Aligning Neural Inductive Biases with Physical Singularities: The ZeroProofML Framework

Zsolt Döme

Abstract

Scientific machine learning frequently requires modeling regimes characterized by singularities (resonance poles, kinematic locks, and censoring boundaries) where quantities become undefined or non-identifiable. Standard neural networks often exhibit an implicit smoothness bias that clips peaks, produces finite answers for undefined quantities, and varies substantially across random seeds near ill-conditioned regimes. We introduce ZeroProofML, a framework built on Signed Common Meadows (SCM), a totalized arithmetic where division by zero returns an absorptive element $\bot$ that propagates composably through downstream computation, instead of relying on unstructured `Inf`/`NaN` behavior or an ϵ-regularized surrogate. To reconcile strict algebraic semantics with gradient-based learning, we train on smooth projective tuples $\langle N, D\rangle$ and decode strictly at inference time (*Train on Smooth, Infer on Strict*). An angular parameterization of the projective output mitigates magnitude-coupling instabilities and yields symmetric, stable gating. We test the framework across three domains. ZeroProofML achieves near-zero false finite predictions on censored dose–response inputs ($\text{FP}_{\text{cens}} \approx 1.2 \times 10^{-4}$, $\text{FN}_{\text{in}} = 0$), nearly doubles peak-retention yield for extrapolation in a radio-frequency (RF) filter benchmark, and reduces seed-to-seed variance by 31.8× in near-singular inverse kinematics, at the cost of higher mean error relative to an unconstrained baseline. In the Dose experiment, curriculum-scheduled loss weighting recovers full 3-way regime identification, but with a regression quality trade-off that remains open. These results suggest that the choice of arithmetic is a consequential modeling decision in scientific ML, particularly in regimes dominated by censoring or near-singular structure.

1 Introduction

Deep learning's major advances (e.g., convolutions for images, attention for sequences) come from matching the architecture to the structure of the data. In Scientific Machine Learning (SciML), we lack an analogous match for one of the most pervasive features of physical systems: **singularities**. Gravitational poles, resonance peaks, and kinematic locks are not edge cases. They are central to the physics being modeled. Yet the standard toolbox (MLPs, Transformers) comes with an implicit *smoothness bias*: these networks are piecewise smooth / locally Lipschitz, and under typical training and regularization they tend to learn bounded-slope approximations that blunt singular growth. For phenomena that are genuinely non-Lipschitz, that bias can be the wrong shape.

The mismatch has become harder to ignore as PINNs and related SciML methods push neural surrogates into stiff, boundary-dominated regimes where limit behavior actually matters (Raissi et al., 2019). In practice, the consequences look like this:

1. **Soft walls.** An infinite potential barrier (say, an atomic collision) gets approximated by a linear slope—physically nonsensical extrapolation. Rational function classes can represent pole-like growth at low degree and avoid this failure (Boullé et al., 2020).
2. **Peak clipping.** High-Q_f resonance peaks in spectral domains are systematically underestimated. Pole-aware rational parameterizations are a natural fit here; smooth baselines simply attenuate the peak (Boullé et al., 2020; Molina et al., 2020).

3. **Optimization instability.** Near singularities, learned controllers exhibit high seed-to-seed variance, a well-known consequence of ill-conditioning (Higham, 2002; Trefethen & Bau, 1997).

The common workaround is ϵ-regularization: replace P/Q with $P/(Q+\epsilon)$ and hope for the best. This sidesteps the exception, but it also destroys the semantics. A downstream system cannot tell whether a value of 10^6 means "very large" or "undefined." The distinction matters—sometimes fatally—in safety-critical applications.

We take a different route. ZeroProofML replaces standard IEEE floating-point division (where division by zero yields Inf or NaN and, depending on environment, may raise exceptions or set IEEE flags) with **Signed Common Meadows** (SCM)[1] ($\mathbb{F}_\perp$) (Bergstra & Ponse, 2015; Bergstra & Tucker, 2024), an algebraic system where division is *total*: dividing by zero yields an absorptive element $\perp$ that propagates through the computation graph in a predictable, algebraically clean way. In exact arithmetic the semantic boundary is $D = 0$, which maps to $\perp$; in floating point, strict decoding uses a tolerance τ_{infer} to robustly detect proximity to this boundary. Unlike ϵ-regularization, τ_{infer} appears only in this final semantic check (not inside the rational evaluation), and the τ_{train}–τ_{infer} safety buffer (Proposition 3) bounds the probability of triggering strict $\perp$ decoding on finite targets; we additionally log the interval $\tau_{\text{infer}} \leq |D| < \tau_{\text{train}}$ as a diagnostic `gap_mask`.

The challenge is that $\perp$, being absorptive, kills gradients. Consequently, we cannot simply train in strict SCM mode. Our solution is a *Train on Smooth, Infer on Strict* protocol: during training, the network operates on smooth projective tuples $\langle N, D\rangle$ that keep gradients flowing; at inference time, we switch to strict SCM decoding that enforces exact algebraic semantics.

Concretely, our contributions are:

- We formalize how Signed Common Meadows can be used in neural network training, including the smooth-to-strict transition and optional gradient detachment for the renormalization scale (§2–§3).
- We introduce a dual-mode architecture: a *Projective Backbone* that emits homogeneous tuples $\langle N, D\rangle$, paired with a *Strict Inference* decoder that maps small denominators to $\perp$.
- We benchmark across three domains, censored dose–response (pharma), RF filter extrapolation (electronics), and near-singular inverse kinematics (robotics), each targeting a different aspect of singularity handling. The results are mixed in interesting ways, which we think is more informative than a clean sweep.

What we claim, and where to find the evidence. We want to be upfront about what the experiments do and do not show:

1. **Rejection beats hallucination** (Dose). Strict $\perp$ decoding nearly eliminates false in-range predictions on censored inputs with symmetric gating ($\text{FP}_{\text{cens}} \approx 1.2 \times 10^{-4}$; $\text{FN}_{\text{in}} = 0.0$), while a rational-only baseline without $\perp$ hallucinates finite values on $57.3\% \pm 0.28\%$ of censored inputs. See Table 1 and Figure 3. A curriculum-scheduled variant recovers full 3-way regime identification (Macro-F1 ≈ 0.904), suggesting that direction collapse is largely an optimization/order-of-training issue, with a regression trade-off.

2. **Rational structure helps spectral extrapolation** (RF). Under $33\times$ OOD extrapolation, the SCM model nearly doubles (increases by 94%) peak-retention yield and substantially reduces phase error compared to an MLP baseline. See Table 3 and Figure 4.

3. **Projective parameterization stabilizes near-singular optimization** (Robotics). Even when $\perp$ is not needed at inference, the SCM projective/rational parameterization improves optimization stability in an ill-conditioned regime, yielding $31.8\times$ lower seed variance than the unconstrained MLP. See Table 4 and Figure 5.

[1]We use **SCM** to mean *Signed Common Meadows*, not structural causal models.

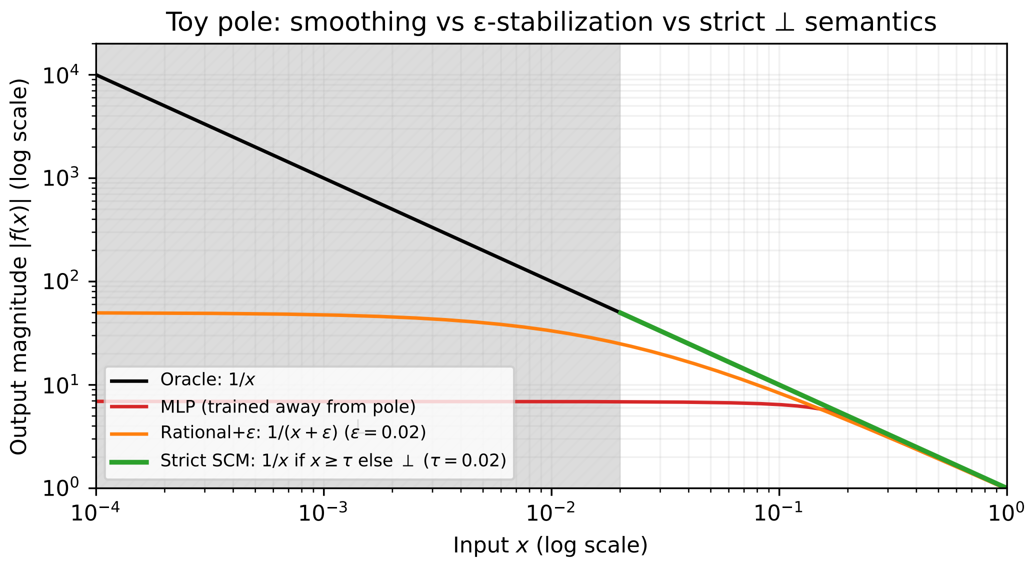


Figure 1: **Toy pole: smoothing vs. ϵ-regularization vs. strict $\perp$.** The MLP is trained on samples away from the pole and extrapolated toward $x \to 0$. A rational+ϵ surrogate saturates at a large finite value ($1/\epsilon$), while strict SCM decoding marks a neighborhood of the pole as $\perp$ (shaded reject region), making singularity handling explicit rather than heuristic.

When to use (and when not to use) SCM. For problems characterized by known singularities, censoring boundaries, or near-singular structure where the model must *say so* rather than paper over it, ZeroProofML is designed to help. For garden-variety smooth regression, standard architectures are fine, and the machinery described here is unnecessary overhead.

1.1 Toy Example: Why $\perp$ is not "a large number"

Why bother with an explicit $\perp$ when ϵ-regularization already "handles" the division? A simple example makes the difference concrete. Consider $f(x) = 1/x$ on $(0, 1]$, trained only on data away from the pole ($x \in [0.2, 1]$) and then evaluated as $x \to 0$ (Figure 1).

An MLP trained with MSE just extrapolates smoothly, it has no mechanism to represent divergence and stays bounded. The ϵ-regularized rational, $1/(x + \epsilon)$, does better in shape but saturates at $1/\epsilon$. It gives you a large number, not an "undefined" flag. Any downstream decision logic now has to guess: is 10^6 a real prediction or an artifact of the regularization? That heuristic boundary is both fragile and domain-specific.

Strict SCM decoding sidesteps the issue entirely. When $|Q(x)|$ drops below a threshold τ_{infer}, the output is $\perp$, an explicit rejection. The model is saying "I cannot produce a meaningful answer here," which is exactly the right behavior near a pole. Downstream systems can then invoke a fallback policy rather than acting on a hallucinated number.

This toy example is purely didactic and is not part of our benchmark suite; figure-generation details are listed in Appendix C.

2 Theoretical Framework

When we try to build neural surrogates that behave sensibly near singularities, we keep running into the same mismatch. Gradient-based optimization wants smooth, well-behaved functions. Poles and "undefined" regimes are neither. We do not think this is just a numerical nuisance. The arithmetic and the decoding semantics matter. In this section, we first spell out where smooth deep networks fail in singular limits, and then we introduce the algebraic machinery we use to make singularity handling explicit. Importantly, we do not claim that the individual propositions below are deep or novel in isolation: they are simple engineering guarantees and restatements of standard facts included to make the design constraints *auditable*. The theoretical contribution lives in the coherent *system design*: embedding meadow-style $\perp$ semantics into differentiable computation, separating smooth training from strict semantic inference (Train on Smooth, Infer on Strict), and (for censoring) using an angular projective parameterization and loss scheduling to stabilize the strict boundary.

2.1 The Limits of Smoothness

Standard neural networks (MLPs and Transformers alike, with ReLU, SiLU, or Tanh) inherit a strong *smoothness bias*. With Lipschitz activations at finite width, the resulting function is Lipschitz. That is great for interpolation. It is also the wrong default when the phenomenon you care about is a singular limit (resonance like $1/x$, barrier-like growth, or other non-Lipschitz behavior).

When we push a smooth model into a singular regime, what we usually get is an "affine fallback": for ReLU networks, restriction to any fixed ray $t \mapsto tv$ is eventually affine in t (Proposition 1 states the 1D case); for saturating activations such as Tanh the output itself is bounded; for smooth non-saturating activations such as SiLU the slope is bounded but the magnitude need not be—neither is piecewise affine. We refer to this bounded-response failure mode as the "Soft Wall" effect.

The fix we pursue is to constrain the hypothesis class to rational functions $P(\mathbf{x})/Q(\mathbf{x})$. As Boullé et al. (2020) note, *"rational neural networks approximate smooth functions more efficiently than ReLU networks with exponentially smaller depth."* For our purposes, though, efficiency is not the main point. The point is shape: rationals can represent steep transitions and divergent asymptotes in a way piecewise-affine extrapolation simply does not. Proposition 1 makes that contrast precise. The following proposition restates a standard structural property of ReLU networks (affine tails) alongside the usual asymptotics of rational functions; we include it to make the representation gap explicit and citable within our framework (Telgarsky, 2017).

Proposition 1 (Extrapolation hierarchy: affine vs. rational growth)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a feed-forward ReLU MLP. Then there exists $R > 0$ such that for all $|x| > R$, $f(x)$ is affine in x on each ray, hence grows at most linearly as $|x| \to \infty$. In contrast, if $g(x) = P(x)/Q(x)$ is a rational function with $\deg(P) = p$ and $\deg(Q) = q$ and $Q(x) \neq 0$ for sufficiently large $|x|$, then $g(x) = \Theta(|x|^{p-q})$ (asymptotically bounded above and below) as $|x| \to \infty$.*

Proof sketch. For ReLU MLPs, the activation pattern becomes constant outside a compact set on each ray, reducing the network to an affine map. For rationals, dividing the numerator and denominator by x^q and retaining leading terms captures arbitrary polynomial growth/decay rates. Related representation limits of ReLU networks are discussed in Telgarsky (2017).

2.2 The ϵ-Hack and Semantic Hallucination

Rationals address the representation issue, but they immediately surface a numerical one: division by zero.

In practice, the usual reflex is ϵ-regularization. We replace P/Q with $P/(Q + \epsilon)$ or $P/\sqrt{Q^2 + \epsilon^2}$ to avoid Inf or NaN (and other exceptional values) under floating-point evaluation (Higham, 2002).

Related rational-function parameterizations in deep networks also require denominator stabilization (e.g., Padé activation units (Molina et al., 2020)).

We explicitly avoid using ϵ-regularization as the semantic story. Yes, bounding the denominator stabilizes training. But it also forces mathematically undefined states to become *large, finite surrogates* (e.g., saturation at $1/\epsilon$). In a safety-critical loop, that is a problem: downstream logic cannot reliably tell the difference between a genuine extreme value and a regularization artifact that is masking an unidentifiable regime. Therefore, an ϵ-regularized model can look confident exactly where it should abstain.

2.3 Totalized Arithmetic via Meadows

We address this by stopping the encoding of "undefined" as "very large." Instead, we treat singular outcomes as explicit algebraic values. Concretely, we move from the standard real/complex field to the axioms of **Common Cancellation Meadows of Characteristic 0**. While meadow theory itself is well-established (Bergstra & Ponse, 2015; Bergstra & Tucker, 2025), our focus is on using it as a practical inductive bias in gradient-trained neural surrogates, where strict $\perp$ semantics must coexist with smooth gradient flow.

As Bergstra & Ponse (2015) describe it: *"Common meadows are fields expanded with a total inverse function. Division by zero produces an additional value … that propagates through all operations…"* In our setting, that means $1/0$ stops being a crash and becomes a value with explicit propagation rules.

