semiconcave and semiconvex duality. *Mathematics of Control, Signals, and Systems*, 37:537–571, 2025.

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.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026.

Minh Hoang, Azza Fadhel, Aryan Deshwal, Janardhan Rao Doppa, and Trong Nghia Hoang. Learning surrogates for offline black-box optimization via gradient matching. In *ICML*, 2024.

Jörn-Henrik Jacobsen, Arnold W. M. Smeulders, and Edouard Oyallon. i-RevNet: Deep invertible networks. In *ICLR*, 2018.

Lukas Koller, Tobias Ladner, and Matthias Althoff. Out of the shadows: Exploring a latent space for neural network verification. In *ICLR*, 2026.

Suhas Kotha, Christopher Brix, Zico Kolter, Krishnamurthy Dvijotham, and Huan Zhang. Provably bounding neural network preimages. In *NeurIPS*, 2023.

Junhyeong Lee, Donggeun Park, Mingyu Lee, Hugon Lee, Kundo Park, Ikjin Lee, and Seunghwa Ryu. Machine learning-based inverse design methods considering data characteristics and design space size in materials design and manufacturing: a review. *Materials Horizons*, 10(12):5436–5456, 2023.

Luca Marzari, Manuele Bicego, Ferdinando Cicalese, and Alessandro Farinelli. On the probabilistic learnability of compact neural network preimage bounds. In *AAAI*, 2026.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22:1–64, 2021.

Samuel Pfrommer, Brendon G. Anderson, Julien Piet, and Somayeh Sojoudi. Asymmetric certified robustness via feature-convex neural networks. In *NeurIPS*, 2023.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Artur M. Schweidtmann and Alexander Mitsos. Deterministic global optimization with artificial neural networks embedded. *Journal of Optimization Theory and Applications*, 180:925–948, 2019.

Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, 2018.

Calvin Tsay, Jan Kronqvist, Alexander Thebelt, and Ruth Misener. Partition-based formulations for mixed-integer optimization of trained ReLU neural networks. In *NeurIPS*, 2021.

Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1): 25–57, 2006.

Xiyue Zhang, Benjie Wang, Marta Kwiatkowska, and Huan Zhang. PREMAP: A unifying PREiMage APproximation framework for neural networks. *JMLR*, 26:1–44, 2025.

A Additional theoretical results

A.1 Armotized factorization

Proposition A.1 (Amortized factorization)**.** *Let $C_{\mathrm{fact}}(g)$ denote the one-time cost of instantiating the exact geometric representation $\mathcal{P}_{F_\theta}(g) = \bigcup_{r\in\mathcal{A}(g)} E_r(g)$ including the computation of the radii $R_r(g) = \phi_r^{-1}(g - \beta_r)$. Suppose that K downstream queries are subsequently performed over $\mathcal{P}_{F_\theta}(g)$, and let C_k^{geo} denote the cost of the k-th query using only the resulting geometric representation. Then, the total cost is*

$$C_{\mathrm{fact}}(g) + \sum_{k=1}^{K} C_k^{\mathrm{geo}}. \tag{14}$$

All dependence on the radial transformations ϕ_r is confined to the one-time factorization cost $C_{\mathrm{fact}}(g)$. Hence, the forward-model-dependent overhead per downstream query is

$$\frac{C_{\mathrm{fact}}(g)}{K}, \tag{15}$$

which vanishes as $K \to \infty$.

Proof. See Supplement B. □

Proposition A.1 makes the computational effect of factorization explicit. For each target level, TRIO pays the cost of the expressive radial transformations only once when constructing the preimage; every subsequent membership query, projection, or downstream optimization problem reuses the same finite geometry without evaluating ϕ_r again. Consequently, the forward-model-dependent cost is amortized across repeated uses of the learned preimage and becomes negligible when the same target geometry supports many downstream queries.

A.2 APPROXIMATION RATES

Theorem A.2 (Finite-expert approximation rate)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and let $f : \mathcal{X} \to \mathbb{R}$ be L-Lipschitz with $L > 0$. Suppose that, for some $h > 0$, there exist centers $c_1, \dots, c_Q \in \mathcal{X}$ such that*

$$\sup_{x \in \mathcal{X}} \min_{r=1,\dots,Q} \|x - c_r\|_2 \le h. \tag{16}$$

Then there exists a TRIO predictor from the restricted isotropic quadratic subclass,

$$F_\theta(x) = \min_{r=1,\dots,Q} \left[\beta_r + \lambda \|x - c_r\|_2^2\right], \tag{17}$$

with the common coefficient $\lambda = \frac{L}{2h}$, such that

$$\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \le Lh. \tag{18}$$

Consequently, the corresponding preimages satisfy, simultaneously for every target level $g \in \mathbb{R}$,

$$\mathcal{P}_f(g - Lh) \subseteq \mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + Lh). \tag{19}$$

Proof. See Supplement B. □

Theorem A.2 makes the approximation–complexity tradeoff explicit: if the Q expert centers cover the domain at resolution h, then an isotropic quadratic TRIO predictor achieves uniform error at most Lh. At the same time, the same resolution controls the learned preimages uniformly across all target levels through a corresponding target-space error of size Lh.

Corollary A.3 (Rate in the number of experts)**.** *Let $h_Q(\mathcal{X})$ denote the optimal covering radius of $\mathcal{X}$ using Q centers. Under the assumptions of Theorem A.2,*

$$\inf_{F_\theta \in \mathcal{F}_Q} \sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \le L h_Q(\mathcal{X}), \tag{20}$$

where $\mathcal{F}_Q$ denotes the restricted isotropic quadratic TRIO class with at most Q experts.

In particular, if the covering numbers of $\mathcal{X}$ satisfy

$$N(\mathcal{X}, h) \le C_{\mathcal{X}} h^{-d}, \tag{21}$$

then

$$\inf_{F_\theta \in \mathcal{F}_Q} \sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \le L \left(\frac{C_{\mathcal{X}}}{Q}\right)^{1/d}. \tag{22}$$

Hence, on bounded d-dimensional domains, the restricted TRIO subclass achieves the constructive rate

$$\|F_\theta - f\|_\infty = \mathcal{O}\left(Q^{-1/d}\right). \tag{23}$$

Proof. See Supplement B. □

Corollary A.3 translates the geometric covering resolution into a rate in the number of experts. On bounded d-dimensional domains with standard covering-number scaling, the restricted TRIO class achieves the constructive rate $\|F_\theta - f\|_\infty = \mathcal{O}(Q^{-1/d})$.

A.3 From forward errors to optimization guarantees

The optimization guarantees in Section 4.2 are exact with respect to the learned predictor F_θ. A uniform approximation bound provides a direct link from these guarantees to the unknown ground-truth problem. In particular, if

$$\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| < \varepsilon, \tag{24}$$

then shifting the target level by ε yields the enclosure

$$\mathcal{P}_{F_\theta}(g - \varepsilon) \subseteq \mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g + \varepsilon). \tag{25}$$

The learned preimage at the tightened level $g - \varepsilon$ is guaranteed to contain only ground-truth feasible points. Conversely, every ground-truth feasible point is guaranteed to lie inside the learned preimage at the relaxed level $g + \varepsilon$. Since both sets are explicit under TRIO, this enclosure can be propagated directly to downstream optimization, which yields a ground-truth feasible solution together with a computable global guarantee.

Theorem A.4 (Ground-truth optimization)**.** *Suppose that* $\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \leq \varepsilon$*, and let* $x^\star_{\text{in}} \in \arg\min_{x \in \mathcal{P}_{F_\theta}(g-\varepsilon)} J(x)$*. Assume that the relevant optimal values are finite. Then*

$$x^\star_{\text{in}} \in \mathcal{P}_f(g) \tag{26}$$

and its suboptimality with respect to the ground-truth constrained problem satisfies

$$0 \leq J(x^\star_{\text{in}}) - \inf_{x \in \mathcal{P}_f(g)} J(x) \leq \inf_{x \in \mathcal{P}_{F_\theta}(g-\varepsilon)} J(x) - \inf_{x \in \mathcal{P}_{F_\theta}(g+\varepsilon)} J(x). \tag{27}$$

Proof. See Supplement B. □

If a bound on the forward error is available, Theorem A.4 gives an end-to-end guarantee. Optimizing over the tightened learned preimage produces a point that is guaranteed to satisfy the ground-truth constraint. Moreover, solving the same problem at the relaxed target level yields a computable certificate on how far this solution can be from the unknown ground-truth optimum. As both bounding preimages are tractable finite unions under TRIO, the certificate is obtained using the same region-wise optimization procedure as in Equation 9.