Let $\mathbb{F}$ be a field ($\mathbb{R}$ or $\mathbb{C}$). The meadow $\mathbb{F}_\perp = \mathbb{F} \cup \{\perp\}$ extends the field with a totalized inverse and an absorptive error element $\perp$:

$$\frac{1}{0} = \perp, \quad x + \perp = \perp, \quad x \cdot \perp = \perp \quad (\forall x \in \mathbb{F}_\perp) \tag{1}$$

Why $\perp$ is not "just NaN".
Provenance: IEEE NaN can arise from many floating-point exceptional events (0/0, overflow, invalid ops, uninitialized values), and downstream code typically cannot recover *why*. In our SCM decoder, $\perp$ is emitted only by the explicit semantic check $|Q(\mathbf{x})| < \tau_{\text{infer}}$ at a known graph location.
Calibration: NaN boundaries are hardware/library artifacts; $\perp$ is a calibrated decision boundary shaped by training (separation and strict-boundary trigger control via $\tau_{\text{train}} > \tau_{\text{infer}}$, Proposition 3) and tuned at deployment via a τ_{infer} sweep (Appendix Figure 7).
Protocol: NaNs may propagate through arithmetic, but real pipelines often replace them (`nan_to_num`, `dropna`, default fills). We therefore carry $\perp$ in two channels: a NaN payload for passive propagation and an explicit `bottom_mask` for active tracking through silent NaN replacement.

Why do we prefer meadows over other types of algebra, like wheels (Carlström, 2004)? The main reason is that meadows allow for something called *fracterm flattening*. According to Bergstra & Tucker (2023), *"the equational axioms of common meadow fracterm calculus enable all fracterms to be flattened."* In practice, this means a complicated nest of meadow operations can be rewritten as a single fraction $P(\mathbf{x})/Q(\mathbf{x})$.

This is what makes the architecture workable. We can evaluate a forward pass without NaNs/Infs and postpone the singularity decision to a single end-of-network check (if $|Q(\mathbf{x})| < \tau_{\text{infer}}$, output $\perp$). In exact arithmetic one could take $\tau_{\text{infer}} = 0$. In floating point we set $\tau_{\text{infer}} > 0$ as a proximity guard for the semantic boundary $Q = 0$—and in censored tasks (Dose) we deliberately choose it as the operational decision line inside the $\tau_\perp < \tau_{\text{infer}} < \tau_{\text{finite}}$ separation hierarchy. If some sub-computation hits $\perp$, absorption forces the whole expression to collapse to $\perp$ as well. So rejection is not an afterthought; it is built into the semantics. Proposition 2 makes that concrete for censored targets.

Proposition 2 (Censored targets admit a projective $\perp$ encoding)**.** *Consider a dose–response assay with a finite concentration window $[c_{\min}, c_{\max}]$ and target $y = \log_{10}(\text{IC50})$. For samples where* IC50 *is identifiable within the assay window, we encode a finite target as a homogeneous tuple $\langle y_n, y_d \rangle = \langle y, 1 \rangle$. For censored samples (outside the window), the value is non-identifiable, and we encode only direction via $\langle y_n, y_d \rangle = \langle -1, 0 \rangle$ (below range) or $\langle +1, 0 \rangle$ (above range). Under strict SCM decoding, the prediction is finite if $|Q(\mathbf{x})| \geq \tau_{\text{infer}}$ and maps to $\perp$ otherwise, yielding a first-class "censored" state rather than a large finite surrogate.*

Proof sketch. The encoding is a direct application of homogeneous/projective targets: finite values correspond to points with $y_d \neq 0$, while censored values correspond to the ideal boundary $y_d = 0$. Strict SCM inference interprets the boundary as $\perp$ and propagates it algebraically.

2.4 Preserving Direction with Weak Signs

There is still a missing piece. In a plain common meadow, both $1/0$ and $-1/0$ map to the same $\perp$. For many scientific tasks, that is not enough. Direction matters: control effort can flip across a kinematic lock, and in a dose assay it matters whether the censoring is below-range or above-range.

To keep that directional context without breaking the meadow semantics, we add a **sign operator** $s(\cdot)$ on top of the algebra. This follows signed common meadows (Bergstra & Tucker, 2024) (building on signed involutive meadows (Bergstra & Ponse, 2017)):

$$s(z) = \begin{cases} z/|z| & \text{if } z \neq 0 \ \wedge \ z \neq \perp \\ 0 & \text{if } z = 0 \\ \perp & \text{if } z = \perp \end{cases} \tag{2}$$

For real and complex domains, this is a "weak sign": it projects non-singular values onto the unit sphere.

Now, strict equality to zero almost never occurs in training due to floating-point noise. We therefore implement a *hysteresis-capable* variant of the weak-sign operator (inspired by signed common meadows and signed involutive meadows). When $|z|$ enters a small singular band $|z| \leq \tau_{\text{hyst}}$, it can freeze and return the last observed orientation; it updates again only after leaving a wider release band $|z| \geq \tau_{\text{hyst}}(1 + \eta)$. (In the Dose benchmark we do not need history and use a simple sign test on the numerator for below/above; ties are broken deterministically.) For Dose, we additionally evaluate an *angular* parameterization of the projective circle (a specialization of the projective tuple) that removes magnitude gauge freedom and makes the separation and direction objectives geometrically orthogonal by construction (§3).

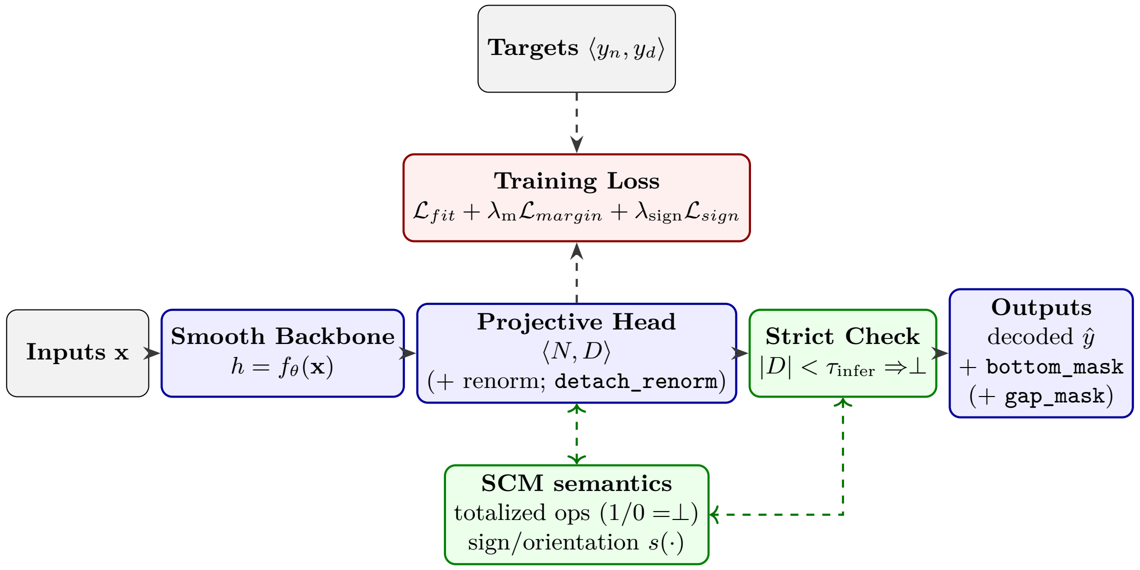
Design rationale (what is new here). The propositions in §2 are included as deployment-relevant sanity checks, not as standalone theoretical results. What is new is the system-level recipe: (i) treat $\perp$ as a first-class semantic output via meadows, (ii) train in a smooth projective space and enforce semantics only at inference, (iii) use an approximately scale-invariant projective fit loss that remains defined on singular targets, and (iv) introduce an empirically motivated angular specialization and curriculum schedule (Dose) to remove gauge-driven failure modes at the strict boundary.

3 Methodology: The ZeroProofML Architecture

We implement ZeroProofML using a "Train on Smooth, Infer on Strict" philosophy. This separates the numerical requirements of backpropagation from the semantic requirements of safety-critical deployment. This smooth-to-strict split is the core theoretical design move: it lets us optimize with well-behaved gradients while reserving strict $\perp$ semantics (and coverage monitoring) for the deployed computation. The overall architecture is visualized as a tuple-flow in Figure 2.

3.1 Concrete instantiation (RF resonator poles)

To make the framework concrete, we summarize the RF configuration used in our reference suite. (Exact run commands and bookkeeping live in Appendix C.)



Targets $\langle y_n, y_d \rangle$
Training Loss
$\mathcal{L}_{fit} + \lambda_{\mathrm{m}}\mathcal{L}_{margin} + \lambda_{\mathrm{sign}}\mathcal{L}_{sign}$
Inputs $\mathbf{x}$
Smooth Backbone
$h = f_\theta(\mathbf{x})$
Projective Head
$\langle N, D \rangle$
(+ renorm; detach_renorm)
Strict Check
$|D| < \tau_{\mathrm{infer}} \Rightarrow \perp$
Outputs
decoded $\hat{y}$
+ bottom_mask
(+ gap_mask)
SCM semantics
totalized ops ($1/0 = \perp$)
sign/orientation $s(\cdot)$

Figure 2: **The** ZeroProofML **architecture as a tuple-flow.** A smooth backbone maps inputs to features; a projective head emits a numerator/denominator projective tuple $\langle N, D \rangle$ (optionally renormalized for stable training; task-specific heads may denote this pair as (P, Q)). Strict inference applies a single denominator check ($|D| < \tau_{\mathrm{infer}} \Rightarrow \perp$) and returns a decoded scalar together with a `bottom_mask` and an optional `gap_mask` when $\tau_{\mathrm{train}} > \tau_{\mathrm{infer}}$. (Dose-specific losses such as $\mathcal{L}_{sep}$ are omitted; see Eq. (14).)

Task. Predict the complex resonator response $H(j\omega)$ from features $\mathbf{x} = [\log_{10}\omega, \log_{10}\omega_0, \log_{10} Q_f]$, where Q_f is the *quality factor*. Models output $(\Re H_\theta, \Im H_\theta)$; supervision uses a log-magnitude loss on $|H_\theta|$ (below). Training uses moderate quality factors ($Q_{f,\mathrm{train}} \leq 30$) while the test distribution includes near-pole high-Q_f regimes ($Q_{f,\mathrm{test}} \leq 1000$).

Data and evaluation. We train on 4096 filters (val 1024, test 2048). For training, each filter contributes 128 frequency samples via mixture sampling (global fraction 0.5). For peak metrics we evaluate each test filter on a per-filter local grid of 4096 frequencies (local span 20 widths; float64 peak evaluation) and report peak-ratio and yield at threshold 0.9.

Optimization (shared across methods). Adam; lr 2×10^{-3}; batch 1024; epochs 30; `restore-best` using validation MSE. Loss is log-magnitude MSE: $\big(\log(1 + |H_\theta|) - \log(1 + |H|)\big)^2$. Because supervision is magnitude-only, phase is not directly identified for unconstrained complex regressors; we report phase error as a diagnostic of inductive bias/coherence rather than an objective-aligned metric. For the constrained transfer family with real coefficients, the remaining ambiguity is consistent with a global sign (hence modulo π).

Models.

- **MLP (Deep):** direct regression with hidden dims $(512, 512, 512, 512)$ and SiLU.
- **MLP-Iso:** capacity-matched baseline with hidden dims $(128, 128)$ and SiLU.
- **RationalTransfer (non-SCM):** backbone $(128, 128)$ with ReLU predicts real polynomial coefficients for a rational transfer head $P(j\omega)/Q(j\omega)$ with degrees $(\deg P, \deg Q) = (2, 2)$ and monic Q (leading coefficient fixed to 1).
- **ZeroProofML-SCM (shared denom):** a hypernetwork conditioned only on $(\log_{10}\omega_0, \log_{10} Q_f)$ outputs coefficients for a constrained transfer family with degrees $(\deg P, \deg Q) = (0, 2)$ and monic Q; denominator coefficients are parameterized with `softplus`. The complex response is evaluated as $(P \cdot \overline{Q})/|Q|^2$ with SCM-safe division and a bottom mask when $|Q|$ is below τ_{infer} (implemented as `denom_eps`$= 10^{-8}$ in the RF code).
- **SCM (decoupled):** an ablation that uses two independent SCM rational heads for $(\Re H, \Im H)$ (no shared denominator), each with degrees $(6, 6)$ on a learned scalar latent.

3.2 Tier 1: Projective Directional Optimization (Training)

Numerical instability during backpropagation is the practical problem we are trying to dodge here. In our reference implementations, the backbone remains a standard real-valued network, but the head emits projective tuples $\langle N, D\rangle$ that represent the scalar N/D and are optimized with projective losses. Homogeneous coordinates still provide the right algebraic model for composing rational subgraphs:

$$\langle N_1, D_1\rangle + \langle N_2, D_2\rangle \to \langle N_1 D_2 + N_2 D_1, D_1 D_2\rangle \tag{3}$$

$$\text{Affine}(W, b, \langle N, D\rangle) \to \langle W \cdot N + b \cdot D, D\rangle \tag{4}$$

The tuple $\langle N, D\rangle$ is a homogeneous coordinate representation of the scalar N/D; scaling $\langle N, D\rangle \mapsto \langle \alpha N, \alpha D\rangle$ leaves the represented value unchanged. We therefore interpret tuple magnitude as a gauge degree of freedom and introduce a renormalization step to avoid overflow/underflow without changing semantics:

$$S = \sqrt{N^2 + D^2} + \varepsilon_{\text{renorm}}, \quad (N', D') \leftarrow (N/S, D/S) \tag{5}$$

Here $\varepsilon_{\text{renorm}} > 0$ is a small numerical-stability constant (we use $\varepsilon_{\text{renorm}} = 10^{-9}$ in all experiments; not tuned). We use two distinct detachment controls: (i) **Renorm detachment** (`detach_renorm`): whether to apply stop_gradient$(\cdot)$ to the norm term $\sqrt{N^2 + D^2}$ in the backward pass; and (ii) **Fit-scale detachment** (`detach_scale`): whether to apply stop_gradient$(\cdot)$ to the scale term in $\mathcal{L}_{fit}$ (below).

Angular projective parameterization (Dose follow-up). The free-magnitude tuple $\langle N, D\rangle \in \mathbb{R}^2$ is the general case, but for Dose we also evaluate a specialization that removes the magnitude gauge freedom entirely by parameterizing the projective circle with an angle:

$$N = \cos\theta, \quad D = \sin\theta. \tag{6}$$

This constrains $\langle N, D\rangle$ to the unit circle, so renormalization is unnecessary by construction. Rejection corresponds to θ near $k\pi$ (where $|D| = |\sin\theta| \approx 0$), while censored direction maps to which endpoint of the projective circle the model selects (sign of N). Empirically, this parameterization stabilizes the accept/reject gate (eliminating the FN_{in} instability observed under the free-magnitude parameterization in some schedules), yielding near-zero FP_{cens} and FN_{in}, and makes it harder for optimization to evade separation by shrinking tuple magnitude. This is an empirically grounded design choice motivated by the Dose ablation sequence, not a new algebraic result.

Curriculum scheduling for separation (Dose). In Dose, the censoring separation objective can dominate early and push many censored samples toward the poles before the backbone has established reliable directional features. We therefore evaluate a simple curriculum: set $\lambda_\perp = 0$ for an initial phase (direction establishment), then ramp $\lambda_\perp$ to its target value over subsequent epochs. A "hard" schedule (quickly ramping to $\lambda_\perp$=20) recovers 3-way regime identification (Macro-F1 ≈ 0.904) but can degrade conditional regression; a gentler schedule with lower peak separation weight provides a high-safety operating point with substantially better finite MAE (§5).

Projective directional optimization. When `detach_renorm` is enabled, S is treated as a constant in the backward pass. Consequently, the Jacobian of the normalization is a scalar rescaling ($\partial(N/S)/\partial N = 1/S$ and $\partial(D/S)/\partial D = 1/S$), rather than including additional terms that couple updates to the tuple magnitude through $\partial S/\partial(N, D)$. Operationally, this makes optimization less sensitive to magnitude shrinkage of $\langle N, D\rangle$ and encourages progress via changes in direction in projective space $\mathbb{P}(\mathbb{R}^2)$ (where $\mathbb{P}(\mathbb{R}^2)$ denotes the real projective line), while still allowing N and D to change. When `detach_renorm` is disabled, gradients propagate through S, which can be beneficial when the task includes explicit geometric constraints on $|D|$ (e.g., the Dose separation loss (14)). In particular for Dose, we need to reliably drive denominators into the censored band ($|D| \to 0$) on singular targets; setting `detach_renorm=False` lets the separation objective backpropagate through the renormalization scale so it can coordinate updates to the full tuple geometry rather than pushing on D alone while changes in N implicitly change S in the forward pass but remain invisible to that gradient signal. We report both behaviors as ablations in the DOSE benchmark (Table 6 in

Appendix A). These renormalized projective dynamics provide stable learning signals even near $D \approx 0$. The motivation is numerical stability near ill-conditioned regimes (overflow/underflow and cancellation) (Higham, 2002).

3.3 Tier 2: Fused Rational Units (Inference)

For deployment scenarios requiring deeper rational subgraphs, the library provides lightweight **fracterm flattening** utilities for small fused rational units (FRUs): given an explicitly represented rational expression, it can be flattened into an explicit polynomial fraction $P(\mathbf{x})/Q(\mathbf{x})$ (Bergstra & Tucker, 2023) (a symbolic helper, not a whole-network compilation pass), enabling singularity handling via a single denominator check. In the experiments reported here, the rational heads are shallow enough that P and Q are parameterized directly by the network head—flattening is not needed. FRUs become relevant when composing multiple rational layers or embedding SCM heads inside larger differentiable pipelines, which we leave to future work. Concretely, strict decoding applies:

$$\text{Output} = \begin{cases} \bot & \text{if } |Q(\mathbf{x})| < \tau_{\text{infer}} \\ P(\mathbf{x})/Q(\mathbf{x}) & \text{otherwise} \end{cases} \tag{7}$$

Strict decoding spec (and what inference returns). We use the phrase *strict SCM decoding* for the explicit map $\text{decode}_{\tau_{\text{infer}}} : (P, Q) \mapsto \bot$ if $|Q| < \tau_{\text{infer}}$ and P/Q otherwise. In the projective-tuple notation used elsewhere, the same numerator/denominator pair is written as $\langle N, D \rangle$. We use D for the generic denominator coordinate and Q when discussing task-specific transfer heads (so the same strict check may appear as $|D| < \tau_{\text{infer}}$ or $|Q| < \tau_{\text{infer}}$). In the library, strict inference returns three objects: a decoded scalar (with a NaN payload on bottom entries), an explicit `bottom_mask`, and an optional `gap_mask` when $\tau_{\text{train}} > \tau_{\text{infer}}$. In implementation terms, strict decoding represents $\bot$ via a dual carrier `(value=NaN, bottom_mask=True)`. The mask is the semantic carrier; the payload is a convenience and may be NaN (strict decoding) or zeroed (masked arithmetic primitives), depending on the context. (We still recommend treating `bottom_mask` as the authoritative carrier of the $\bot$ semantics.)

Downstream composition (why keep a `bottom_mask`).

```
# Pattern A: naive pipeline (NaN silently replaced)
ic50 = model.predict(x)
ic50 = nan_to_num(ic50, nan=0)
ti = ld50 - ic50         # finite, confidently wrong

# Pattern B: SCM pipeline (mask survives cleanup)
ic50, mask = scm_model.predict(x)
ti = ld50 - ic50         # NaN propagates
ti = nan_to_num(ti, nan=0)
assert mask[i] == True   # still tracks bottom
```

The point is not that NaN *cannot* propagate—it is that real code routinely breaks NaN propagation via cleanup, so the `bottom_mask` is insurance against silently producing finite defaults. For the Dose benchmark, we additionally define a *3-class regime label* from the same pair (P, Q): if $|Q| < \tau_{\text{infer}}$ we predict "censored" and use $\text{sign}(P)$ to choose `below` vs. `above`; otherwise we predict `in-range`. This keeps the scalar output semantics simple (finite vs. $\bot$) while still letting us report 3-class Macro-F1. For Dose, we write the numerator orientation test as `P<0` for readability; in code we treat the tie case `P=0` as `censored_above` (rare in practice). The theoretical justification is the fracterm flattening result for (common) meadows: meadow terms can be rewritten as a single fraction P/Q (Bergstra & Tucker, 2023). In our experiments, P and Q are parameterized directly by the network head; fracterm flattening becomes relevant when composing deeper rational subgraphs.

Algorithm 1 Strict decoding spec. `qabs` denotes $|Q|$.

```
function STRICT_DECODE($P, Q;\ \tau_{\text{infer}},\ \tau_{\text{train}} = \texttt{None}$)
    qabs $\leftarrow |Q|$
    bottom $\leftarrow \neg\texttt{isfinite}(P) \lor \neg\texttt{isfinite}(Q) \lor (\texttt{qabs} < \tau_{\text{infer}})$
    gap $\leftarrow (\tau_{\text{train}} \neq \texttt{None}) \land (\tau_{\text{train}} > \tau_{\text{infer}}) \land (\tau_{\text{infer}} \leq \texttt{qabs} < \tau_{\text{train}})$
    if bottom then
        $y \leftarrow$ NaN
    else
        $y \leftarrow P/Q$
    end if
    return ($y$, bottom_mask = bottom, gap_mask = gap)
end function

function DOSE_REGIME($P, Q;\ \tau_{\text{infer}}$)
    if $|Q| < \tau_{\text{infer}}$ then
        if $P < 0$ then
            return censored_below
        else
            return censored_above
        end if
    else
        return in_range
    end if
end function
```

Engineering note (fracterm flattening as a design principle). The fracterm flattening theorem justifies implementing SCM rational components so that singularity detection reduces to a single P/Q denominator check (Bergstra & Tucker, 2023). In our experiments, P and Q are parameterized directly by the network head (no runtime compilation); FRU utilities are only relevant when composing deeper rational subgraphs.

Threshold selection (making τ_{infer} the primary semantic knob). The implementation exposes multiple threshold-like hyperparameters; for practical use we group them into three tiers:

- **Tier 1 (task-defined; rarely tuned).** τ_{sing} is a floating-point label tolerance for detecting $y_d = 0$ (singular target) in the training tuples; we set it once at a machine-epsilon-scale default (and do not tune it), and in practice it is insensitive because $y_d \in \{0, 1\}$ by construction. τ_{hyst} (and hysteresis ratio η) define the weak-sign hysteresis band; for most tasks we keep these at fixed defaults, and for Dose we do not use history and instead use a simple sign test on the numerator for below/above.

- **Tier 2 (decision boundary; one degree of freedom).** $\tau_{\perp} < \tau_{\text{infer}} < \tau_{\text{finite}}$ define a nested separation hierarchy. In practice we treat τ_{infer} as the operational decision line for strict decoding and set the others as fixed ratios, e.g., $\tau_{\perp} = \tau_{\text{infer}}/2.5$ and $\tau_{\text{finite}} = 2\tau_{\text{infer}}$, reducing three thresholds to one. The train–infer gap buffer is set similarly, $\tau_{\text{train}} = c\,\tau_{\text{infer}}$ (e.g., $c \in \{4, 10\}$); when denominators are bounded away from zero by construction (e.g., SCM-Basic in Robotics), the exact choice of c is immaterial.

- **Tier 3 (loss weights; standard ML tuning).** The remaining scalars (e.g., $\lambda_{\perp}, \lambda_{\text{fin}}, \lambda_{\text{sign}}$) are multi-objective loss weights and are tuned like any other composite objective; they are not specific to SCM.

Thus, while the method uses several thresholds, the only semantically meaningful decision threshold is τ_{infer}; the rest can be set by fixed defaults/ratios plus standard loss-weight tuning.

3.4 Training Dynamics: The Target-Tuple Protocol

Standard Mean Squared Error (MSE) is undefined for singular targets. We utilize a composite objective function:

$$\mathcal{L} = \mathcal{L}_{fit} + \lambda_{\mathrm{m}}\mathcal{L}_{margin} + \lambda_{\mathrm{sign}}\mathcal{L}_{sign} \tag{8}$$

While each term is individually simple, the composite loss is a design contribution as a system: $\mathcal{L}_{fit}$ provides an approximately scale-invariant regression signal in projective space, $\mathcal{L}_{sign}$ resolves orientation on singular targets, and $\mathcal{L}_{margin}$ links training to a deployment-relevant strict boundary via an explicit safety buffer.

Implicit Fit Loss. We minimize the projective cross-product, normalized by a scale term to prevent nullity collapse:

$$\mathcal{L}_{fit} = \mathbb{E}\left[\frac{(N \cdot y_d - D \cdot y_n)^2}{D^2 y_d^2 + N^2 y_n^2 + \gamma}\right] \tag{9}$$

This loss is approximately scale-invariant (exact when $\gamma = 0$) and numerically stable even when $D \to 0$. Here $\gamma > 0$ is a small numerical guard (we use $\gamma = 10^{-9}$ in all experiments; not tuned) to avoid division by zero in degenerate cases. When `detach_scale` is enabled, we apply stop_gradient($\cdot$) to the scale term $D^2 y_d^2 + N^2 y_n^2$ in the backward pass: detaching the normalization prevents the optimizer from reducing $\mathcal{L}_{fit}$ by shrinking $\langle N, D\rangle$ instead of improving direction. Disabling `detach_scale` also lets losses that target $|D|$ (e.g., censoring separation) propagate through the fit normalization (reported as ablations). Note that this normalization pairs the predicted denominator D with the target denominator y_d (equivalently, the scale term is $\|(Ny_n,\ Dy_d)\|^2$ up to γ), so for finite targets (y_d=1) the D^2 component does not scale with the finite target magnitude $|y_n|$ (unlike naive scale terms such as $\|(N, D)\|^2 \cdot \|(y_n, y_d)\|^2$). This "same-index" pairing is intentional: swapping the pairing to $(Ny_d)^2 + (Dy_n)^2$ would make the loss nearly constant on singular targets (y_d=0), weakening the fit loss signal that drives $D \to 0$ in that regime. On singular targets, the scale term reduces to $N^2 y_n^2 + \gamma$ and could become γ-dominated only if the model predicts a finite, target-orthogonal tuple ($N \approx 0$, $|D| \approx 1$); projective renormalization bounds $\|(N, D)\|$, $\mathcal{L}_{sign}$ penalizes such degenerate orientations, and γ provides a well-defined value at the exact crossover.

Sign Consistency Loss. The implicit loss is sign-blind ($+\infty$ and $-\infty$ both yield zero cross-product). We introduce a projective cosine loss for singular targets to enforce correct orientation (with a label tolerance τ_{sing} to accommodate finite-precision representations):

$$\mathcal{L}_{sign} = \mathbb{E}\left[\mathbb{I}(|y_d| < \tau_{\mathrm{sing}}) \cdot \left(1 - \frac{Ny_n + Dy_d}{\|(N, D)\|\|(y_n, y_d)\|}\right)\right] \tag{10}$$

where $\|\cdot\|$ denotes the Euclidean (ℓ_2) norm.

Margin Loss. To control the "gap region" between smooth training and strict inference, we penalize denominators that approach the singular band:

$$\mathcal{L}_{margin} = \mathbb{E}\left[\mathbb{I}(|y_d| \geq \tau_{\mathrm{sing}})\ [\mathrm{ReLU}(\tau_{\mathrm{train}} - |D|)]^2\right]. \tag{11}$$

The indicator restricts the penalty to finite targets (in projective lifting, finite targets have $y_d = 1$ while singular targets have $y_d = 0$).

The following proposition is a direct application of Markov's inequality; we state it explicitly because it makes the τ_{train}–τ_{infer} safety buffer auditable and connects $\mathcal{L}_{margin}$ to a deployment-relevant probability bound.

Proposition 3 (Strict-boundary trigger bound (via $\mathcal{L}_{margin}$))**.** *Fix thresholds $\tau_{\mathrm{train}} > \tau_{\mathrm{infer}} > 0$ and define the strict-boundary trigger event on finite targets*

$$\mathcal{B} = \{|y_d| \geq \tau_{\mathrm{sing}}\ \wedge\ |D| < \tau_{\mathrm{infer}}\}. \tag{12}$$

Then

$$\mathbb{P}(\mathcal{B})\ \leq\ \frac{\mathcal{L}_{margin}}{(\tau_{\mathrm{train}} - \tau_{\mathrm{infer}})^2}. \tag{13}$$

Algorithm 2 ZeroProofML training (Train on Smooth, Infer on Strict)

```
Require: Training data $\{(\mathbf{x}_i, \langle y_{n,i}, y_{d,i}\rangle)\}$, thresholds $\tau_{\text{train}} > \tau_{\text{infer}} > 0$, label tolerance $\tau_{\text{sing}}$, weights $\lambda_{\text{m}}, \lambda_{\text{sign}}$, detachment flags detach_renorm, detach_scale, optimizer $\mathcal{O}$
 1: Initialize network parameters $\theta$ (projective backbone + rational head)
 2: for each minibatch $\mathcal{M}$ do
 3:     Forward pass in projective mode: $\langle N, D\rangle \leftarrow f_\theta(\mathbf{x})$ for all $(\mathbf{x}, \langle y_n, y_d\rangle) \in \mathcal{M}$
 4:     Compute $\mathcal{L}_{fit}$ using projective cross-product Eq. (9)
 5:     Compute margin penalty $\mathcal{L}_{margin}$ to discourage $|D| < \tau_{\text{train}}$ using Eq. (11)
 6:     Compute sign/orientation loss $\mathcal{L}_{sign}$ for singular targets using Eq. (10)
 7:     $\mathcal{L} \leftarrow \mathcal{L}_{fit} + \lambda_{\text{m}}\mathcal{L}_{margin} + \lambda_{\text{sign}}\mathcal{L}_{sign}$
 8:     Update $\theta \leftarrow \mathcal{O}(\theta, \nabla_\theta \mathcal{L})$
 9: end for
10: Inference: decode strictly with a single check: if $|D(\mathbf{x})| < \tau_{\text{infer}}$ then output $\perp$, else output $N(\mathbf{x})/D(\mathbf{x})$
```

Proof sketch. On $\mathcal{B}$ we have $\tau_{\text{train}} - |D| \geq \tau_{\text{train}} - \tau_{\text{infer}}$, hence $\mathbb{I}(|y_d| \geq \tau_{\text{sing}})\,[\text{ReLU}(\tau_{\text{train}} - |D|)]^2 \geq (\tau_{\text{train}} - \tau_{\text{infer}})^2 \cdot \mathbb{I}(\mathcal{B})$. Taking expectations and using the definition of $\mathcal{L}_{margin}$ in (11) yields (13). □

Interpretation. Proposition 3 links the deployment-relevant strict trigger ($|D| < \tau_{\text{infer}}$ on finite targets) to the training-time margin loss and makes the $\tau_{\text{train}} - \tau_{\text{infer}}$ buffer auditable. We additionally log the interval $\tau_{\text{infer}} \leq |D| < \tau_{\text{train}}$ as a diagnostic `gap_mask` (returned by strict inference when $\tau_{\text{train}} > \tau_{\text{infer}}$), but we treat it as an empirical indicator rather than a quantity bounded by Proposition 3. In the Robotics benchmark, the SCM-Basic denominator is bounded away from zero by construction (**see Appendix B**) and the stored test sets yield `gap_mask`$\equiv 0$.

Censoring separation loss (Dose). For the censored dose–response task, we replace the one-sided margin penalty $\mathcal{L}_{margin}$ with a *bidirectional* separation loss: censored targets should decode to $\perp$ (small $|D|$) while finite targets should remain safely away from the singular band (large $|D|$). We implement this as a task-specific separation objective with two thresholds:

$$\mathcal{L}_{sep} = \mathbb{E}\left[\begin{array}{l}\lambda_\perp\,\mathbb{I}(|y_d| < \tau_{\text{sing}})\,[\text{ReLU}(|D| - \tau_\perp)]^2 \\ +\,\lambda_{\text{fin}}\,\mathbb{I}(|y_d| \geq \tau_{\text{sing}})\,[\text{ReLU}(\tau_{\text{finite}} - |D|)]^2\end{array}\right], \tag{14}$$

where $\lambda_\perp$ and λ_{fin} weight the censored and finite terms, and $\tau_\perp < \tau_{\text{finite}}$ defines a "no-man's-land" to discourage ambiguous denominators near the decision boundary. This is the bidirectional analogue of the generic $\tau_{\text{train}} > \tau_{\text{infer}}$ gap: in Dose, we train censored samples toward $|D| \leq \tau_\perp$ and finite samples toward $|D| \geq \tau_{\text{finite}}$, with strict inference at τ_{infer} satisfying $\tau_\perp < \tau_{\text{infer}} < \tau_{\text{finite}}$. In domains without explicit censoring labels, we use the one-sided gap-control form (11).