A.4 A GALOIS PERSPECTIVE

Galois perspective: For any scalar predictor F_θ, let

$$\mathcal{P}^\star_{F_\theta}(A) \coloneqq \sup_{x \in A} F_\theta(x), \tag{28}$$

with $\mathcal{P}^\star_{F_\theta}(\varnothing) = -\infty$. Then, $\mathcal{P}_{F_\theta}$ and $\mathcal{P}^\star_{F_\theta}$ satisfy

$$\mathcal{P}^\star_{F_\theta}(A) \le g \quad \Longleftrightarrow \quad A \subseteq \mathcal{P}_{F_\theta}(g), \tag{29}$$

and therefore form a Galois connection $\mathcal{P}^\star_{F_\theta} \dashv \mathcal{P}_{F_\theta}$ between $(2^{\mathcal{X}}, \subseteq)$ and $(\overline{\mathbb{R}}, \le)$ (Cousot & Cousot, 1977; 1979). This relation is canonical to any scalar predictor; the idea of TRIO is to construct predictors for which the corresponding backward map $\mathcal{P}_{F_\theta}(g)$ admits an exact and explicit finite representation.

A.5 HELPING LEMMAS

Lemma A.5 (Uniform approximation by isotropic quadratic experts)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and let $f \in C(\mathcal{X})$. Then, for every $\varepsilon > 0$, there exist a finite number of experts Q, centers $c_1, \dots, c_Q \in \mathcal{X}$, offsets $\beta_1, \dots, \beta_Q \in \mathbb{R}$, and a common coefficient $\lambda > 0$ such that*

$$F_\theta(x) = \min_{r=1,\dots,Q} \left[\beta_r + \lambda \|x - c_r\|_2^2\right] \tag{30}$$

satisfies

$$\sup\{x \in \mathcal{X} : \ |F_\theta(x) - f(x)| < \varepsilon\}. \tag{31}$$

Hence, the restricted isotropic quadratic subclass of TRIO is dense in $C(\mathcal{X})$ under the uniform norm.

Proof. See Supplement B. □

B PROOFS

Proof of Theorem 4.1. Consider first a finite target level $g \in \mathbb{R}$. By definition of the learned preimage,

$$x \in \mathcal{P}_{F_\theta}(g) \quad \Longleftrightarrow \quad x \in \mathcal{X} \text{ and} F_\theta(x) \le g. \tag{32}$$

Using the minimum representation of F_θ in Equation 3, we have

$$F_\theta(x) \le g \quad \Longleftrightarrow \quad \min_{r=1,\dots,Q} [\beta_r + \phi_r(d_r(x))] \le g \tag{33}$$

$$\Longleftrightarrow \quad \exists r \in \{1,\dots,Q\} : \beta_r + \phi_r(d_r(x)) \le g. \tag{34}$$

Thus, the outer minimum in the forward predictor becomes a disjunction over experts under the target-level query.

Because $d_r(x) \ge 0$, $\phi_r(0) = 0$, and ϕ_r is increasing,

$$\phi_r(d_r(x)) \ge 0. \tag{35}$$

Consequently, an expert with $g < \beta_r$ cannot satisfy the inequality in Equation 34. We may therefore restrict attention to the active set

$$\mathcal{A}(g) = \{r : g \ge \beta_r\}. \tag{36}$$

For every $r \in \mathcal{A}(g)$,

$$\beta_r + \phi_r(d_r(x)) \le g \quad \Longleftrightarrow \quad \phi_r(d_r(x)) \le g - \beta_r \tag{37}$$

$$\Longleftrightarrow \quad d_r(x) \le \phi_r^{-1}(g - \beta_r) \tag{38}$$

$$\Longleftrightarrow \quad d_r(x) \le R_r(g), \tag{39}$$

where the second equivalence follows from strict monotonicity of ϕ_r.

Substituting the definition

$$d_r(x) = \sqrt{(x - c_r)^\top A_r (x - c_r)} \tag{40}$$

and using $R_r(g) \ge 0$ gives

$$d_r(x) \le R_r(g) \quad \Longleftrightarrow \quad (x - c_r)^\top A_r (x - c_r) \le R_r(g)^2. \tag{41}$$

The latter condition is precisely $x \in E_r(g)$. Combining the preceding equivalences therefore yields

$$x \in \mathcal{P}_{F_\theta}(g) \quad \Longleftrightarrow \quad x \in E_r(g) \quad \text{for at least one } r \in \mathcal{A}(g) \tag{42}$$

$$\Longleftrightarrow \quad x \in \bigcup_{r \in \mathcal{A}(g)} E_r(g). \tag{43}$$

Since this holds for every $x \in \mathcal{X}$,

$$\mathcal{P}_{F_\theta}(g) = \bigcup_{r \in \mathcal{A}(g)} E_r(g). \tag{44}$$

For the extended target levels, the result follows by the natural conventions. Since F_θ is real-valued,

$$\mathcal{P}_{F_\theta}(-\infty) = \varnothing, \qquad \mathcal{P}_{F_\theta}(+\infty) = \mathcal{X}. \tag{45}$$

At $g = -\infty$, the active set is empty and hence the union is empty. At $g = +\infty$, all experts are active; defining $R_r(+\infty) = +\infty$ makes every $E_r(+\infty) = \mathcal{X}$, so their union equals $\mathcal{X}$. Thus the stated factorization holds for all $g \in \overline{\mathbb{R}}$. □

Proof of Equation 9. For each active region, define

$$v_r \coloneqq \inf_{x \in E_r(g)} J(x). \tag{46}$$

Since

$$E_r(g) \subseteq \mathcal{P}_{F_\theta}(g) \qquad \forall r \in \mathcal{A}(g), \tag{47}$$

optimization over the larger set cannot yield a larger infimum. Therefore,

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x) \leq v_r \qquad \forall r \in \mathcal{A}(g), \tag{48}$$

and hence

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x) \leq \min_{r \in \mathcal{A}(g)} v_r. \tag{49}$$

Conversely, since

$$\mathcal{P}_{F_\theta}(g) = \bigcup_{r \in \mathcal{A}(g)} E_r(g), \tag{50}$$

every $x \in \mathcal{P}_{F_\theta}(g)$ belongs to at least one active region. Thus, for some $r \in \mathcal{A}(g)$,

$$x \in E_r(g). \tag{51}$$

By definition of v_r,

$$J(x) \geq v_r \geq \min_{s \in \mathcal{A}(g)} v_s. \tag{52}$$

Since this holds for every $x \in \mathcal{P}_{F_\theta}(g)$, taking the infimum over $\mathcal{P}_{F_\theta}(g)$ gives

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x) \geq \min_{r \in \mathcal{A}(g)} v_r. \tag{53}$$

Combining Equation 49 and Equation 53, we obtain

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x) = \min_{r \in \mathcal{A}(g)} \inf_{x \in E_r(g)} J(x). \tag{54}$$

Finally, suppose that the regionwise infima are attained. Since $\mathcal{A}(g)$ is finite, there exists

$$r^\star \in \arg\min_{r \in \mathcal{A}(g)} v_r. \tag{55}$$

Let

$$x^\star_{r^\star} \in \arg\min_{x \in E_{r^\star}(g)} J(x). \tag{56}$$

Then

$$J(x^\star_{r^\star}) = v_{r^\star} = \inf_{x \in \mathcal{P}_{F_\theta}(g)} J(x), \tag{57}$$

so $x^\star_{r^\star}$ is a globally optimal solution over the complete learned preimage.

□

Proof of Theorem 4.2. The optimization of a linear functional over an ellipsoid is a standard support-function calculation (Rockafellar, 1970; Boyd & Vandenberghe, 2004). We provide the short derivation here for completeness and then apply it to the finite union induced by TRIO.

Fix a target level g with $\mathcal{A}(g) \neq \varnothing$. By Theorem 4.1,

$$\mathcal{P}_{F_\theta}(g) = \bigcup_{r \in \mathcal{A}(g)} E_r(g), \tag{58}$$

where

$$E_r(g) = \left\{ x \in \mathbb{R}^d : (x - c_r)^\top A_r (x - c_r) \leq R_r(g)^2 \right\}. \tag{59}$$

Since the union is finite,

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} a^\top x = \min_{r \in \mathcal{A}(g)} \inf_{x \in E_r(g)} a^\top x. \tag{60}$$

We therefore solve the optimization problem over an arbitrary active region $E_r(g)$. Since $A_r \succ 0$, let $A_r^{1/2}$ denote its unique symmetric positive-definite square root. Introduce the change of variables

$$y \coloneqq A_r^{1/2}(x - c_r). \tag{61}$$

Equivalently,

$$x = c_r + A_r^{-1/2} y. \tag{62}$$

Under this transformation, the ellipsoidal constraint becomes

$$\|y\|_2 \leq R_r(g), \tag{63}$$

while the objective satisfies

$$a^\top x = a^\top c_r + a^\top A_r^{-1/2} y = a^\top c_r + \left(A_r^{-1/2} a \right)^\top y. \tag{64}$$

Hence,

$$\inf_{x \in E_r(g)} a^\top x = a^\top c_r + \inf_{\|y\|_2 \leq R_r(g)} \left(A_r^{-1/2} a \right)^\top y. \tag{65}$$

By the Cauchy–Schwarz inequality,

$$\left(A_r^{-1/2} a \right)^\top y \geq - \left\| A_r^{-1/2} a \right\|_2 \|y\|_2 \geq -R_r(g) \left\| A_r^{-1/2} a \right\|_2. \tag{66}$$

This bound is attained at

$$y_r^\star = -R_r(g) \frac{A_r^{-1/2} a}{\left\| A_r^{-1/2} a \right\|_2}. \tag{67}$$

The denominator is strictly positive because $a \neq 0$ and A_r is positive definite. Moreover,

$$\left\| A_r^{-1/2} a \right\|_2^2 = a^\top A_r^{-1} a. \tag{68}$$

Defining

$$s_r(a) \coloneqq \sqrt{a^\top A_r^{-1} a}, \tag{69}$$

we therefore obtain

$$\inf_{x \in E_r(g)} a^\top x = a^\top c_r - R_r(g) s_r(a). \tag{70}$$

Substituting Equation 70 into Equation 60 yields

$$\inf_{x \in \mathcal{P}_{F_\theta}(g)} a^\top x = \min_{r \in \mathcal{A}(g)} \left[a^\top c_r - R_r(g) \sqrt{a^\top A_r^{-1} a} \right]. \tag{71}$$

It remains to recover a globally optimal point. Transforming $y_r^\star$ in Equation 67 back to the original coordinates gives

$$x_r^\star = c_r + A_r^{-1/2} y_r^\star \tag{72}$$

$$= c_r - R_r(g) \frac{A_r^{-1} a}{\sqrt{a^\top A_r^{-1} a}}. \tag{73}$$

Let

$$r^\star \in \arg\min_{r \in \mathcal{A}(g)} \left[a^\top c_r - R_r(g) \sqrt{a^\top A_r^{-1} a} \right]. \tag{74}$$

Then $x_{r^\star}^\star \in E_{r^\star}(g) \subseteq \mathcal{P}_{F_\theta}(g)$ and attains the smallest objective value over all active regions. Consequently,

$$x^\star = c_{r^\star} - \frac{R_{r^\star}(g)}{\sqrt{a^\top A_{r^\star}^{-1} a}} A_{r^\star}^{-1} a \tag{75}$$

is a globally optimal solution over the complete learned preimage.

Finally, substituting

$$R_r(g) = \phi_r^{-1}(g - \beta_r) \tag{76}$$

into Equation 70 gives the target-dependent value function

$$V_a(g) = \min_{r:\, g \geq \beta_r} \left[a^\top c_r - \sqrt{a^\top A_r^{-1} a}\, \phi_r^{-1}(g - \beta_r) \right]. \tag{77}$$

Thus, the globally optimal value over the generally nonconvex learned preimage is obtained by evaluating one closed-form scalar expression for each active expert and selecting the minimum. □

Proof of Theorem 4.3. Let $\varepsilon > 0$. By Lemma A.5, there exists a finite predictor G_θ from the restricted isotropic quadratic subclass,

$$G_\theta(x) = \min_{r=1,\dots,Q} \left[\beta_r + \lambda\|x - c_r\|_2^2\right], \tag{78}$$

with a common coefficient $\lambda > 0$, such that

$$\sup_{x\in\mathcal{X}} |G_\theta(x) - f(x)| < \frac{\varepsilon}{2}. \tag{79}$$

Define the shifted predictor

$$F_\theta(x) \coloneqq G_\theta(x) - \frac{\varepsilon}{2}. \tag{80}$$

This remains in the same restricted isotropic quadratic subclass, since

$$F_\theta(x) = \min_{r=1,\dots,Q} \left[\left(\beta_r - \frac{\varepsilon}{2}\right) + \lambda\|x - c_r\|_2^2\right]. \tag{81}$$

By Equation 79, for every $x \in \mathcal{X}$,

$$f(x) - \frac{\varepsilon}{2} < G_\theta(x) < f(x) + \frac{\varepsilon}{2}. \tag{82}$$

Subtracting $\varepsilon/2$ gives

$$f(x) - \varepsilon < F_\theta(x) < f(x) \qquad \forall x \in \mathcal{X}. \tag{83}$$

Hence,

$$\sup_{x\in\mathcal{X}} |F_\theta(x) - f(x)| < \varepsilon. \tag{84}$$

It remains to compare the complete families of sublevel-set preimages. Fix an arbitrary target level $g \in \mathbb{R}$.

First, let

$$x \in \mathcal{P}_f(g). \tag{85}$$

Then $f(x) \le g$, and Equation 83 gives

$$F_\theta(x) < f(x) \le g. \tag{86}$$

Therefore,

$$x \in \mathcal{P}_{F_\theta}(g), \tag{87}$$

and thus

$$\mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g). \tag{88}$$

Conversely, let

$$x \in \mathcal{P}_{F_\theta}(g). \tag{89}$$

Then $F_\theta(x) \le g$. Again using Equation 83,

$$f(x) < F_\theta(x) + \varepsilon \le g + \varepsilon. \tag{90}$$

Hence,

$$x \in \mathcal{P}_f(g + \varepsilon), \tag{91}$$

which yields

$$\mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + \varepsilon). \tag{92}$$

Combining Equation 88 and Equation 92, we obtain

$$\mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + \varepsilon). \tag{93}$$

Since $g \in \mathbb{R}$ was arbitrary, this inclusion holds simultaneously for every target level:

$$\mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + \varepsilon) \qquad \forall g \in \mathbb{R}. \tag{94}$$

Together with Equation 84, this proves that a single finite TRIO predictor can approximate f uniformly while simultaneously approximating its entire family of sublevel-set preimages up to an arbitrarily small shift in target level. Finally, the construction uses only the restricted subclass with $A_r = I$ and $\phi_r(d) = \lambda d^2$, with the same common $\lambda > 0$ for all experts. □

Proof of Theorem A.2. Let $c_1, \ldots, c_Q \in \mathcal{X}$ satisfy

$$\sup_{x \in \mathcal{X}} \min_{r=1,\ldots,Q} \|x - c_r\|_2 \leq h. \tag{95}$$

Choose

$$\lambda \coloneqq \frac{L}{2h}, \qquad \beta_r \coloneqq f(c_r) - \frac{Lh}{2}, \tag{96}$$

and define

$$F_\theta(x) = \min_{r=1,\ldots,Q} \left[f(c_r) - \frac{Lh}{2} + \frac{L}{2h}\|x - c_r\|_2^2 \right]. \tag{97}$$

This is a restricted TRIO predictor with

$$A_r = I, \qquad \phi_r(d) = \frac{L}{2h} d^2 \tag{98}$$

for every expert r.