The complete training and inference procedure is summarized in Algorithm 2.

4 Related Work

Totalized arithmetic and alternatives to IEEE division. Common meadows provide an algebraic semantics for division-by-zero by introducing an absorptive bottom element and total inverse (Bergstra & Ponse, 2015). Signed common meadows extend this with sign information, which motivates our orientation tracking (Bergstra & Tucker, 2024); see also the precursor signed involutive meadow formulation (Bergstra & Ponse, 2017). As a contrasting algebraic choice, wheels are another approach to division by zero that differs from meadows in axioms and intended semantics (Carlström, 2004). While these algebraic systems are well-studied, their use as inductive biases inside gradient-trained neural models—together with an explicit smooth-to-strict transition and strict decoding/coverage semantics—has received far less attention.

Rational networks and pole-aware function classes. Rational function classes have strong approximation properties and relate closely to standard neural architectures (Telgarsky, 2017). Rational neural networks (e.g., rational activations) can represent steep transitions and pole-like behavior more directly than purely piecewise-linear extrapolation, motivating our rational inductive bias in spectral and asymptotic tasks (Boullé et al., 2020). Prior work on learnable rational activations discusses numerical issues when denominators become small and motivates "safe" parameterizations; ZeroProofML instead treats $Q(x) = 0$ as a first-class semantic outcome via $\perp$ (Molina et al., 2020). More broadly, the dominant framing in rational-network work is to regularize away near-zero denominators; our goal is to make the denominator boundary semantically meaningful and to expose it as an explicit rejection state.

Scientific ML context and singular regimes. Physics-Informed Neural Networks (PINNs) are a standard approach to SciML that regularizes learning via physics residuals in the loss (Raissi et al., 2019). Conceptually, PINNs regularize the *loss landscape* via soft constraints (physics residual penalties), whereas ZeroProofML regularizes the *hypothesis class* via hard architectural/algebraic constraints (rational structure, totalized division, and strict $\perp$ decoding). These approaches are orthogonal: SCM models can be trained with PINN-style residual terms when PDE/ODE structure is available, while still retaining explicit rejection semantics at singularities.

Singularities beyond function space. Singular learning theory studies singularities in parameter space (e.g., degenerate Fisher information) and their implications for generalization (Watanabe, 2009; Wei et al., 2023). This is orthogonal to our focus on forward singularities in the learned map, but motivates reporting variance and reproducibility metrics.

Singularity detection and sharp-gradient resolution. Recent work explores neural methods for resolving sharp gradients and for detecting singularities (Wang et al., 2025; Derevianko et al., 2025), as well as applications to singular differential equations (Cayuso et al., 2025). These lines are complementary to ZeroProofML's goal of providing numerically safe singular semantics and rejection/coverage monitoring in learned models.

5 Experiments: A Singularity Spectrum

We evaluate ZeroProofML across three domains and compare it to deliberately strong baselines. In particular, we include strong over-parameterized MLP baselines (4-layer MLPs; up to 45.8× more parameters than the SCM heads in this suite, e.g., RF: 791,042 vs 17,283 parameters). That makes the comparisons conservative. Still, the capacity question is fair, so we also include *iso* baselines (MLP-Iso) with parameter counts matched to the SCM heads. Throughout, we report mean±std across 10 seeds and lean on paired, per-seed comparisons (including "10/10 seed wins") because the seed is the unit of replication.

Before the domain-specific details, here is what each benchmark is meant to stress. Together, they span a spectrum from *hard rejection semantics* (Dose), through *structural extrapolation* (RF), to *optimization stability in ill-conditioned regimes* (Robotics):

- **Dose (informational singularity):** can the model *reject* censored inputs without hallucinating finite values?
- **RF (spectral poles):** does a shared complex denominator preserve amplitude and phase under extreme Q_f extrapolation?
- **Robotics (near-singular optimization stability):** when $\perp$ never triggers, does the projective/rational parameterization still stabilize training in an ill-conditioned regime (lower seed variance)?

Wall-clock training time reporting. For Dose and Electronics (RF), "training time" is training+selection wall-clock and includes the per-epoch validation evaluation used for model selection ("restore best"). We do not try to strip that out. Robotics is a little messier: the baseline MLP and the SCM model were trained with different batch sizes (MLP: 1024; SCM: 256). So we also report normalized throughput (seconds per optimizer step, and samples/sec computed from n_{train}×epochs divided by wall-clock).

Data splits (fixed across methods). Splits are fixed *per seed* and shared across all methods within each domain. Concretely: (i) **Dose:** n_{train}=200,000, n_{val}=20,000, n_{test}=30,000; (ii) **RF:** n_{train}=4096 filters, n_{val}=1024, n_{test}=2048; (iii) **Robotics:** n_{total}=200,000 samples with a fixed split n_{train}=160,000, n_{test}=40,000 on a holdout-B0 dataset file (recorded and hashed in each seed's `comprehensive_comparison.json`).

Hyperparameters and selection protocol. Within each domain we keep the training recipe fixed (epochs, learning rate, optimizer, and selection rule) *within each reported suite* and change only the model class. Dose includes a 10-seed reference suite (30 epochs) plus targeted angular/curriculum follow-ups (50 epochs); those follow-ups use the same data generator and evaluation protocol, but intentionally modify the training schedule to study direction–regression effects. For Dose and RF we use `restore-best` based on a validation metric computed each epoch (and we include that validation evaluation in the wall-clock timings). For Dose, the validation score prioritizes regime identification and then conditional regression (score $= (1 - \text{macroF1}) + 0.05 \cdot \min(20, \text{finiteMAE})$); for RF it is validation MSE. Importantly, we do not tune hyperparameters separately for each method in the reported 10-seed suite. The goal here is ceteris-paribus comparison within suites, not a best-possible leaderboard. Fixed settings are listed in Table 17 (Appendix C).

Statistical reporting. The unit of replication is the random seed (n=10). We report mean±std across seeds. For headline claims we also report paired summaries: a paired bootstrap 95% confidence interval for the mean difference (20,000 resamples over seeds), a paired Wilcoxon signed-rank test (two-sided; exact; zeros excluded), and an exact paired sign test. When we write "10/10 seed wins," we treat it as an empirical win-rate with a Clopper–Pearson 95% interval (for 10/10: $[0.692, 1.0]$). Figures show per-seed distributions (violin+box + seed points). Full outputs are stored under `derived/paired_stats.json` in the reference run directory (and summarized in Table 12 in Appendix A).

Metrics and $\perp$ handling (global policy). We fix one evaluation policy per domain and keep it the same across methods: (i) **Dose:** we treat evaluation as two objectives: (A) *safety* (reject censored; minimize FP_{cens} and FN_{in}) and (B) *directional utility* (if censored, classify below vs. above; summarized by per-class F1 and Macro-F1). For SCM models, a bottom decode ($|D| < \tau_{\text{infer}}$) maps to *censored*; we resolve below vs. above by the numerator sign (Decoding spec in §3). For *finite MAE*, we evaluate only on samples that are (true in-range) $\cap$ (predicted in-range); bottom outputs are treated as out-of-range and excluded. (ii) **RF:** peak ratio and yield are computed from each test filter's response on a dense local grid around the oracle peak. Any invalid numerical output is treated as a failure for yield; in our reference suite, the strict bottom mask did not trigger on the RF test sets (valid rate = 1.0), so phase errors are reported over the full test set. (iii) **Robotics:** all reported metrics are finite-valued regressions against the damped least squares (DLS) oracle. For SCM-Basic, D is bounded away from zero by construction, so $\perp$ does not occur; our evidence there is about optimization stability and seed variance, not rejection semantics.

5.1 Domain A: Pharma/Health (Dose; two objectives: safety and direction)

Task: Given noisy Hill curves sampled over a finite assay window $[c_{\min}, c_{\max}]$, predict $\log_{10}(\text{IC50})$ when identifiable and output a *censored* decision otherwise. Censored targets are encoded as projective points $\langle \pm 1, 0 \rangle$ and must decode to a strict bottom state ($\perp$).

In drug-discovery pipelines, IC50 estimates feed directly into therapeutic-index calculations and candidate-selection decisions. A model that silently returns a finite surrogate for a non-identifiable IC50 can propagate a confidently wrong potency estimate through downstream stages—potentially advancing the wrong compound or miscalculating a safety margin.

Evaluation protocol (two objectives). Primary metric: FP_{cens} (safety; reject censored). **Secondary metrics:** FN_{in}, censored-direction F1 (below vs. above among true censored points; Figure 3b), and finite MAE computed on accepted points.

Results: This is the most direct test of the paper's core idea: censoring boundaries should be treated as algebraic outcomes ($\perp$), not "very large numbers." It is also explicitly a two-objective benchmark: we want *fail-safe rejection* on censored inputs (safety) *and* correct below/above direction when rejection occurs (utility).

Recommended configurations (Pareto frontier). The strict SCM semantics are the same in all variants ($|D| < \tau_{\text{infer}} \Rightarrow \perp$); what changes is the parameterization and training schedule. We recommend two operating points (Table 1): (i) **Default (gating + regression):** Angular SCM with a gentle separation curriculum (50 epochs; $\lambda_\perp$ ramps to 10). It yields symmetric gating with $\text{FP}_{\text{cens}} = 1.24 \times 10^{-4} \pm 2.95 \times 10^{-4}$ and $\text{FN}_{\text{in}} = 0.00 \pm 0.00$, and strong conditional regression with finite MAE 0.0765 ± 0.00684. (ii) **Direction-aware:** Angular SCM with a harder curriculum (50 epochs; $\lambda_\perp$ ramps to 20), which recovers full 3-way regime identification (Macro-F1 0.904 ± 0.0013) at a regression cost (finite MAE 0.393 ± 0.259).

For scale, each seed's test split has $n_{\text{test}} = 30{,}000$ samples. Aggregating across the 10-seed soft-curriculum angular run, the model makes 26 false in-range predictions out of 209,974 censored test points, and zero false censorings out of 90,026 in-range test points. In contrast, Rational+ϵ has no way to express censoring as a first-class state and hallucinates in-range predictions on $57.3\% \pm 0.28\%$ of censored inputs.

The rejection mechanism is not uncertainty-based: the denominator magnitude $|D(\mathbf{x})|$ is trained as an identifiability gate in the projective encoding (finite vs. censored), not as a confidence score. A perfectly confident model would still decode to $\perp$ on a censored input, because censoring is a property of the measurement window rather than the model's epistemic state.

Macro-F1 is informative but not the headline safety metric: it mixes (i) in-range identification with (ii) below/above direction among censored points. Under the default operating point, direction can still collapse on one censored direction under the tuple-only sign rule, limiting macro-F1 despite near-perfect safety; the hard curriculum shows that this is not an architectural limitation but a training-ordering effect (Figure 3b).

TwoStage serves as an explicit predict-plus-reject baseline: it treats censoring as a separate discrete decision and then performs conditional regression on the finite subset. This is an excellent solution when censoring labels are available and the censoring mechanism matches training and test (Table 1).

The paired statistics line up with the per-seed story. Using FP_{cens} (false in-range on true censored), the paired mean reduction (Rational+$\epsilon \rightarrow$ Angular SCM soft curriculum) is 0.5730 with a 95% bootstrap CI $[0.5713, 0.5747]$; Wilcoxon $p = 1.95 \times 10^{-3}$ and sign-test $p = 1.95 \times 10^{-3}$ (10/10 seed wins; win-rate CI $[0.692, 1.0]$).

On capacity control: `TwoStage_Iso` matches the strong TwoStage baseline macro-F1, so high boundary-classification performance here is not simply "more parameters wins."

A pragmatic alternative is to decouple censored direction from the tuple while keeping the accept/reject gate strict. We include a frozen-head protocol over all 10 seeds: for each seed we freeze the trained strict-SCM tuple model and train only a tiny 2-way direction head on censored samples. On the true-censored test subset, this raises the censored-direction Macro-F1 from 0.333 ± 0.033 (tuple-only sign rule) to 0.856 ± 0.0018 (direction head), without changing the accept/reject gate (since (P, Q) and the $|Q| < \tau_{\text{infer}}$ check are unchanged). See Appendix Table 8 (artifacts under `results/paper_suite_dose/run_dose_strict_ckpt_10seed/`).

Table 1: **Domain A (Dose): main results (10 seeds).** This benchmark is explicitly two-objective: (A) *safety* (reject censored; minimize $\mathrm{FP}_{\mathrm{cens}}$) and (B) *directional utility* (if censored, distinguish below vs. above; summarized by Macro-F1 / per-class F1). TwoStage is an explicit predict-plus-reject baseline (censoring classifier plus conditional regressor). We report FP/FN as *class-conditional* rates: $\mathrm{FP}_{\mathrm{cens}} := \#(\text{true censored} \wedge \text{pred in-range})/\#(\text{true censored})$ and $\mathrm{FN}_{\mathrm{in}} := \#(\text{true in-range} \wedge \text{pred censored})/\#(\text{true in-range})$. "Binary regime BA" is $\frac{1}{2}\big((1-\mathrm{FP}_{\mathrm{cens}}) + (1-\mathrm{FN}_{\mathrm{in}})\big)$. Finite MAE is computed only on (true in-range) $\cap$ (predicted in-range). Angular SCM rows are targeted follow-ups under the same data generator and evaluation protocol, using the angular parameterization and curriculum schedules described in §3.

Method	Macro-F1↑	Regime BA↑	$\mathrm{FP}_{\mathrm{cens}}$ ↓	$\mathrm{FN}_{\mathrm{in}}$ ↓	$\mathrm{MAE}_{\mathrm{finite}}$ ↓
TwoStage	0.904 ± 0.00122	1.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.0498 ± 0.00157
TwoStage-Iso	0.904 ± 0.00122	1.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.0510 ± 0.00104
Rational+ϵ	0.485 ± 0.000851	0.713 ± 0.00141	0.573 ± 0.00282	0.00 ± 0.00	0.148 ± 0.0800
Angular SCM variants (strict decoding).					
Angular SCM (soft curriculum; default)	0.576 ± 0.000839	1.00 ± 0.00	$1.24 \times 10^{-4} \pm 2.95 \times 10^{-4}$	0.00 ± 0.00	0.0765 ± 0.00684
Angular SCM (hard curriculum; direction)	0.904 ± 0.00132	1.00 ± 0.00	$3.71 \times 10^{-4} \pm 6.11 \times 10^{-4}$	0.00 ± 0.00	0.393 ± 0.259

In contrast, ZeroProofML targets a different desideratum: it makes censoring an algebraic outcome of the same projective object (via $|D(\mathbf{x})| < \tau_{\mathrm{infer}} \Rightarrow \perp$), aligning rejection with the learned singular structure and enabling downstream use without bespoke conditional logic at every call site.

Composability stress test (post-hoc). This is where SCM semantics are meant to matter. In practice, dose models are rarely used alone; they feed into downstream calculations. A simple example is a therapeutic index proxy, $\log_{10}(\mathrm{TI}) = \log_{10}(\mathrm{LD50}) - \log_{10}(\mathrm{IC50})$. For censored inputs, $\log_{10}(\mathrm{IC50})$ is non-identifiable, so the composed quantity should be treated as undefined.

We quantify a "silent failure" rate: among truly censored test points, how often does the pipeline still produce a finite-looking scalar? (This is *conditional* on the censored subset; it matches $\mathrm{FP}_{\mathrm{cens}}$ in Table 1.) There are two integration patterns. *Guarded* means the caller checks a censoring decision and rejects; *naive* means the caller ignores censoring and uses the scalar output everywhere.