We first establish the upper approximation bound. Fix any $x \in \mathcal{X}$. By the covering assumption, there exists an expert r such that

$$d \coloneqq \|x - c_r\|_2 \leq h. \tag{99}$$

Since f is L-Lipschitz,

$$f(c_r) \leq f(x) + Ld. \tag{100}$$

Using expert r as a candidate in the minimum,

$$F_\theta(x) - f(x) \leq Ld - \frac{Lh}{2} + \frac{L}{2h} d^2. \tag{101}$$

The right-hand side is increasing in $d \geq 0$. Since $d \leq h$,

$$F_\theta(x) - f(x) \leq Lh - \frac{Lh}{2} + \frac{Lh}{2} \tag{102}$$

$$= Lh. \tag{103}$$

We next establish the lower bound. Consider any expert $r \in \{1, \ldots, Q\}$ and write

$$d_r \coloneqq \|x - c_r\|_2. \tag{104}$$

By Lipschitz continuity,

$$f(c_r) \geq f(x) - Ld_r. \tag{105}$$

Therefore,

$$f(c_r) - \frac{Lh}{2} + \frac{L}{2h} d_r^2 - f(x) \geq -Ld_r - \frac{Lh}{2} + \frac{L}{2h} d_r^2 \tag{106}$$

$$= \frac{L}{2h}(d_r - h)^2 - Lh \tag{107}$$

$$\geq -Lh. \tag{108}$$

Hence every expert satisfies

$$f(c_r) - \frac{Lh}{2} + \frac{L}{2h}\|x - c_r\|_2^2 \geq f(x) - Lh. \tag{109}$$

Taking the minimum over all experts preserves the bound:

$$F_\theta(x) \geq f(x) - Lh. \tag{110}$$

Combining Equation 103 and Equation 110 yields

$$|F_\theta(x) - f(x)| \leq Lh. \tag{111}$$

Since $x \in \mathcal{X}$ was arbitrary,

$$\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \leq Lh. \tag{112}$$

It remains to establish the corresponding preimage bounds. The uniform approximation estimate implies

$$f(x) - Lh \leq F_\theta(x) \leq f(x) + Lh \qquad \forall x \in \mathcal{X}. \tag{113}$$

Fix an arbitrary target level $g \in \mathbb{R}$.

If

$$x \in \mathcal{P}_f(g - Lh), \tag{114}$$

then $f(x) \leq g - Lh$, and Equation 113 gives

$$F_\theta(x) \leq f(x) + Lh \leq g. \tag{115}$$

Thus,

$$\mathcal{P}_f(g - Lh) \subseteq \mathcal{P}_{F_\theta}(g). \tag{116}$$

Conversely, if

$$x \in \mathcal{P}_{F_\theta}(g), \tag{117}$$

then $F_\theta(x) \leq g$. Again using Equation 113,

$$f(x) \leq F_\theta(x) + Lh \leq g + Lh, \tag{118}$$

and therefore

$$\mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + Lh). \tag{119}$$

Since g was arbitrary, we conclude that

$$\mathcal{P}_f(g - Lh) \subseteq \mathcal{P}_{F_\theta}(g) \subseteq \mathcal{P}_f(g + Lh) \qquad \forall g \in \mathbb{R}. \tag{120}$$

This proves the theorem. □

Proof of Corollary A.3. Let

$$h_Q(\mathcal{X}) \coloneqq \inf_{c_1,\dots,c_Q \in \mathcal{X}} \sup_{x \in \mathcal{X}} \min_{r=1,\dots,Q} \|x - c_r\|_2 \tag{121}$$

denote the optimal covering radius of $\mathcal{X}$ using Q centers.

Fix any $\eta > 0$. By the definition of the infimum, there exist centers $c_1, \dots, c_Q \in \mathcal{X}$ such that

$$\sup_{x \in \mathcal{X}} \min_{r=1,\dots,Q} \|x - c_r\|_2 \leq h_Q(\mathcal{X}) + \eta. \tag{122}$$

Applying Theorem A.2 with

$$h = h_Q(\mathcal{X}) + \eta \tag{123}$$

therefore yields a predictor $F_\theta \in \mathcal{F}_Q$ satisfying

$$\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \leq L\left(h_Q(\mathcal{X}) + \eta\right). \tag{124}$$

Hence,

$$\inf_{F_\theta \in \mathcal{F}_Q} \sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \leq L\left(h_Q(\mathcal{X}) + \eta\right). \tag{125}$$

Since $\eta > 0$ was arbitrary, letting $\eta \downarrow 0$ gives

$$\inf_{F_\theta \in \mathcal{F}_Q} \sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \leq L h_Q(\mathcal{X}). \tag{126}$$

Now suppose that the covering numbers of $\mathcal{X}$ satisfy

$$N(\mathcal{X}, h) \leq C_{\mathcal{X}} h^{-d}. \tag{127}$$

Choose

$$h = \left(\frac{C_{\mathcal{X}}}{Q}\right)^{1/d}. \tag{128}$$

Then Equation 127 gives

$$N(\mathcal{X}, h) \leq C_{\mathcal{X}} \left(\frac{C_{\mathcal{X}}}{Q}\right)^{-1} = Q. \tag{129}$$

Thus, $\mathcal{X}$ can be covered by at most Q Euclidean balls of radius h, and therefore

$$h_Q(\mathcal{X}) \leq \left(\frac{C_{\mathcal{X}}}{Q}\right)^{1/d}. \tag{130}$$

Substituting this bound into Equation 126 yields

$$\inf_{F_\theta \in \mathcal{F}_Q} \sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| \leq L \left(\frac{C_{\mathcal{X}}}{Q}\right)^{1/d}. \tag{131}$$

Consequently,

$$\inf_{F_\theta \in \mathcal{F}_Q} \|F_\theta - f\|_\infty = O\left(Q^{-1/d}\right), \tag{132}$$

which proves the corollary. □

Proof of Theorem A.4. The uniform approximation bound implies that, for every $x \in \mathcal{X}$,

$$f(x) - \varepsilon \le F_\theta(x) \le f(x) + \varepsilon. \tag{133}$$

We first establish the corresponding inclusion of feasible sets. Let

$$x \in \mathcal{P}_{F_\theta}(g - \varepsilon). \tag{134}$$

Then

$$F_\theta(x) \le g - \varepsilon. \tag{135}$$

Using Equation 133,

$$f(x) \le F_\theta(x) + \varepsilon \le g, \tag{136}$$

and hence

$$x \in \mathcal{P}_f(g). \tag{137}$$

Therefore,

$$\mathcal{P}_{F_\theta}(g - \varepsilon) \subseteq \mathcal{P}_f(g). \tag{138}$$

Conversely, let

$$x \in \mathcal{P}_f(g). \tag{139}$$

Then $f(x) \le g$, and Equation 133 gives

$$F_\theta(x) \le f(x) + \varepsilon \le g + \varepsilon. \tag{140}$$

Thus,

$$x \in \mathcal{P}_{F_\theta}(g + \varepsilon), \tag{141}$$

so

$$\mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g + \varepsilon). \tag{142}$$

Combining Equation 138 and Equation 142 yields

$$\mathcal{P}_{F_\theta}(g - \varepsilon) \subseteq \mathcal{P}_f(g) \subseteq \mathcal{P}_{F_\theta}(g + \varepsilon). \tag{143}$$

Since

$$x^\star_{\mathrm{in}} \in \mathcal{P}_{F_\theta}(g - \varepsilon), \tag{144}$$

the first inclusion in Equation 143 immediately gives

$$x^\star_{\mathrm{in}} \in \mathcal{P}_f(g), \tag{145}$$

so $x^\star_{\mathrm{in}}$ is feasible for the ground-truth constrained problem.

Next, define

$$v_{\mathrm{in}} \coloneqq \inf_{x \in \mathcal{P}_{F_\theta}(g-\varepsilon)} J(x), \tag{146}$$

$$v_f \coloneqq \inf_{x \in \mathcal{P}_f(g)} J(x), \tag{147}$$

$$v_{\mathrm{out}} \coloneqq \inf_{x \in \mathcal{P}_{F_\theta}(g+\varepsilon)} J(x). \tag{148}$$

Because minimization over a larger feasible set cannot increase the optimal value, the inclusions in Equation 143 imply

$$v_{\mathrm{out}} \le v_f \le v_{\mathrm{in}}. \tag{149}$$

By definition of $x^\star_{\mathrm{in}}$,

$$J(x^\star_{\mathrm{in}}) = v_{\mathrm{in}}. \tag{150}$$

Hence,

$$J(x^\star_{\mathrm{in}}) - v_f = v_{\mathrm{in}} - v_f \ge 0, \tag{151}$$

where the inequality follows from Equation 149. Moreover, since $v_f \ge v_{\mathrm{out}}$,

$$v_{\mathrm{in}} - v_f \le v_{\mathrm{in}} - v_{\mathrm{out}}. \tag{152}$$

Therefore,

$$0 \le J(x^\star_{\mathrm{in}}) - \inf_{x \in \mathcal{P}_f(g)} J(x) \le \inf_{x \in \mathcal{P}_{F_\theta}(g-\varepsilon)} J(x) - \inf_{x \in \mathcal{P}_{F_\theta}(g+\varepsilon)} J(x), \tag{153}$$

which proves the stated suboptimality certificate. □

Proof of Lemma A.5. The use of finite minima of quadratic basis functions is related to min-of-quadratics approximation constructions based on semiconcave and semiconvex representations; see, e.g., Dower (2025). We give a direct construction for the restricted TRIO subclass considered here.

Let $\varepsilon > 0$ and set

$$\eta \coloneqq \frac{\varepsilon}{3}. \tag{154}$$

Since $\mathcal{X}$ is compact and $f \in C(\mathcal{X})$, the function f is uniformly continuous. Hence, there exists $\rho > 0$ such that

$$\|x - y\|_2 < \rho \quad \Longrightarrow \quad |f(x) - f(y)| < \eta \qquad \forall x, y \in \mathcal{X}. \tag{155}$$

Compactness also implies that f attains its minimum and maximum on $\mathcal{X}$. Define

$$m \coloneqq \min_{x \in \mathcal{X}} f(x), \qquad M \coloneqq \max_{x \in \mathcal{X}} f(x), \qquad \Delta \coloneqq M - m. \tag{156}$$

Choose $\lambda > 0$ such that

$$\lambda \rho^2 > \Delta. \tag{157}$$

Next, choose $\delta > 0$ such that

$$\delta < \rho, \qquad \lambda \delta^2 < \eta. \tag{158}$$

Since $\mathcal{X}$ is compact, there exists a finite δ-net

$$\{c_1, \ldots, c_Q\} \subseteq \mathcal{X} \tag{159}$$

such that, for every $x \in \mathcal{X}$, there exists at least one $r \in \{1, \ldots, Q\}$ satisfying

$$\|x - c_r\|_2 < \delta. \tag{160}$$

For each center, set

$$\beta_r \coloneqq f(c_r), \tag{161}$$

and define

$$F_\theta(x) = \min_{r=1,\ldots,Q} \left[f(c_r) + \lambda \|x - c_r\|_2^2 \right]. \tag{162}$$

This is a special case of TRIO with

$$A_r = I, \qquad \phi_r(d) = \lambda d^2 \tag{163}$$

for every expert r.

We now bound the approximation error at an arbitrary $x \in \mathcal{X}$.

Upper bound. By Equation 160, there exists a center c_r such that

$$\|x - c_r\|_2 < \delta. \tag{164}$$

Since $\delta < \rho$, uniform continuity in Equation 155 gives

$$f(c_r) < f(x) + \eta. \tag{165}$$

Using this expert as a candidate in the minimum and applying Equation 158,

$$\begin{aligned} F_\theta(x) &\leq f(c_r) + \lambda \|x - c_r\|_2^2 && (166)\\ &< f(x) + \eta + \lambda \delta^2 && (167)\\ &< f(x) + 2\eta. && (168) \end{aligned}$$

Lower bound. Consider any expert $r \in \{1, \ldots, Q\}$. We distinguish two cases.

If

$$\|x - c_r\|_2 < \rho, \tag{169}$$

then uniform continuity implies

$$f(c_r) > f(x) - \eta. \tag{170}$$

Since the quadratic term is nonnegative,

$$f(c_r) + \lambda \|x - c_r\|_2^2 > f(x) - \eta. \tag{171}$$

If instead

$$\|x - c_r\|_2 \geq \rho, \tag{172}$$

then, using $f(c_r) \geq m$ and Equation 157,

$$\begin{aligned} f(c_r) + \lambda \|x - c_r\|_2^2 &\geq m + \lambda \rho^2 && (173)\\ &> m + \Delta && (174)\\ &= M && (175)\\ &\geq f(x). && (176) \end{aligned}$$

Thus, in either case,

$$f(c_r) + \lambda \|x - c_r\|_2^2 > f(x) - \eta. \tag{177}$$

Since this holds for every expert, taking the minimum over $r = 1, \ldots, Q$ yields

$$F_\theta(x) > f(x) - \eta. \tag{178}$$

Combining Equation 168 and Equation 178,

$$f(x) - \eta < F_\theta(x) < f(x) + 2\eta. \tag{179}$$

Therefore,

$$|F_\theta(x) - f(x)| < 2\eta = \frac{2\varepsilon}{3} < \varepsilon. \tag{180}$$

Since $x \in \mathcal{X}$ was arbitrary,

$$\sup_{x \in \mathcal{X}} |F_\theta(x) - f(x)| < \varepsilon. \tag{181}$$

Hence, the restricted isotropic quadratic subclass is dense in $C(\mathcal{X})$ under the uniform norm. □

C Details on the experiments

C.1 Data-generating processes and datasets

We evaluate the methods on three complementary regression-to-preimage problems. The first two benchmarks provide analytic ground truth and allow dense evaluation of increasingly challenging nonconvex sublevel sets, whereas the third replaces the analytic data-generating process by a nonlinear AC power-flow simulator. Table 6 summarizes the domains, data construction, and ground-truth evaluation used throughout the experiments.

Benchmark	Setting	Configuration
Six-Hump Camel	Input dimension / domain	2; $x \in [-2, 2] \times [-1.5, 1.5]$
	Model coordinates	$u = (x_1/2,\ 2x_2/3) \in [-1, 1]^2$
	Train / validation / test	100 000 / 5 000 / 5 000
	Preimage ground truth	Analytic f_{SH} evaluated on an independent 1201×1201 grid
	Thresholds	$g \in \{-0.8, -0.1, 0.4, 0.9, 2.15\}$
Complex powers	Input dimension / domain	2; $\{x \in \mathbb{R}^2 : \|x\|_2 \le 1\}$
	Powers	$p \in \{4, 6, 8, 10, 12\}$
	Train / validation / test	16 384 / 4 096 / 8 192 for each (p, s)
	Sampling	Independent uniform-in-area samples from the unit disk
	Preimage ground truth	Analytic f_p on an independent 801×801 grid; 19 training-label quantile thresholds
Pandapower	Input dimension / domain	5 renewable injections; $x \in [0, 25]^5$ MW
	Model coordinates	$u = x/25 \in [0, 1]^5$
	Physical system	IEEE-30 network; renewable buses $\{7, 17, 25, 2, 15\}$
	Dataset	20 000 deterministic scrambled Sobol designs
	Train / validation / test	12 000 / 4 000 / 4 000

Table 6: Data-generating processes and datasets used in the experiments. The synthetic benchmarks admit exact analytic labels and dense ground-truth preimage evaluation. For Pandapower, labels and physical feasibility are obtained from the frozen nonlinear AC power-flow simulation.