One subtlety: some models do not emit an explicit censoring decision. For TwoStage, the decision is the 3-way classifier output. For SCM variants, the decision is the bottom mask ($|Q| < \tau_{\mathrm{infer}}$). Rational+$\epsilon$ is scalar-only, so the only available "decision" is to *derive* a regime label by thresholding the scalar (the same rule used in the main 3-class evaluation). Put differently, the Rational+ϵ guarded rate is just the FP rate reformulated as a conditional probability on censored points.

Strict SCM decoding avoids a common integration failure mode because censored predictions decode to $\perp$ (represented as NaN in our implementation), which then propagates through arithmetic by default. (We still recommend treating `bottom_mask` as the authoritative carrier of the $\perp$ semantics; NaN is only a convenient payload.) In our 10-seed angular soft-curriculum run, this yields 26 silent finite outputs out of 209,974 censored test points for Angular SCM, versus 209,974/209,974 under naive composition for scalar-only models (by construction). Table 2 reports the corresponding rates.

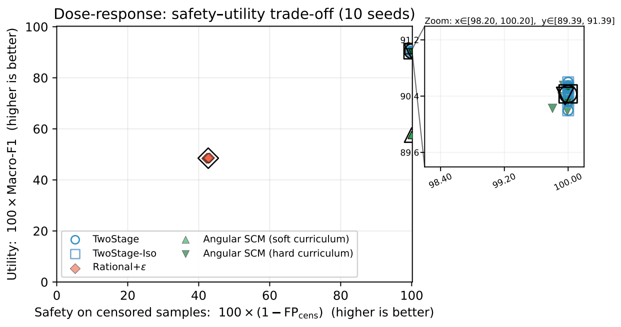
This advantage is not about better classification, it is about making failure hard to ignore in downstream arithmetic. A simple multi-step risk model (thought experiment) is provided in Appendix A (Table 9), but the one-step composition in Table 2 is the empirical point.

5.2 Domain B: Electronics (Spectral Singularity)

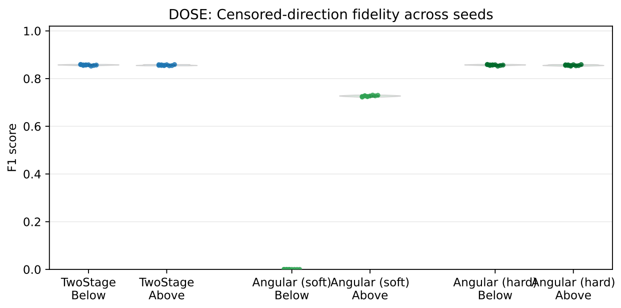
Task: Extrapolate RF bandpass filter behavior from low-Q_f training data ($Q_f \leq 30$) to high-Q_f resonances ($Q_f \leq 1000$, 33× OOD).

Table 2: **Dose composability stress test (one-step; derived; 10 seeds).** "Silent finite on censored" is the conditional rate Pr[pipeline outputs a finite-looking scalar | true sample is censored] for a one-step downstream computation that consumes the predicted $\log_{10}$(IC50) (e.g., a naive TI computation). "Guarded" uses the model's censoring decision (TwoStage logits; SCM bottom mask). For Rational+ϵ (scalar-only), the decision is derived by thresholding the scalar (same rule used for 3-class evaluation), so the guarded rate equals conditional FP. "Naive" ignores censoring and always uses the scalar; for scalar-only models this yields 1.0 by construction. Derived from stored seed metrics plus deterministic test-set stratification; see `derived/dose_composability.json`.

Method	Guarded silent finite↓	Naive TI silent finite↓
TwoStage	0.00 ± 0.00	1.00 ± 0.00
TwoStage-Iso	0.00 ± 0.00	1.00 ± 0.00
Rational+ϵ	0.573 ± 0.00282	1.00 ± 0.00
Angular SCM (soft curriculum)	$1.24 \times 10^{-4} \pm 2.95 \times 10^{-4}$	$1.24 \times 10^{-4} \pm 2.95 \times 10^{-4}$



(a) Safety–utility scatter (10 seeds; points are per-seed; zoom panel shown at right; rectangle marks the zoomed region).



(b) Per-class F1 reveals one-sided collapse (soft curriculum: Below near-zero).

Figure 3: **Dose experiment: class-conditional safety vs. directional utility.** Left: class-conditional safety on censored samples, $100 \times (1 - \mathrm{FP}_{\mathrm{cens}})$ (Table 1), plotted against $100\times$ Macro-F1 (10 seeds). Rational+ϵ has low safety due to frequent finite hallucinations on censored inputs, whereas strict SCM decoding yields high safety. Right: per-class F1 shows that tuple-only direction can still collapse on one censored direction (below vs. above), limiting macro-F1; a harder curriculum schedule can recover direction (with a regression trade-off; Table 1).

High-Q_f resonators appear throughout RF and microwave engineering—from narrowband receivers to radar front-ends—and characterizing their near-peak behavior typically requires expensive full-electromagnetic simulation or physical prototyping. A surrogate that can extrapolate reliably from moderate-Q_f training data to high-Q_f regimes would reduce the number of costly simulations needed during design-space exploration.

Results: Here the architectural point is simple: we tie the real and imaginary parts together with a shared complex denominator $Q(s)$ (here Q denotes the denominator polynomial, distinct from the quality factor Q_f). The hope is that this enforces phase coherence rather than letting the model fit amplitude with inconsistent poles.

Empirically, the yield story is the headline. Success rate (Peak Ratio > 0.9) goes from $39.8\% \pm 4.83\%$ (MLP-Deep) to $77.3\% \pm 15.4\%$ (ZeroProofML-SCM), and the SCM model beats the MLP in 10/10 seeds. The paired mean yield gain is $+37.51$ percentage points with 95% bootstrap CI $[29.66, 45.62]$; Wilcoxon $p = 1.95 \times 10^{-3}$ and sign-test $p = 1.95 \times 10^{-3}$ (10/10 seed wins; win-rate CI $[0.692, 1.0]$).

Peak retention tells the same story from another angle. The MLP clips resonance peaks (median peak ratio 0.638 ± 0.111), while ZeroProofML-SCM retains peak magnitude (0.977 ± 0.0256), again with 10/10 seed wins.

Table 3: **Domain B (Electronics/RF): main results (10 seeds).** p50 denotes the median over test filters (computed per seed). Phase error is measured at each test filter's oracle peak frequency and reported modulo π (global sign ambiguity under log-magnitude supervision).

Method	Peak ratio p50↑	Yield (%)↑	Phase err p50 (deg)↓
Oracle	1 ± 0	100 ± 0	0 ± 0
MLP (Deep)	0.638 ± 0.111	39.8 ± 4.83	52.8 ± 16.2
MLP-Iso (capacity-matched)	0.274 ± 0.0680	12.5 ± 8.0	44.4 ± 11.2
Rational-only (non-SCM)	0.902 ± 0.151	54.0 ± 9.85	42.5 ± 13.9
SCM (decoupled)	0.363 ± 0.147	34.6 ± 5.00	28.0 ± 27.7
ZeroProofML-SCM (shared denom)	0.977 ± 0.0256	77.3 ± 15.4	12.1 ± 10.7

Paired statistics: peak-ratio p50 (median) gain +0.339 (95% CI $[0.278, 0.390]$); phase-error p50 improvement 40.7° (95% CI $[26.8°, 53.2°]$); Wilcoxon $p = 1.95 \times 10^{-3}$ and sign-test $p = 1.95 \times 10^{-3}$ for both.

The baselines help separate "rational" from "SCM." A non-SCM rational transfer model (RationalTransfer) lands in between: yield $54.0\% \pm 9.85\%$ and peak ratio 0.902 ± 0.151 (p50). Meanwhile, the capacity-matched MLP-Iso baseline performs much worse (yield $12.5\% \pm 8.0\%$), so the improvements are not explained by parameter count alone.

The decoupled SCM ablation is the most telling sanity check. It keeps "rational" but drops the shared complex denominator, and performance degrades sharply (yield $34.6\% \pm 5.00\%$ vs. $77.3\% \pm 15.4\%$ for the shared-denominator model). That is exactly what we would expect if the shared $Q(s)$ constraint is doing real work: without it, the model can fit amplitude while drifting in pole structure, and coherence falls apart.

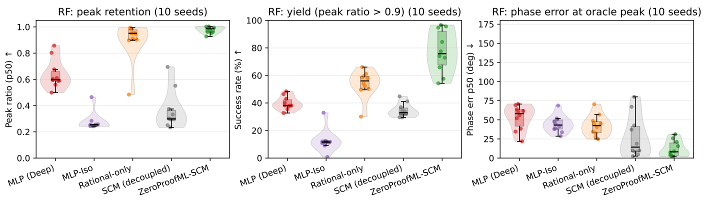
Finally, on phase: measured at each test filter's oracle peak frequency, the median absolute phase error (modulo π, reflecting global sign ambiguity under log-magnitude supervision) is $12.1 \pm 10.7°$ for ZeroProofML-SCM vs $52.8 \pm 16.2°$ for the MLP. This also improves in 10/10 seeds. Put differently, the shared-denominator constraint shows up where we expected it to: in coherence, not just amplitude.

5.3 Domain C: Robotics (Near-Singular Optimization Stability)

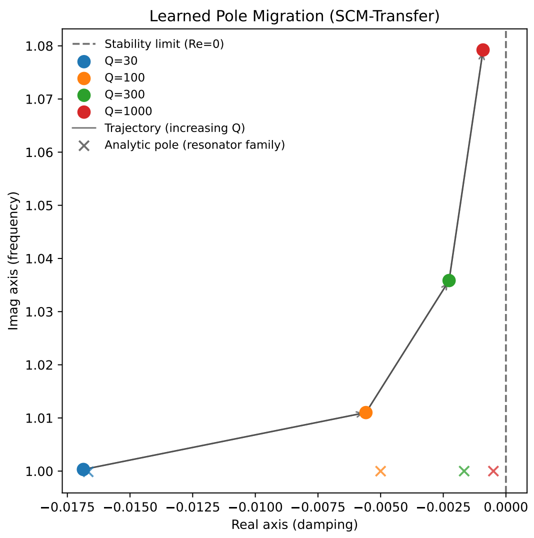
Task: Inverse Kinematics (IK) for a 2R planar arm near full extension ($\det(J) \to 0$). Near singularity, the Jacobian becomes ill-conditioned and small perturbations can cause large changes in pseudo-inverse solutions, a phenomenon commonly analyzed via singular values and condition numbers (Trefethen & Bau, 1997). For deployed manipulators, this regime is not hypothetical: it arises whenever an arm approaches full extension or a workspace boundary. A learned IK solver whose behavior varies substantially across random initializations is difficult to certify, because worst-case guarantees must account for whichever seed was used in production.

What this domain tests (and what it does not). In this domain, the denominator is bounded away from zero by construction, so strict inference produces no $\perp$ decodes (Appendix B). The contribution here is therefore not rejection semantics but optimization geometry: the projective–rational parameterization improves training stability in the near-singular regime where conventional approaches exhibit high seed-to-seed variance. We include this domain precisely because it demonstrates that the framework's benefits extend beyond hard singularity detection.

Results: Precision–Consistency with tail-statistic fidelity. While the unconstrained MLP achieves lower mean MSE (2.28×10^{-4} vs 3.67×10^{-4}), it also shows much larger seed-to-seed variability (variance across seeds of the per-seed test MSE vs. the oracle). That matters if you care about repeatability under random initialization. In safety-critical robotics, we think that variability is part of the story. The 31.8× reduction in seed variance provided by the SCM model suggests a *safety–consistency* trade-off: we give up some average-case precision in exchange for tighter, more predictable behavior across seeds and in tail statistics.



(a) Per-seed metrics (peak retention, yield, phase error).



(b) Learned pole migration (ZeroProofML-SCM, shared denom; seed 10). Circles: learned dominant stable pole; crosses: analytic resonator-family poles; dashed line: stability limit (Re=0). Arrows indicate increasing Q.

Figure 4: **RF experiment: peak retention and phase coherence.** The shared-denominator SCM model preserves resonance amplitude and phase under 33× OOD extrapolation. The decoupled SCM ablation degrades sharply (yield 34.6% vs. 77.3% for shared-denominator), which is the point: without a shared complex denominator $Q(s)$, independent rational heads can fit amplitude while drifting in pole structure, and coherence breaks down.

Bias–variance trade-off (Table 4). We observe a structural bias–variance trade-off: the unconstrained MLP minimizes bias (lower mean MSE) but exhibits high seed-to-seed variance near the kinematic lock, whereas the SCM model accepts higher bias to achieve substantially lower variance. This distinction matters in safety-relevant regimes, where average-case error can be a poor proxy for reproducibility and tail behavior. Extended capacity baselines and auxiliary loss ablations are provided in Tables 10 and 11 in Appendix A. First, the variance effect is real and large: the SCM model's seed variance ($\sigma^2 \approx 9.79 \times 10^{-10}$) is 31.8× lower than the MLP's ($\sigma^2 \approx 3.11 \times 10^{-8}$). That is the result we think a reviewer should take seriously if they care about reproducibility.

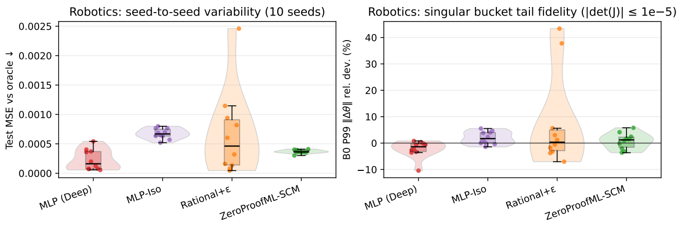
The iso baseline helps contextualize it. With strict capacity matching (MLP-Iso at 4,747 parameters), mean error is much worse ($(6.72 \pm 0.896) \times 10^{-4}$), even though its variance is lower than the over-parameterized MLP ($\sigma^2 \approx 8.02 \times 10^{-9}$). So capacity control does not make the MLP "catch up" on mean error here.

We also include a rational-only (non-SCM) baseline with ϵ-regularized decoding (e.g., decode $\frac{P}{Q+\epsilon}$) to contrast standard regularized rational evaluation against the SCM projective and totalized training setup under the same data, splits, and budget. The projective rational+ϵ model ("Rational+ϵ") is extremely unstable across seeds (mean test MSE 6.71×10^{-4} with $\sigma = 7.41 \times 10^{-4}$; $\sigma^2 \approx 5.49 \times 10^{-7}$) and it performs poorly on the most singular bucket ($+7.47\% \pm 17.86\%$ deviation). The difference is the arithmetic and parameterization choice (totalized versus ϵ-regularized) and the resulting optimization stability.

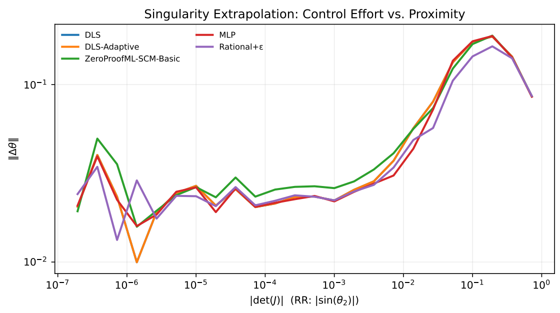
On tail fidelity: in the most singular bucket ($|\det J| \leq 10^{-5}$), the MLP under-reacts relative to the oracle (B0 P99 step magnitude $-2.24\% \pm 3.27\%$). The SCM model shifts that bias in the opposite direction with smaller magnitude ($+0.75\% \pm 3.05\%$). We are careful about the interpretation here: absolute tail deviation is not significantly different across seeds, so we read this mainly as reduced systematic attenuation rather than a blanket improvement in tail accuracy.

Table 4: **Domain C (Robotics/IK): main results (10 seeds).** "Var (σ^2)" denotes the variance across seeds of the per-seed test MSE vs. the oracle. B0 tail fidelity reports the P99 control-magnitude deviation in the most singular bucket ($|\det(J)| \le 10^{-5}$). Rational+ϵ is included ceteris paribus to isolate totalized and projective arithmetic effects (SCM-Basic does not decode $\perp$ in this benchmark). MSE is nonnegative; across seeds it can be skewed/heavy-tailed, hence std can exceed the mean.