Six-Hump Camel. The Six-Hump Camel function provides a low-dimensional but strongly nonconvex benchmark with a multimodal landscape and nontrivial sublevel-set geometry. Its two-dimensional domain additionally permits dense visualization of the true preimage, which makes it useful for studying both the geometric capacity of TRIO and post-hoc preimage extraction.

For $x = (x_1, x_2)$, the regression target is

$$f_{\mathrm{SH}}(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + \left(-4 + 4x_2^2\right) x_2^2. \tag{182}$$

Supervised labels are obtained by evaluating Eq. equation 182 directly at these inputs; hence the synthetic targets are noise-free. Inputs are affinely mapped from the physical rectangle $[-2, 2] \times [-1.5, 1.5]$ to $[-1, 1]^2$ before training, while the target is left on its original scale.

Forward prediction is evaluated on the fixed 5 000-point test set. Separately, geometric quantities are computed on an independent 1201×1201 grid over the full physical rectangle. At threshold g, the ground-truth membership of every grid point is obtained directly as

$$\mathbf{1}\{f_{\mathrm{SH}}(x) \le g\}, \tag{183}$$

rather than from sampled or learned labels. This grid is used for the reported ground-truth preimage IoU and for the independent auditing of extracted preimages.

Complex powers. The complex-power family is a controlled stress test in which the difficulty of the inverse geometry increases while the underlying data-generating process remains exactly known (see Figure 7). Increasing the power p produces progressively more oscillatory and multimodal sublevel sets, which allows forward approximation and preimage recovery to be studied separately under increasing geometric complexity.

Writing $z = x_1 + \mathrm{i}x_2$, the raw response is

$$f_p(x) = \mathrm{Re}\left(e^{-\mathrm{i}\alpha} z^p\right), \qquad \alpha = 25^\circ, \qquad p \in \{4, 6, 8, 10, 12\}. \tag{184}$$

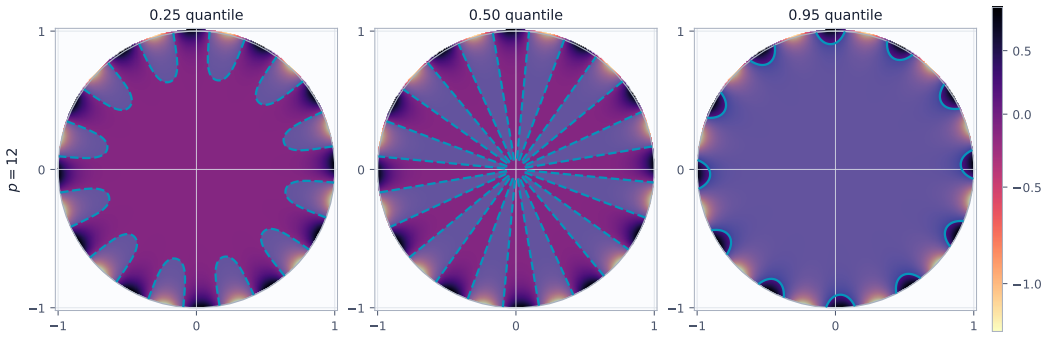


Figure 7: Complex power induces highly multimodel, disconnected preimages. Shown are the preimages at different target level quantiles. TRIO can approximate these arbitrarily close.

The supervised regression target contains the fixed offset

$$y_p(x) = f_p(x) - \delta, \qquad \delta = 0.13\cos(\alpha) \approx 0.1178200. \tag{185}$$

Accordingly, a threshold g in model-output coordinates corresponds to

$$g_{\text{raw}} = g + \delta \tag{186}$$

for the analytic DGP.

For every p, training, validation, and test inputs are sampled independently and uniformly in area from the unit disk. Their labels are obtained by direct evaluation of Equation 184 and Equation 185.

Forward RMSE is computed on the corresponding $8\,192$ held-out test points. Preimage accuracy is evaluated independently on an 801×801 Cartesian grid restricted to the unit disk. For each run, 19 thresholds corresponding to the $5\%, 10\%, \ldots, 95\%$ quantiles of the training labels are considered, and true grid membership is obtained directly from the analytic condition

$$f_p(x) \le g_{\text{raw}}. \tag{187}$$

Thus the backward/preimage evaluation does not reuse the supervised test sample.

Pandapower. The Pandapower benchmark (Thurner et al., 2018) moves from analytic functions to a physics-based power-system simulation. It represents a realistic application setting: the learned constraint approximates AC-grid security under five simultaneous renewable-power injections, and optimized designs are subsequently checked again by the nonlinear AC power-flow model.

We use the IEEE-30 network and place five static renewable generators at buses 7, 17, 25, 2, and 15, in this order, with zero reactive-power injection. For the physical design vector

$$x = (P_1, \ldots, P_5) \in [0, 25]^5 \text{ MW}, \tag{188}$$

a Newton–Raphson AC power flow is solved using flat initialization and at most 20 iterations. For every converged design, the supervised scalar target is the maximum normalized network-security loading,

$$f_{\text{AC}}(x) = \max\left\{ \max_{\ell} \frac{\text{loading}_\ell}{100},\ \max_{t} \frac{\text{loading}_t}{100},\ \max_{b} \frac{V_b}{V_b^{\max}},\ \max_{b} \frac{V_b^{\min}}{V_b} \right\}, \tag{189}$$

where a transformer term is omitted when no corresponding transformer result is present. The physical AC-feasibility condition is

$$f_{\text{AC}}(x) \le 1. \tag{190}$$

The dataset consists of $20\,000$ points from a deterministic scrambled five-dimensional Sobol desig. Each point is evaluated once by the frozen AC power-flow simulation to generate its regression target and feasibility label; all $20\,000$ simulations in the retained dataset converged. The selected $[0, 25]^5$ MW domain deliberately intersects the feasible boundary: approximately 49.25% of the resulting designs satisfy Equation 190. A single permutation assigns $12\,000$, $4\,000$, and $4\,000$ observations to training, validation, and test sets, and these exact splits are shared by all model seeds.

In contrast to the two analytic benchmarks, there is no dense high-dimensional ground-truth grid. Held-out prediction performance is therefore evaluated on the $4\,000$ independently reserved simulator-labeled test designs. For downstream optimization, each returned candidate is converted back to MW and re-evaluated by the same frozen AC power-flow simulation; this simulator evaluation determines physical AC feasibility, whereas any global optimality certificate refers only to the corresponding learned constraint.

C.2 TRIO ARCHITECTURES

TRIO is implemented as a minimum envelope of Q ellipsoidal experts. Each expert combines a learned center and positive-definite quadratic form with a strictly increasing scalar radial function. This separation is the key implementation property: the radial network is used while learning the scalar response, but for any fixed target level it can be inverted once and removed, leaving an explicit union of ellipsoids.

For expert r, let

$$q_r(x) = (x - c_r)^\top A_r (x - c_r), \qquad A_r \succ 0, \tag{191}$$

where c_r is the learned center. We parameterize A_r through a Cholesky factor, which enforces positive definiteness throughout training. The generic predictor takes the form

$$F_\theta(x) = \min_{r=1,\dots,Q} \left[\beta_r + \phi_r(d_r(x))\right], \qquad d_r(x) = \sqrt{q_r(x)}, \tag{192}$$

where β_r is an expert-specific offset and ϕ_r is constrained to be strictly increasing. The experiments use two radial backbones, summarized in Table 7.