Method	MSE↓	Var (σ^2)↓	B0 P99 $\|\Delta\theta\|$ rel. dev. (%)
MLP	$(2.28 \pm 1.76) \times 10^{-4}$	3.11×10^{-8}	-2.24 ± 3.27
MLP-Iso (capacity-matched)	$(6.72 \pm 0.896) \times 10^{-4}$	8.02×10^{-9}	$+1.98 \pm 2.42$
Rational+ϵ	$(6.71 \pm 7.41) \times 10^{-4}$	5.49×10^{-7}	$+7.47 \pm 17.86$
ZeroProofML-SCM-Basic	$(3.67 \pm 0.313) \times 10^{-4}$	9.79×10^{-10}	$+0.754 \pm 3.05$



(a) Seed-to-seed variability and tail fidelity (B0: $|\det(J)| \le 10^{-5}$; P99 rel. dev.: relative deviation in P99 $\|\Delta\theta\|$ vs. DLS).



(b) Control effort vs. proximity (seed 10).

Figure 5: **Robotics experiment: precision–consistency and singular-limit behavior.** The SCM model is more consistent across seeds and more faithful in singular-bucket tail statistics. The binned closed-loop control-effort profiles remain comparable to the DLS and DLS-Adaptive references and do not exhibit runaway behavior near singularity.

We treat the DLS oracle as the reference law and quantify near-singularity behavior via tail magnitude statistics (B0 P99 deviation) and the binned control-effort profile (Figure 5b). We do *not* claim that lower MSE implies unphysical behavior. The more modest claim is that average-case MSE can hide tail deviations that matter in near-singular regimes.

One more thing we did not want to leave as "just a bound." For the train-on-smooth / infer-on-strict protocol, we also provide a deterministic test-set certificate: in Robotics, strict SCM inference produces zero bottom decodes on the test sets, and the stored verification reports 0/800,000 denominator entries inside the $[\tau_{\text{infer}}, \tau_{\text{train}})$ gap across seeds (**see Appendix B**).

Finally, a compute note. Batch sizes were varied to keep projective optimization stable (MLP: 1024; SCM: 256), so the raw wall-clock is not perfectly apples-to-apples. Still, the normalized throughput is comparable in samples/s: SCM reaches $(6.31 \pm 0.0531) \times 10^4$ samples/s, compared to $(4.63 \pm 0.192) \times 10^4$ for the MLP. In practice, this suggests that the algebraic overhead of SCM is not a prohibitive training penalty in this setting.

6 Discussion and Limitations

Constraints are not free. The robotics results make clear that ZeroProofML is not a drop-in improvement. Restricting the hypothesis class to rational functions buys you consistency (the 31.8× variance reduction is striking) but costs you mean accuracy. For a safety-critical deployment where you need to *trust* that the controller will not behave wildly across initializations, that is a good deal. For a setting where average-case error dominates, it may not be.

Censoring works; direction–regression is the remaining tension. The dose–response benchmark asks two different things: (i) the safety question (reject censored inputs; avoid finite hallucinations), and (ii) the utility question (if censored, determine below-range vs. above-range). On this supervised fixed-window task, explicit censoring classifiers (TwoStage) provide a strong ceiling; SCM is motivated by representing censoring as a first-class algebraic value ($\perp$) intended to remain meaningful under downstream composition.

The new angular variants show that *direction is solvable*, but not free. The soft-curriculum angular operating point yields symmetric gating with near-zero $\mathrm{FP}_{\mathrm{cens}}$ and $\mathrm{FN}_{\mathrm{in}}$=0 while achieving strong finite regression (MAE ≈ 0.076), but tuple-only direction can still collapse (Macro-F1 ≈ 0.576). A harder curriculum recovers full 3-way regime identification (Macro-F1 ≈ 0.904), matching TwoStage, but degrades finite MAE substantially (Table 1; see Appendix Figure 6). Put differently: the framework can say "this is undefined" very reliably, and it can learn direction when trained in the right order, but reconciling direction with high-quality regression remains an open optimization/scheduling problem under our current losses.

Why SCM is not just abstention thresholding. At a glance, strict decoding resembles abstention: "if $|Q|$ is small, reject." The resemblance is superficial.

(1) $\perp$ is a value, not a policy. Selective prediction (El-Yaniv & Wiener, 2010) returns a real-valued estimate plus an uncertainty score, and an external policy decides whether to reject. SCM instead outputs $\perp$ as an algebraic value. Under strict decoding we implement this as a dual carrier (`value=NaN` plus `bottom_mask=True`): arithmetic pipelines that ignore the mask still see NaNs by default, and pipelines that clean NaNs can still retain the rejection semantics via the mask.

(2) $\perp$ composes without additional machinery. Uncertainty is local and does not propagate through a pipeline unless it is re-estimated (or explicitly propagated) at each stage. $\perp$ propagates through arithmetic by the meadow axioms (e.g., $\perp + x = \perp$), and the `bottom_mask` survives silent NaN replacement. Table 2 shows the consequence under naive one-step downstream arithmetic: scalar-only models are *always* finite on censored points by construction, while strict SCM is almost never finite because $\perp$ is embedded (NaN payload) and propagates.

(3) The rejection reason differs. Selective prediction rejects because the model is uncertain (epistemic/aleatoric). SCM rejects because the quantity is non-identifiable under the measurement protocol (e.g., assay-window censoring); Proposition 3 bounds the probability of triggering strict $\perp$ decoding on finite targets, and we additionally log a diagnostic `gap_mask` for $\tau_{\text{infer}} \leq |D| < \tau_{\text{train}}$.

On a fixed, single-step benchmark with known censoring labels, the practical difference between strict SCM rejection and a well-tuned abstention policy can be modest. The distinction becomes consequential in multi-step pipelines, in settings where censoring labels are unavailable at test time, and when downstream code cannot be trusted to propagate rejection flags consistently. Empirical evaluation under multi-step real-world pipelines is an important direction for future work.

Strict-boundary control. Proposition 3 gives a distributional bound linking the probability of triggering strict $\perp$ decoding on finite targets ($|D| < \tau_{\text{infer}}$) to the margin loss. We log the τ_{infer}–τ_{train} interval as an empirical `gap_mask` diagnostic. In robotics, the denominator stays bounded away from zero by construction and we provide a deterministic test-set certificate. For domains with explicit singular targets, a microbenchmark confirms that $\mathcal{L}_{margin}$ and $\mathcal{L}_{sign}$ do produce non-trivial gradients near poles. Details are in **Appendix B**.

7 Conclusion

The core idea behind ZeroProofML is simple: if your problem has singularities, your arithmetic should be able to represent them, not sweep them under a regularization constant. Signed Common Meadows give us that capability, and the Train on Smooth, Infer on Strict protocol makes it compatible with gradient-based optimization.

Across three domains, the framework does what we hoped in some respects and falls short in others. In Dose, strict $\perp$ decoding with an angular parameterization yields symmetric gating with near-zero false in-range on censored inputs ($\mathrm{FP}_{\mathrm{cens}} \approx 10^{-4}$; $\mathrm{FN}_{\mathrm{in}} = 0$), while a harder curriculum recovers direction (Macro-F1 ≈ 0.904)

Table 5: **Conceptual comparison: "large number" vs. undefined vs. reject option.** This table is about semantics and failure modes, not benchmark performance.

Approach	Near $Q \approx 0$	Composable?	Typical failure mode
ϵ-regularization ($P/(Q+\epsilon)$)	returns large but finite values	No (policy must guess)	"semantic hallucination": extreme values can be regularization artifacts
IEEE Inf/NaN	returns Inf/NaN/flags (env-dependent)	Fragile—NaN propagates through arithmetic but is routinely replaced (`nan_to_num`, `dropna`, default fills)	NaN provenance is lost—downstream cannot distinguish singularity from overflow or bug; cleanup code silently replaces NaN with finite defaults
Selective prediction (ensemble, MC dropout, conformal)	returns (value, uncertainty); policy rejects high-uncertainty points. Prediction exists whether or not rejected.	Policy-level only—uncertainty score does not propagate through downstream arithmetic; must re-estimate at every pipeline stage	rejection skipped at pipeline boundaries; confident-but-wrong extrapolation in singular regimes; uncertainty $\neq$ non-identifiability
SCM strict decoding ($\perp$)	returns $\perp$ (no real-valued prediction) as an algebraic value	Yes—dual-carrier (`bottom_mask` + NaN payload) survives silent NaN replacement	orientation/direction learning can still collapse (Dose), but rejection is explicit

at a regression cost. In RF, the rational parameterization substantially outperforms MLPs on spectral peak retention under extreme extrapolation. The robotics results are more nuanced: SCM delivers much tighter seed consistency, but at the cost of mean precision relative to an unconstrained baseline.

The key finding from these experiments is not that SCM should replace standard arithmetic everywhere—it should not—but that the choice of arithmetic and decoding semantics is a genuine modeling decision, one that matters most in regimes where singular-limit behavior and safe rejection are not optional extras but core requirements. Drug-discovery pipelines that compose potency estimates, RF design workflows that extrapolate to untested resonator regimes, and robotic controllers that must behave predictably near workspace boundaries are all settings where the cost of a silently wrong number can exceed the cost of an explicit "I don't know."

Reproducibility Statement

Code is available at `[Anonymized for Review]`. RF and robotics experiments run via the `scripts/paper_vps` suite; the dose experiment via `scripts/run_paper_suite_dose.py`. The reference suite directory (including `derived/paired_stats.json`, `derived/dose_composability.json`, and other machine-readable verification artifacts) is `results/paper_suite_ps2/run_ps2_iso_10seed/`. Dose baseline artifacts are under `results/paper_suite_dose/run_dose_final_cosine_lb20_lf1_ls10_final/`; angular curriculum follow-ups are under `results/paper_suite_dose/nextsteps_angular_soft_curriculum_ext_10seed_20260215_213849/full/angular_soft_curriculum_ext/` and `results/paper_suite_dose/nextsteps_angular_curriculum_50_10seed_20260215_231921/full/angular_curriculum_50/`. We also provide a number-level claim audit (`scripts/paper_vps/claim_audit.py`) that checks the paper's headline tables against stored run artifacts. Run taxonomy and provenance details are in `experiments.md` and Appendix Tables 14–18.

Broader Impact Statement

This work focuses on architectural inductive biases for scientific machine learning. While the adoption of explicit singularity handling can improve the safety, predictability, and reliability of neural surrogates in physical regimes, it does not pose direct, foreseeable negative societal impacts.

References

Jan A. Bergstra and Alban Ponse. Division by zero in common meadows. In *Software, Services, and Systems: Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering*, volume 8950 of *Lecture Notes in Computer Science*, pp. 46–61. Springer, 2015. doi: 10.1007/978-3-319-15545-6_6.

Jan A. Bergstra and Alban Ponse. Probability functions in the context of signed involutive meadows (extended abstract). In *Recent Trends in Algebraic Development Techniques: 23rd International Workshop, WADT 2016*, volume 10644 of *Lecture Notes in Computer Science*, pp. 73–87. Springer, 2017. doi: 10.1007/978-3-319-72044-9_6.

Jan A. Bergstra and John V. Tucker. On the axioms of common meadows: Fracterm calculus, flattening and incompleteness. *The Computer Journal*, 66(7):1565–1572, 2023. doi: 10.1093/comjnl/bxac026.

Jan A. Bergstra and John V. Tucker. Fracterm calculus for signed common meadows. *Transmathematica*, 2024. doi: 10.36285/tm.97.

Jan A. Bergstra and John V. Tucker. A complete finite axiomatisation of the equational theory of common meadows. *ACM Transactions on Computational Logic*, 26(1):1–28, 2025. doi: 10.1145/3689211.

Nicolas Boullé, Yuji Nakatsukasa, and Alex Townsend. Rational neural networks. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 14243–14253, 2020.

Jesper Carlström. Wheels — on division by zero. *Mathematical Structures in Computer Science*, 14(1): 143–184, 2004. doi: 10.1017/S0960129503004110.

Ramiro Cayuso, Mario Herrero-Valea, and Enrico Barausse. Deep learning solutions to singular ordinary differential equations: From special functions to spherical accretion. *Physical Review D*, 111(6):064082, 2025. doi: 10.1103/PhysRevD.111.064082. arXiv:2409.20150.

Nadiia Derevianko, Ioannis G. Kevrekidis, and Felix Dietrich. Neural network-based singularity detection and applications. *arXiv preprint arXiv:2509.10110*, 2025.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5):1605–1641, 2010. URL https://www.jmlr.org/papers/v11/el-yaniv10a.html.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002. ISBN 0-89871-521-0. doi: 10.1137/1.9780898718027.

Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Padé activation units: End-to-end learning of flexible activation functions in deep networks. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=BJlBSkHtDS.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Matus Telgarsky. Neural networks and rational functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3387–3393. PMLR, 2017.

Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997. ISBN 978-0-89871-361-9.

Yongji Wang, Tristan Léger, Ching-Yao Lai, and Tristan Buckmaster. Resolving sharp gradients of unstable singularities to machine precision via neural networks. *arXiv preprint arXiv:2511.22819*, 2025.

Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Number 25 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2009. doi: 10.1017/CBO97805 11800474.

Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella. Deep learning is singular, and that's good. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9253–9265, 2023. doi: 10.1109/TNNLS.2022.3167409.

A Extended Tables

Dose SCM ablations

Dose model variants (for reproducibility). Table 6 reports controlled ablations of the SCM DOSE model: **(i) detachment:** `NoDetachRenorm` sets `detach_renorm=False` (allow gradients through the tuple renormalization), while `NoDetachScale` sets `detach_scale=False` in the implicit fit loss (9); **(ii) losses:** `NoSign` sets λ_{sign}=0 and `NoMargin` disables censoring-separation losses (sets $\lambda_{\perp}$=λ_{fin}=0); **(iii) initialization/parameterization:** `NoShock` removes the small-denominator initialization bias on the Q-head, and `QAnchorOnes` fixes the Q-head anchor to ones; **(iv) fit objective:** `DecodedMSE` replaces the implicit projective fit loss with a decoded MSE on finite samples plus a pole penalty on censored samples; **(v) training data balance:** `DirBalanceSampler` applies a direction-balanced sampler on the training split (balances censored class 0 vs. class 2 while preserving the overall in-range frequency).

Table 6: **Dose SCM ablations (10 seeds).** These variants isolate which SCM components control boundary safety and which affect direction/orientation. Here FP_{test} and FN_{test} are *test-normalized* rates (per-example over the full test set), to keep the table directly comparable across variants.

Variant	Macro-F1↑	FP_{test} ↓	FN_{test} ↓	Finite MAE↓
ZeroProofML-SCM (free-magnitude tuples)	0.558 ± 0.255	0.199 ± 0.258	0.00943 ± 0.0201	2.91 ± 2.20
NoShock	0.544 ± 0.283	0.242 ± 0.314	$1.0 \times 10^{-5} \pm 3.16 \times 10^{-5}$	1.42 ± 0.661
NoSign	0.328 ± 0.191	0.400 ± 0.330	$4.0 \times 10^{-5} \pm 7.67 \times 10^{-5}$	1.31 ± 1.10
NoMargin	0.497 ± 0.336	0.360 ± 0.315	0.00699 ± 0.0221	2.94 ± 2.84
NoDetachScale	0.270 ± 0.157	0.117 ± 0.190	0.210 ± 0.145	10.2 ± 13.5
QAnchorOnes	0.402 ± 0.297	0.444 ± 0.290	0.00482 ± 0.00892	5.07 ± 3.23
DecodedMSE	0.443 ± 0.210	0.370 ± 0.238	0.00 ± 0.00	0.112 ± 0.0469
NoDetachRenorm	0.491 ± 0.168	$3.33 \times 10^{-6} \pm 1.05 \times 10^{-5}$	0.0599 ± 0.126	0.199 ± 0.251
NoDetachRenorm+DirBalance	0.511 ± 0.115	$6.33 \times 10^{-5} \pm 2.00 \times 10^{-4}$	0.0303 ± 0.0959	0.154 ± 0.225
NoDetachRenorm+AuxCE	0.474 ± 0.160	$3.33 \times 10^{-6} \pm 1.05 \times 10^{-5}$	0.0598 ± 0.126	0.0830 ± 0.0211

Dose angular SCM variants (targeted follow-ups; 10 seeds)

Table 7: **Dose: angular SCM variant progression (10 seeds; targeted follow-ups).** All rows use strict SCM decoding ($|D| < \tau_{\text{infer}} \Rightarrow \perp$) but vary the parameterization and training protocol. FP_{test} and FN_{test} are *test-normalized* rates over the full test set; FP_{cens} and FN_{in} are class-conditional rates on the true-censored and true in-range subsets, respectively. FN_{in} is 0.0 for all variants shown except the negative-result row. Bottom-rate is the fraction of test points decoded to $\perp$; in this suite the censored prevalence is ≈ 0.700, so bottom-rate near 0.700 is expected under correct gating.

Variant	What it tests	Macro-F1↑	FP_{test} ↓	FN_{test} ↓	FP_{cens} ↓	FN_{in} ↓	$\text{MAE}_{\text{finite}}$ ↓	Bottom-rate
Angular (plain)	Unit-circle parameterization only	0.5755 ± 0.000864	$(1.07 \pm 1.65) \times 10^{-4}$	0.00 ± 0.00	$(1.52 \pm 2.35) \times 10^{-4}$	0.00 ± 0.00	0.175 ± 0.0880	0.700 ± 0.00202
Angular + DirBalance	Class-imbalance control (censored directions)	0.5756 ± 0.000843	$(1.67 \pm 2.36) \times 10^{-5}$	0.00 ± 0.00	$(2.38 \pm 3.37) \times 10^{-5}$	0.00 ± 0.00	0.164 ± 0.125	0.700 ± 0.00196
Angular + hard curriculum (30 ep)	Direction recovery (short budget)	0.9040 ± 0.00138	$(3.60 \pm 4.51) \times 10^{-4}$	0.00 ± 0.00	$(5.13 \pm 6.42) \times 10^{-4}$	0.00 ± 0.00	0.658 ± 0.223	0.700 ± 0.00216
Angular + hard curriculum (50 ep)	Direction recovery + more runway	0.9040 ± 0.00132	$(2.60 \pm 4.29) \times 10^{-4}$	0.00 ± 0.00	$(3.71 \pm 6.11) \times 10^{-4}$	0.00 ± 0.00	0.393 ± 0.259	0.700 ± 0.00219
Angular + soft curriculum (50 ep)	Default: regression recovery + safety	0.5756 ± 0.000839	$(0.867 \pm 2.06) \times 10^{-4}$	0.00 ± 0.00	$(1.24 \pm 2.95) \times 10^{-4}$	0.00 ± 0.00	0.0765 ± 0.00684	0.700 ± 0.00196
Angular + curriculum + fMSE	Negative result (collapse)	0.195 ± 0.111	0.657 ± 0.126	$(0.667 \pm 2.11) \times 10^{-4}$	0.937 ± 0.179	$(2.22 \pm 7.03) \times 10^{-4}$	4.42 ± 2.69	0.0434 ± 0.125

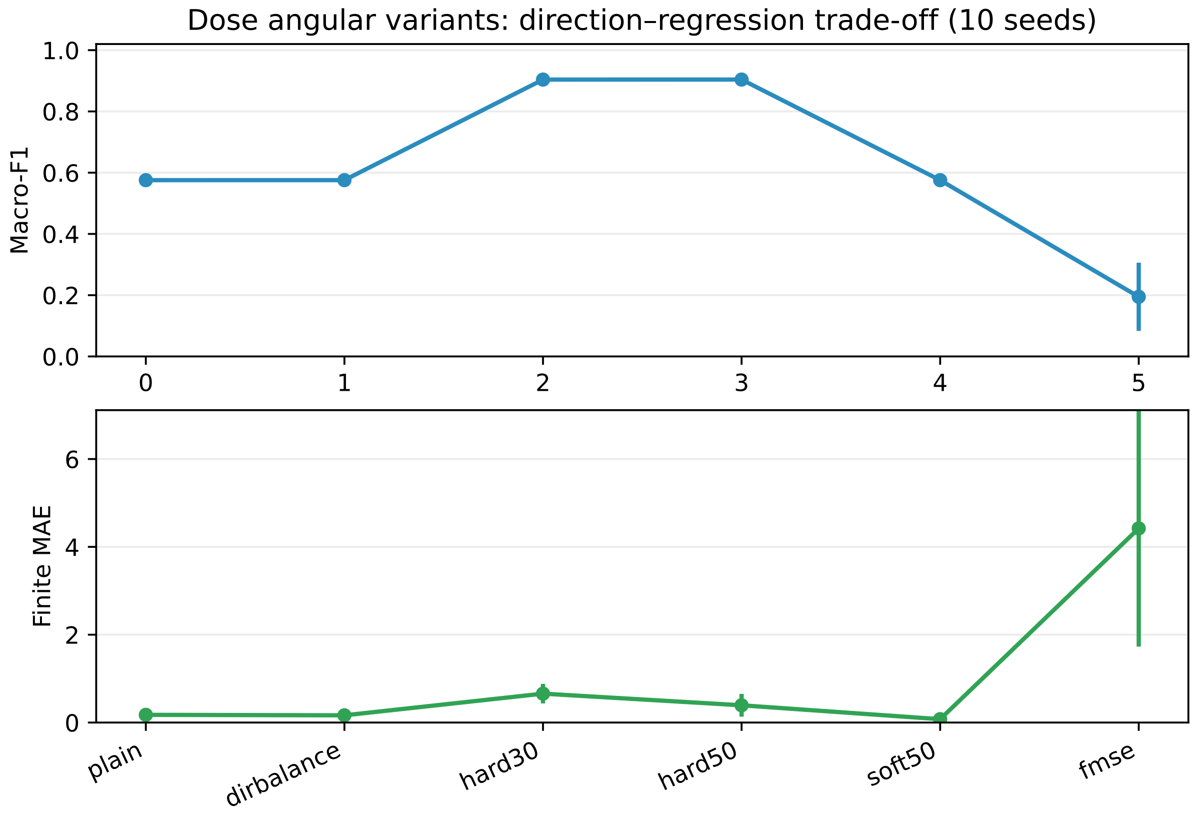


Figure 6: **Dose: direction–regression trade-off across angular variants (10 seeds).** Macro-F1 improves under harder curricula, while finite MAE increases, illustrating the scheduling/optimization tension discussed in §6. Shorthands match Table 7: `plain` (unit-circle only), `dirbalance` (censored-direction reweighting), `hard30`/`hard50` (hard curriculum; 30/50 epochs), `soft50` (soft curriculum; 50 epochs), `fmse` (adds finite-MSE term; negative result).

Dose direction head on frozen strict-SCM checkpoints (10 seeds)

Table 8: **Dose: decoupling censored direction via a frozen auxiliary head (10 seeds).** For each seed (1–10), we start from a trained strict-SCM checkpoint (tuple (P, Q)) and freeze *all* parameters except a tiny 2-way direction head trained only on censored samples. At inference time, the head is used only when the strict bottom gate triggers ($|Q| < \tau_{\text{infer}}$); therefore $\perp$ semantics and composability are unchanged. We report direction-only performance on the *true-censored* test subset (below vs. above). Mean±std across seeds; artifacts under `results/paper_suite_dose/run_dose_strict_ckpt_10seed/`.

Variant	Censored-direction Macro-F1↑	F1 (below)↑	F1 (above)↑
SCM (strict; tuple-only sign)	0.333 ± 0.033	0.301 ± 0.317	0.364 ± 0.384
SCM (strict + frozen DirHead)	0.856 ± 0.0018	0.857 ± 0.0021	0.856 ± 0.0019

Dose inference-threshold sensitivity (targeted follow-up)

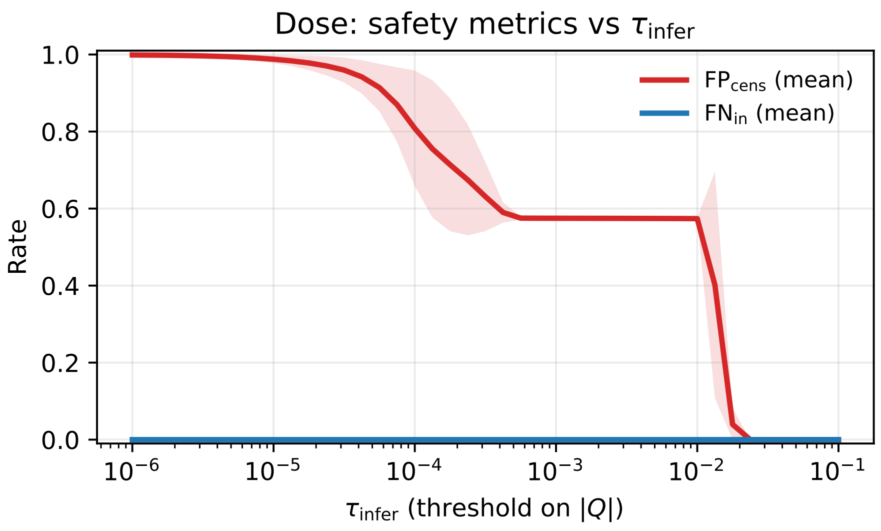


Figure 7: **Dose: post-hoc τ_{infer} sensitivity (targeted; 3 seeds).** FP_{cens} (false in-range on censored) decreases monotonically as τ_{infer} increases, while FN_{in} (false censoring on in-range) is non-decreasing and may remain near zero over much of the sweep (depending on operating point). This scan is computed by sweeping τ_{infer} at inference time for a fixed trained angular SCM (soft curriculum) model (no retraining). Aggregated mean±std across seeds. Artifacts are saved under `derived/dose_tau_infer_sweep.json` in `results/paper_suite_dose/run_dose_tau_infer_sweep_angular_soft_3seed_20260216_105708/`.

Dose composability risk model (thought experiment)

The one-step composability result (Table 2) is empirical evidence about what happens under a naive downstream computation. As an *optional* thought experiment—not evidence—we also include a simple multi-step risk model: suppose a pipeline is intended to propagate a "censored" flag across k=3 downstream boundaries, but each boundary independently drops the flag with probability δ=0.05. For scalar-only models, dropping the flag causes censored points to be treated as finite; for strict SCM decoding, the rate remains unchanged because $\perp$ is embedded in the value (NaN) rather than in an external flag.

Table 9: **Dose composability risk model (thought experiment; derived; 10 seeds).** A 3-step flag-propagation model with per-boundary drop probability δ=0.05. This is a risk model, not empirical evidence; it is derived from the one-step guarded rates plus the specified (k, δ) assumptions. See `derived/dose_composability.json`.

Method	Toy chain (3-step)↓
TwoStage	0.143 ± 0.00
TwoStage-Iso	0.143 ± 0.00
Rational+ϵ	0.634 ± 0.00242
Angular SCM (soft curriculum)	$1.24 \times 10^{-4} \pm 2.95 \times 10^{-4}$

B Proofs and Verification

Proofs (details)

Proof of Proposition 3. We restate the proof here for completeness. On $\mathcal{B}$ (Eq. (12)) we have $|D| < \tau_{\text{infer}}$, hence $\tau_{\text{train}} - |D| \geq \tau_{\text{train}} - \tau_{\text{infer}}$ and $[\text{ReLU}(\tau_{\text{train}} - |D|)]^2 \geq (\tau_{\text{train}} - \tau_{\text{infer}})^2$. Therefore $\mathbb{I}(|y_d| \geq \tau_{\text{sing}})\,[\text{ReLU}(\tau_{\text{train}} - |D|)]^2 \geq (\tau_{\text{train}} - \tau_{\text{infer}})^2 \cdot \mathbb{I}(\mathcal{B})$. Taking expectations and using the definition of $\mathcal{L}_{margin}$ in (11) yields (13). □

Table 10: **Robotics SCM ablations (10 seeds): auxiliary losses and stabilization.** Reported as test MSE mean±std.

Method	MSE↓
SCM-Basic	$(3.67 \pm 0.313) \times 10^{-4}$
NoSign	$(3.67 \pm 0.313) \times 10^{-4}$
NoMargin	$(3.67 \pm 0.313) \times 10^{-4}$
DetachScale	$(4.05 \pm 0.688) \times 10^{-4}$

Table 11: **Robotics additional baselines (10 seeds).** These baselines are included to defuse "ϵ-hack" objections and to contextualize the precision–consistency trade-off.

Method	MSE↓	Var (σ^2)↓	B0 P99 $\|\Delta\theta\|$ rel. dev. (%)
MLP	$(2.28 \pm 1.76) \times 10^{-4}$	3.11×10^{-8}	-2.24 ± 3.27
MLP-Iso (capacity-matched)	$(6.72 \pm 0.896) \times 10^{-4}$	8.02×10^{-9}	$+1.98 \pm 2.42$
Smooth ($P/\sqrt{Q^2+\alpha^2}$)	$(1.48 \pm 1.39) \times 10^{-4}$	1.94×10^{-8}	$+1.17 \pm 3.69$
LearnableEps ($P/(Q+\epsilon)$, learned)	$(1.74 \pm 2.31) \times 10^{-4}$	5.33×10^{-8}	$+0.217 \pm 2.69$
EpsEnsemble (3 members)	$(2.61 \pm 2.25) \times 10^{-4}$	5.05×10^{-8}	$+1.44 \pm 4.36$
ZeroProofML-SCM-Basic	$(3.67 \pm 0.313) \times 10^{-4}$	9.79×10^{-10}	$+0.754 \pm 3.05$

Table 12: **Paired statistics for headline claims (10 seeds).** We report the paired mean difference with a 95% bootstrap CI (20,000 resamples over seeds), the paired Wilcoxon signed-rank test (two-sided; exact; zeros excluded), and the exact paired sign test. "Win-rate" is the fraction of seeds where the direction of improvement holds, with a Clopper–Pearson 95% CI. When multiple rows show 10/10 seed wins without ties, the Wilcoxon and sign-test p-values repeat because the corresponding statistics hit their extremal values for n=10. All values are computed from the reference run's per-seed artifacts and saved under `derived/paired_stats.json`. For the variance-ratio row, Wilcoxon/sign tests and win-rate are not applicable because the ratio is computed across seeds (not from per-seed paired differences), so we report N/A.

Domain	Metric	Paired mean diff (95% CI)	Wilcoxon p	Sign p	Win-rate (95% CI)
Dose	$\mathrm{FP_{cens}}$ reduction (Rational+ϵ $\to$ Angular SCM soft curriculum)	0.5730 $[0.5713, 0.5747]$	1.95×10^{-3}	1.95×10^{-3}	10/10 $[0.692, 1.0]$
Dose	Regime BA gain (Angular SCM soft curriculum $-$ Rational+ϵ)	0.2865 $[0.2857, 0.2873]$	1.95×10^{-3}	1.95×10^{-3}	10/10 $[0.692, 1.0]$
RF	Yield gain (SCM$-$MLP)	37.51 $[29.66, 45.62]$	1.95×10^{-3}	1.95×10^{-3}	10/10 $[0.692, 1.0]$
RF	Peak p50 gain (SCM$-$MLP)	0.339 $[0.278, 0.390]$	1.95×10^{-3}	1.95×10^{-3}	10/10 $[0.692, 1.0]$
RF	Phase p50 improvement (MLP$-$SCM)	$40.7°$ $[26.8°, 53.2°]$	1.95×10^{-3}	1.95×10^{-3}	10/10 $[0.692, 1.0]$
Robotics	Variance ratio Var(MLP)/Var(SCM)	31.8 $[12.1, 127.2]$	N/A	N/A	N/A

Verification and auxiliary-loss activation

Strict-boundary trigger analysis. A theoretical buffer exists between the smooth training boundary τ_{train} and the strict inference threshold τ_{infer}. Proposition 3 yields:

$$P_{\text{bot}} \leq \frac{\mathcal{L}_{margin}}{(\tau_{\text{train}} - \tau_{\text{infer}})^2}, \tag{15}$$

where $P_{\text{bot}} = \mathbb{P}(|y_d| \geq \tau_{\text{sing}} \ \wedge \ |D| < \tau_{\text{infer}})$. We additionally log the interval $\tau_{\text{infer}} \leq |D| < \tau_{\text{train}}$ as an empirical `gap_mask` diagnostic.