Benchmark	Radial backbone	Q	Units per region	Initialization	Parameters
Complex powers	Broken-Power	128	–	Standard	1 152
Six-Hump Camel	Neural	$\{16, 32, 64, 128, 256\}$	8	Mixed low-target FPS	$31 \times Q$
Pandapower	Neural	128	32	Mixed low-target FPS	15 104

Table 7: TRIO configurations used in the experiments. The Broken-Power model is used for the complex-power benchmark, whereas the neural radial backbone is used for Six-Hump Camel and Pandapower.

Exact compilation. For a target level g, only experts with

$$\beta_r \le g \tag{193}$$

can contribute to the sublevel set. Since ϕ_r is strictly increasing, each active expert admits a unique radius

$$R_r(g) = \phi_r^{-1}(g - \beta_r)\,. \tag{194}$$

Consequently,

$$\{x : F_\theta(x) \le g\} = \bigcup_{r:\beta_r \le g} \left\{x : (x - c_r)^\top A_r (x - c_r) \le R_r(g)^2\right\}. \tag{195}$$

Thus, radial inversion is performed only once per active expert and target. Afterwards, preimage membership and downstream optimization use the detached quadratic representation in Equation 195; no radial-network evaluation is required.

For the neural radial laws, the inverse in Equation 194 is obtained by a one-dimensional bracketed root solve. In the Pandapower implementation, the tolerance is 10^{-12}. The complex-power Broken-Power backbone is likewise strictly monotone and is inverted once for each active expert before projection.

Radial backbones. For the complex-power benchmark, we use the Broken-Power backbone with $Q = 128$ experts and standard initialization. This is the same model for all powers $p \in \{4, 6, 8, 10, 12\}$ and contains 1 152 trainable parameters.

For Six-Hump Camel, each expert instead uses an anchored monotone Wide-Tanh radial map with eight Tanh units per region,

$$\phi_r(d) = a_r d + \sum_{j=1}^{8} v_{rj} \left[\tanh(w_{rj}(d - \kappa_{rj})) - \tanh(-w_{rj}\kappa_{rj})\right], \tag{196}$$

where a_r, v_{rj}, and w_{rj} are constrained to be positive. The subtraction term anchors the radial map at $\phi_r(0) = 0$, while the positive linear component prevents saturation and ensures strict monotonicity. The implementation uses the numerically stabilized radial distance

$$d_{\varepsilon,r}(x) = \sqrt{q_r(x) + \varepsilon} - \sqrt{\varepsilon}, \qquad \varepsilon = 10^{-12}. \tag{197}$$

Hence, if the inverse radial value is R_r, the corresponding detached quadratic constraint is

$$q_r(x) \le \left(R_r + \sqrt{\varepsilon}\right)^2 - \varepsilon. \tag{198}$$

The Pandapower model uses the same monotone neural-radial principle with 32 Tanh units per expert. For $v_r = \log(1 + q_r)$, its expert score is

$$\beta_r + a_r v_r + \sum_{j=1}^{32} w_{rj} \left[\tanh(s_{rj} v_r + t_{rj}) - \tanh(t_{rj})\right], \tag{199}$$

with a_r, w_{rj}, and s_{rj} constrained to be positive. This parameterization is again anchored and strictly increasing in the quadratic radius. Once the scalar inverse $v_r^\star$ has been found, the compiled ellipsoid radius follows directly from

$$q_r(x) \le \exp(v_r^\star) - 1. \tag{200}$$

Mixed low-target initialization. For the neural Wide-Tanh variants, we use a training-data-only initialization designed to place experts both throughout the input domain and in regions of small target values. Half of the Q centers are selected by deterministic farthest-point sampling (FPS) over the complete normalized training set. The remaining half are selected by FPS restricted to

$$\mathcal{D}_{\text{low}} = \{(x_i, y_i) : y_i \le q_{0.1}(y_{\text{train}})\}, \tag{201}$$

where $q_{0.1}$ denotes the empirical 10% target quantile. Given the previously selected centers, FPS iteratively chooses

$$c_{k+1} = \arg\max_{x_i \in \mathcal{D}} \min_{j \le k} \|x_i - c_j\|_2^2. \tag{202}$$

For every selected training point $x_{i(r)}$, we initialize

$$c_r = x_{i(r)}, \qquad \beta_r = y_{i(r)}. \tag{203}$$

The procedure uses only the training split and requires no analytic knowledge of the underlying data-generating process or simulator. It is used for the Six-Hump Camel and Pandapower experiments, while the complex-power experiments retain the standard Broken-Power initialization.

C.3 NEURAL BASELINES

We compare TRIO against standard unconstrained neural predictors and an input convex neural network (ICNN). Whenever architectures are compared at a fixed capacity, baseline widths are chosen to match the number of trainable parameters of TRIO as closely as possible. The CPWL MLP uses ReLU activations, while the Smooth MLP uses Tanh activations. The ICNN uses Softplus activations and structurally nonnegative hidden-to-hidden and output weights to preserve convexity of its scalar output. The architectures are summarized in Tables 8 and 9.

Benchmark	**Model**	**Architecture**	**Parameters**
Complex powers	TRIO	$Q = 128$ Broken-Power experts	1 152
	CPWL MLP	$2 \to 24 \to 42 \to 1$, ReLU	1 165
	Smooth MLP	$2 \to 24 \to 42 \to 1$, Tanh	1 165
	ICNN	two hidden layers, width 31, Softplus	1 212
Pandapower	TRIO	$Q = 128$ Wide-Tanh experts	15 104
	CPWL MLP	$5 \to 117 \to 121 \to 1$, ReLU	15 102
	Smooth MLP	$5 \to 117 \to 121 \to 1$, Tanh	15 102
	ICNN	two hidden layers, width 116, Softplus	15 086

Table 8: Parameter-matched predictor architectures for the complex-power and Pandapower experiments. Baseline widths are selected to closely match the trainable parameter count of TRIO.

For the Six-Hump Camel capacity study, the parameter budget varies with the number of TRIO experts. The matched CPWL MLP and ICNN architectures are therefore adjusted separately at each value of Q (see Table 9).

Q	TRIO		CPWL MLP		ICNN	
	Architecture	Params.	Hidden widths	Params.	Hidden width	Params.
16	16 experts	496	$(16, 25)$	499	19	516
32	32 experts	992	$(23, 37)$	995	28	1 011
64	64 experts	1 984	$(37, 48)$	1 984	41	2 012
128	128 experts	3 968	$(54, 68)$	3 971	59	3 956
256	256 experts	7 936	$(73, 103)$	7 945	85	7 908

Table 9: Parameter matching for the Six-Hump Camel capacity experiment. The CPWL MLP has two ReLU hidden layers with the listed widths; the ICNN has two Softplus hidden layers of the listed common width.

C.4 TRAINING HYPERPARAMETERS

All predictors, including TRIO and the neural baselines, are trained by direct scalar mean-squared error using Adam in double precision. We use the same core optimization settings across all three benchmarks (see Table 10).

Setting	Loss	Optimizer	Learning rate	Batch size	Precision	Validation interval
Configuration	MSE	Adam	10^{-3}	1,024	float64	500 steps

Table 10: Shared training hyperparameters used across the experiments.

For TRIO, the initial optimization phase uses a differentiable soft minimum over experts, with temperature cosine-annealed from 0.20 to 0.01 over the first $120{,}000$ steps. Training then switches to the literal hard minimum in Equation 192. Validation always evaluates the hard predictor, and all reported results use the checkpoint with the best validation performance. For comparability, the baseline models use the same $120{,}000$-step shield before early stopping and checkpoint eligibility, while training their standard forward maps throughout.

C.5 PREIMAGE EXTRACTION AND DOWNSTREAM OPTIMIZATION

The experiments distinguish between the learned scalar predictor and the procedure used to recover or optimize over its sublevel set. For TRIO, Equation 195 gives the complete learned preimage explicitly as a finite union of ellipsoids. The neural baselines instead require a post-hoc extraction or optimization procedure. Table 11 summarizes the methods used in the experiments.