Robotics gap verification. In the robotics experiment, strict SCM inference produced no bottom decodes on the test set (0/40,000 samples per seed at the configured τ_{infer}=10^{-6}). Moreover, for the deployed SCM-Basic configuration ($D = 1 + \text{softplus}(\cdot) + q_{\min}$ with $q_{\min}$=10^{-4}), $|D| \geq 1 + q_{\min}$ holds by construction, so the gap cannot be entered. As a sanity check on the exact stored test sets, we additionally verified that the gap mask is identically zero for SCM-Basic across all seeds (0/800,000 denominator entries in $[\tau_{\text{infer}}, \tau_{\text{train}})$); a deterministic test-set certificate is saved at `derived/rr_gap_verification.json` in the reference run.

Inference-threshold sensitivity (post-hoc). Because strict inference is defined by a threshold on denominator magnitude, a natural question is how much the results depend on the particular value of τ. Figure 8 reports inference-only sensitivity scans from the stored reference suite: for RF we plot the fraction of test points with $|Q(j\omega)| < \tau$ (the strict-check trigger for the shared-denominator SCM transfer head), and for robotics we plot a histogram of the SCM-Basic denominator entries $|D|$ with the configured $(\tau_{\text{infer}}, \tau_{\text{train}})$ overlaid. These scans are computed by re-running forward passes from the saved checkpoints on the deterministic regenerated test sets (no retraining); machine-readable outputs are saved under `derived/tau_infer_sensitivity.json` in the reference run. For Dose, our reference run does not store checkpoints, so we cannot do a full τ_{infer} sweep from stored artifacts; we instead report per-seed $|Q|$ quantiles in the same derived JSON as a coarse sanity check. To directly visualize the safety trade-off in Dose, we additionally include a targeted 3-seed follow-up sweep of τ_{infer} for strict SCM decoding (Appendix Figure 7; inference-only given the trained model).

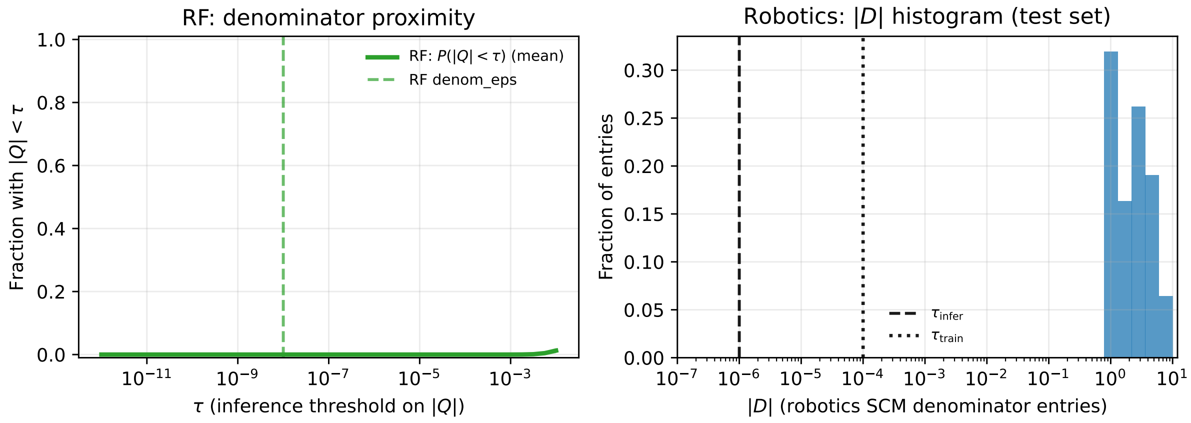


Figure 8: **Post-hoc inference-threshold sensitivity (no retraining).** Left: RF shared-denominator SCM transfer head—fraction of test points with $|Q(j\omega)| < \tau$ as τ varies (mean over seeds). The configured RF strict threshold τ_{infer} (implemented as `denom_eps`= 10^{-8}) lies far left, consistent with "no bottoms" on the RF test sets. Right: Robotics SCM-Basic denominator magnitude $|D|$ on the test set (aggregated across seeds); $|D|$ is bounded away from zero by construction and sits far from $(\tau_{\text{infer}}, \tau_{\text{train}})$.

When do auxiliary losses activate? The SCM objective (fit + margin + sign + rejection terms) is intended for settings with true pole-like behavior (singular targets) and/or for explicitly controlling the τ_{infer}–τ_{train} gap. In our main robotics benchmark (regularized DLS targets; SCM-Basic), these auxiliary terms have limited opportunity to contribute, consistent with the ablations. To verify that $\mathcal{L}_{margin}$ and $\mathcal{L}_{sign}$ do activate in a pole-target setting, we provide a small synthetic microbenchmark (`scripts/run_active_losses_micro.py`) that injects explicit $\pm\infty$ labels (lifted to $(Y_n, Y_d) = (\pm 1, 0)$) and reports the non-zero loss components and their effect on denominator behavior. In this microbenchmark (10 seeds), ablating $\mathcal{L}_{sign}$ increases the rate of large-error failures, while ablating $\mathcal{L}_{margin}$ can yield catastrophic collapses where $|Q|$ enters the τ_{infer}–τ_{train} gap on a non-trivial fraction of finite samples (Table 13; visually illustrated in Figure 9).

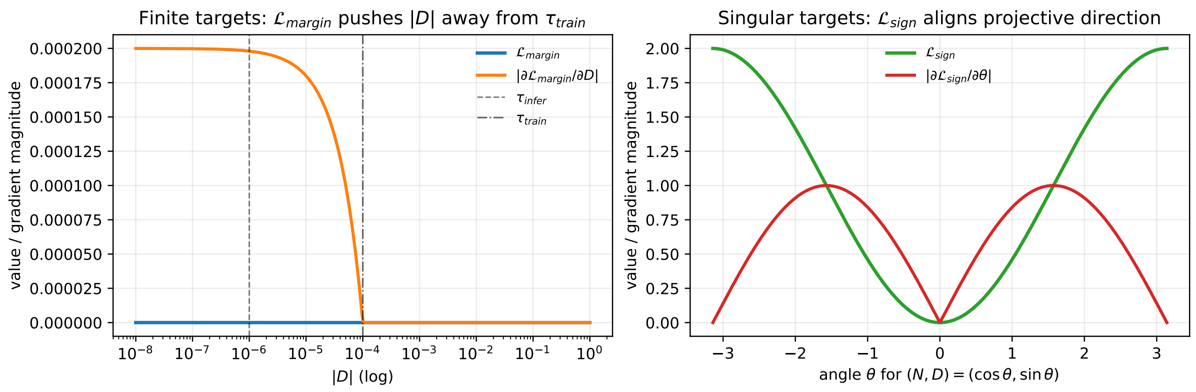


Figure 9: **Auxiliary losses respond sharply near singular regimes.** Curves show both loss values and corresponding gradient magnitudes. Left: for finite targets, $\mathcal{L}_{margin}$ produces large gradients when $|D| < \tau_{\text{train}}$, pushing denominators away from the gap. Right: for singular targets ($y_d \approx 0$), $\mathcal{L}_{sign}$ induces non-zero gradients that align the projective direction, resolving the sign ambiguity of the implicit fit loss. This is a didactic visualization; benchmark evidence is reported in Table 13.

Table 13: **Active-losses microbenchmark (10 seeds): pole-target setting.** The task injects explicit $\pm\infty$ labels (lifted to $(Y_n, Y_d) = (\pm 1, 0)$). We report robust statistics due to heavy-tailed behavior: median (IQR) across seeds and failure rates. "Margin violations" are the maximum finite-sample fraction per seed with $|Q| < \tau_{\text{train}}$ (i.e., entering the decoding clamp / gap region).

Variant	Finite MSE (med [IQR])↓	$P(\text{MSE} > 1)$	$P(\text{MSE} > 10)$	Max margin violations
Full ($\lambda_{\text{m}}>0, \lambda_{\text{sign}}>0$)	0.0709 [0.330]	0.20	0.10	9.77×10^{-4}
NoSign ($\lambda_{\text{sign}}=0$)	0.0742 [1.28]	0.30	0.20	2.93×10^{-3}
NoMargin ($\lambda_{\text{m}}=0$)	0.180 [0.302]	0.10	0.10	0.275

C Reproducibility

Experimental taxonomy (paper roles)

Table 14: **Baseline taxonomy used across domains.** Each paper claim is mapped to one of these controlled roles.

Role	Purpose
Oracle	Analytic ground truth or trusted numerical solver defining "physical correctness".
Strong baseline	Over-parameterized baseline (MLP or TwoStage) trained under the same protocol/budget.
MLP-Iso	Capacity-controlled baseline with parameter count matched to SCM heads.
Rational-only (non-SCM)	Isolates rational inductive bias without strict SCM $\perp$ semantics.
ZeroProofML (SCM)	Full SCM semantics (totalized division, $\perp$ propagation, strict inference).

Table 15: **Code-to-paper mapping (models).** This avoids conflating "rational" with "SCM".

Domain	Code name	Paper label	Role
Dose	`TwoStage`	TwoStage	Strong baseline
Dose	`TwoStage_Iso`	TwoStage-Iso	MLP-Iso
Dose	`Rational+eps`	Rational+ϵ	Rational-only (non-SCM)
Dose	`ZeroProofML_SCM_NoDetachRenorm_Angular`	Angular SCM (soft/hard curriculum; Table 1)	ZeroProofML (SCM)
Dose	`ZeroProofML_SCM_NoDetachRenorm`	SCM (strict; free-magnitude tuples)	Ablation
Dose	`ZeroProofML_SCM_NoDetachRenorm_AuxCE`	SCM (strict; +AuxCE)	Ablation
Dose	`ZeroProofML_SCM_NoDetachRenorm_DirHead_DetachBackbone`	Two-head direction (shared gate)	Ablation
Dose	`ZeroProofML_SCM`	ZeroProofML-SCM (free-magnitude tuples)	Ablation
RF	`Analytic-Oracle`	Oracle	Oracle
RF	`MLP-Deep-SiLU`	MLP (Deep)	Strong baseline
RF	`MLP-Iso-SiLU`	MLP-Iso (SiLU)	MLP-Iso
RF	`RationalTransfer`	Rational-only (non-SCM)	Rational-only (non-SCM)
RF	`ZeroProofML-SCM-Transfer`	ZeroProofML-SCM (shared denom)	ZeroProofML (SCM)
RF	`ZeroProofML-SCM-Decoupled`	SCM (decoupled)	Ablation
Robotics	`MLP`	MLP	Strong baseline
Robotics	`MLP-Iso`	MLP-Iso	MLP-Iso
Robotics	`ZeroProofML-SCM-Basic`	ZeroProofML-SCM-Basic	ZeroProofML (SCM)
Robotics	`Rational+eps`	Rational+ϵ	Rational-only (non-SCM)
Robotics	`DLS`, `DLS-Adaptive`	DLS references	Oracle-style references

Table 16: **Timing (selected 10-seed runs).** Dose/RF report training+selection wall-clock including per-epoch validation evaluation. For Dose we report both the 30-epoch reference suite and the 50-epoch angular follow-ups reported in Table 1. Robotics timings come from per-seed `comparison_table.csv` (microbench inference).

Domain	Method	Train wall-clock (s)	Inference (μs/sample)
Dose	TwoStage (strong baseline; 30 ep)	143 ± 1.79	n/a
Dose	TwoStage-Iso (capacity-matched; 30 ep)	60.8 ± 0.785	n/a
Dose	Rational+ϵ (non-SCM; 30 ep)	65.0 ± 0.851	n/a
Dose	SCM (free-magnitude; NoDetachRenorm; 30 ep)	83.9 ± 1.02	n/a
Dose	Rational+ϵ (non-SCM; 50 ep follow-up)	125 ± 11.6	n/a
Dose	Angular SCM (soft curriculum; 50 ep)	146 ± 14.8	n/a
Dose	Angular SCM (hard curriculum; 50 ep)	143 ± 13.1	n/a
RF	MLP (Deep)	$(1.33 \pm 0.0377) \times 10^3$	n/a
RF	MLP-Iso (capacity-matched)	452 ± 2.24	n/a
RF	RationalTransfer	464 ± 1.83	n/a
RF	ZeroProofML-SCM (shared denom)	471 ± 2.84	n/a
Robotics	MLP	519 ± 22.3	10.2 ± 0.334
Robotics	MLP-Iso (capacity-matched)	190 ± 1.59	6.19 ± 0.092
Robotics	ZeroProofML-SCM-Basic	380 ± 3.24	1.46 ± 0.109

Table 17: **Hyperparameters (frozen within each reported 10-seed suite).** Within a suite, all methods share the same optimizer and schedule; only the model class differs. Dose/RF use `restore-best` model selection on a per-epoch validation metric.

Domain	Key settings
Global	Projective renormalization constant $\varepsilon_{\text{renorm}}$=$10^{-9}$; implicit-fit guard γ=10^{-9} (fixed; not tuned).
Dose	Adam; lr 2×10^{-3}; batch 512; `restore-best`.
	Thresholds: $\tau_{\perp}$=0.01, τ_{infer}=0.025, τ_{finite}=0.05.
	Reference suite (30 ep; free-magnitude tuples): $\lambda_{\perp}$=20, λ_{fin}=1, λ_{sign}=10; cosine sign loss with balancing.
	Angular follow-ups (50 ep; $N = \cos\theta, D = \sin\theta$): λ_{fin}=1, λ_{sign}=10; soft curriculum uses $\lambda_{\perp}$=0 for 10 epochs then ramps to 10 slowly; hard curriculum ramps $\lambda_{\perp}$ to 20 quickly and warm-starts λ_{sign} (20→10 over 10 epochs).
RF	Adam; lr 2×10^{-3}; epochs 30; batch 1024; log-magnitude MSE loss.
	Strict threshold: τ_{infer} implemented as `denom_eps`= 10^{-8}.
	Train/test: $Q_{f,\text{train}} \leq 30$, $Q_{f,\text{test}} \leq 1000$; peak grid (local, 4096 points); yield threshold 0.9.
	SCM transfer conditioning: $(\log_{10}\omega_0, \log_{10} Q_f)$.
Robotics	Adam; lr 10^{-3}; epochs 150.
	Batch sizes: MLP 1024, SCM 256; MLP hidden 256×4, SCM hidden 64×2.
	Target: DLS λ=0.1.
	SCM: q=softplus, $q_{\min}$=10^{-4}, τ_{infer}=10^{-6}, τ_{train}=10^{-4}; implicit fit with non-detached fit-scale normalization (`implicit-no-detach-scale`).

Table 18: **Model sizes (parameter counts).** Counts are measured from reported model metadata (Dose) and saved checkpoints (RF) or per-seed robotics comparison tables.

Domain	Method	Parameters
Dose	TwoStage (strong baseline)	203,780
Dose	TwoStage-Iso (capacity-matched)	5,641
Dose	Rational+ϵ (non-SCM)	5,641
Dose	SCM (free-magnitude; NoDetachRenorm)	5,641
Dose	Angular SCM (NoDetachRenorm_Angular)	5,569
RF	MLP (Deep)	791,042
RF	MLP-Iso (capacity-matched)	17,282
RF	RationalTransfer	17,669
RF	ZeroProofML-SCM (shared denom)	17,283
RF	SCM (decoupled)	17,181
Robotics	MLP	199,170
Robotics	MLP-Iso (capacity-matched)	4,747
Robotics	ZeroProofML-SCM-Basic	4,740
Robotics	Rational+ϵ	199,438

Figure generation (scripts)

All empirical figures in the paper are generated from stored run directories (no retraining), using the `scripts/paper_vps` plotting utilities. For Dose, this includes the 30-epoch reference suite and the angular follow-up runs listed in the reproducibility statement. The toy figure in §1.1 is generated by `scripts/toy/generate_toy_pole_figure.py`.