Predictor	Inverse procedure	Used for	Learned-model guarantee
TRIO	Compiled ellipsoid union	Preimage extraction	Exact complete preimage
TRIO	Ellipsoid-wise optimization	Downstream optimization	Global optimum over learned preimage
CPWL MLP	PREMAP2	Preimage extraction	Certified inner approximation
CPWL MLP	Gurobi	Downstream optimization	Global on `OPTIMAL`
Smooth MLP	IPOPT	Downstream optimization	Local
Smooth MLP	SCIP	Downstream optimization	Global on `optimal`
ICNN	CLARABEL	Downstream optimization	Convex global solve on `optimal`

Table 11: Preimage extraction and downstream optimization procedures. Guarantees refer to the frozen learned predictor, not to the analytic data-generating process or the Pandapower AC equations.

Six-Hump preimage extraction. The quantitative extraction experiment uses the model parameter-matched at $Q = 64$ for $g = 0.4$. For TRIO, the learned sublevel set is obtained directly from Equation 195; hence its model-relative extraction coverage is exactly one. We compare this representation against PREMAP2 applied to the parameter-matched CPWL MLP.

PREMAP2 is run in its input space splitting mode and only its certified inner approximation is used. Since PREMAP2 operates as a classifier, the scalar sublevel-set condition is represented by the two-output map

$$G(x) = \begin{bmatrix} g - F_{\mathrm{CPWL}}(x) \\ 0 \end{bmatrix}, \tag{204}$$

for which classification into the first output is equivalent to $F_{\mathrm{CPWL}}(x) \le g$. We use refinement budgets

$$B \in \{64, 128, 256, 512, 1024, 2048\}. \tag{205}$$

For the certified inner set I_B and the complete learned CPWL preimage $S_{\mathrm{CPWL}}(g)$, extraction coverage is evaluated as

$$C_B = \frac{|I_B|}{|S_{\mathrm{CPWL}}(g)|} = \mathrm{IoU}(I_B, S_{\mathrm{CPWL}}(g)), \tag{206}$$

where the second equality follows from the certified inclusion $I_B \subseteq S_{\mathrm{CPWL}}(g)$. All reported certificates are independently decoded and checked on the 1201×1201 evaluation grid. The reported construction times measure once-per-target preimage extraction, i.e., TRIO compilation or PREMAP2 refinement, rather than subsequent point-membership queries.

Complex-power projection. For the complex-power benchmark, the downstream problem is the nearest-point projection

$$\min_x \|x - q\|_2^2 \qquad \text{s.t.} \qquad F_\theta(x) \le g, \quad \|x\|_2 \le 1. \tag{207}$$

For every power p, we use three target levels whose true sublevel sets have probability masses 0.2, 0.5, and 0.8 under uniform sampling from the unit disk. Twenty query points are drawn for each level conditional on being truly infeasible, yielding 60 nontrivial projection problems per power. The same queries are used for every predictor and model seed.

For TRIO, the learned constraint is represented as a finite union of ellipsoids,

$$\mathcal{P}_{F_\theta}(g) = \bigcup_{r \in \mathcal{A}(g)} E_r(g), \qquad E_r(g) = \left\{ x : (x - c_r)^\top A_r (x - c_r) \le R_r(g)^2 \right\}. \tag{208}$$

This representation enables direct optimization without evaluating the neural predictor. For the complex-power projection problem,

$$\min_{x} \|x - q\|_2^2 \qquad \text{s.t.} \qquad x \in \mathcal{P}_{F_\theta}(g), \tag{209}$$

the global solution is obtained by solving the projection problem on each active ellipsoid independently and selecting the candidate with the smallest distance. Projection onto a single ellipsoid is solved analytically through a one-dimensional monotone root-finding problem derived from the KKT conditions. Thus, the complete projection reduces to a finite number of scalar solves over the active experts.

The CPWL MLP is encoded exactly and solved with Gurobi; the Smooth MLP is optimized both with deterministic eight-start local IPOPT and with global SCIP spatial branch-and-bound, with a 30 s per-query limit for SCIP. The ICNN sublevel set is convex and is represented in CVXPY and solved using CLARABEL.

Because the complex-power DGP is analytic, the true projection optimum is computed independently rather than using any learned model. Writing a query as $q = \rho e^{\mathrm{i}\psi}$, the boundary of each angular branch can be reduced to a one-dimensional polar problem; all branches, relevant endpoints, and stationary candidates are considered. This reference computation is used only for evaluation and is excluded from all optimizer timings.

Pandapower calibration and optimization. The Pandapower experiment optimizes total renewable generation under a learned AC-security constraint,

$$\max_{u \in [0,1]^5} 25 \sum_{j=1}^{5} u_j \qquad \text{s.t.} \qquad F_\theta(u) \le \tau_\theta. \tag{210}$$

The threshold τ_θ is calibrated separately for every trained predictor using the validation set only. For a candidate threshold τ, we compute the false-feasible rate among truly AC-infeasible validation points,

$$\mathrm{FFR}_{\mathrm{val}}(\tau) = \frac{\#\{i : f_{\mathrm{AC}}(x_i) > 1,\, F_\theta(u_i) \le \tau\}}{\#\{i : f_{\mathrm{AC}}(x_i) > 1\}}, \tag{211}$$

and choose the largest candidate threshold satisfying $\mathrm{FFR}_{\mathrm{val}}(\tau) \le 10^{-3}$. No test-set or downstream-optimization outcome is used for calibration.

For TRIO, the calibrated constraint again compiles to a union of ellipsoids. The linear objective in Equation 210 is optimized exactly over the intersection of each ellipsoid with the box $[0, 1]^5$, and the best solution across active experts is returned. When box constraints are present, violated coordinates are fixed to their active bounds and the remaining free variables define a lower-dimensional ellipsoidal slice. The same closed-form solution is then applied recursively on the reduced ellipsoid. The best feasible candidate across all active ellipsoids is the global optimum of the compiled learned union intersected with the design box.

The parameter-matched CPWL MLP is encoded exactly as a mixed-integer linear model and solved by Gurobi. The Smooth MLP is solved using both SCIP spatial branch-and-bound and deterministic eight-start IPOPT, while the ICNN is represented as a convex epigraph problem and solved using CLARABEL. SCIP uses a 30 s solver limit. IPOPT is treated as a local method irrespective of agreement among its starts.

Finally, every returned design is converted from normalized coordinates to MW and evaluated by the frozen Pandapower AC power-flow simulation. This evaluation determines physical AC feasibility. In contrast, the "global" solver labels in Table 11 refer exclusively to optimality with respect to the corresponding learned constraint.

C.6 Evaluation conventions

All reported multi-seed results are aggregated over seeds 101–110 using the arithmetic mean and sample standard deviation. Uncertainty values in tables and figures denote this sample standard deviation across seeds with an AMD Ryzen 7 Pro CPU and 32GB of RAM.

For preimage evaluation, membership is always evaluated against the relevant ground-truth or learned-model set using the independent geometry grids described in Section C.1. For TRIO, exactness refers to equality between the neural predictor and its compiled ellipsoidal representation, not to equality with the underlying analytic function or simulator. Similarly, optimization certificates refer to the frozen learned constraint, while physical feasibility in the Pandapower experiment is determined only after re-evaluation by the AC power-flow simulator.

All experiments use Python implementations with double-precision arithmetic. The main learning experiments use PyTorch. The complex-power and Six-Hump experiments use NumPy/SciPy-based numerical utilities for geometry and optimization. The Pandapower experiments additionally use the Pandapower AC power-flow solver. Downstream optimization uses Gurobi (Gurobi Optimization, LLC, 2026) for CPWL MLPs, SCIP (Bestuzheva et al., 2025) and IPOPT (Wächter & Biegler, 2006) for Smooth MLP baselines, and CLARABEL (Goulart & Chen, 2026) through CVXPY for ICNNs (Amos et al., 2017). The PREMAP2 (Björklund et al., 2026; Zhang et al., 2025) comparison is executed in a separate Python environment due to its different dependency requirements. Details can be found in the attached code base.