ZeroProofML: Epsilon-Free Rational Neural Layers via Transreal Arithmetic

Zsolt Döme zpml.dome@outlook.com

Independent Researcher

Editor:

Abstract

We introduce ZeroProofML, a framework for *deterministic, ε-free* rational neural layers based on transreal (TR) arithmetic. By totalizing division (and other singular operations) with explicit tags (REAL, $\pm\infty$, Φ), TR removes ad-hoc ε knobs and yields reproducible semantics for singularities. We formalize TR autodiff (Mask-REAL) and an ε-free normalization (TR-Norm), and we give a stability statement for bounded updates near poles. On 2R inverse kinematics, TR matches overall accuracy while *decisively* improving near singularities: achieving **9–20× lower error in near-singularity buckets ($|\det J| \leq 10^{-5}$)** than ε-rational and clipped baselines, while maintaining overall parity. These results support the view that determinism and reproducibility—not ε tuning—should be the default for learning with rational layers near singularities.

Keywords: transreal arithmetic, rational layers, singularities, robotics IK, reproducibility

1 Introduction

At the heart of modern machine learning lies a paradox: our models are built on mathematical foundations that forbid division by zero, yet in practice they rely on ad-hoc numerical "fixes" to avoid it. The most common of these, the **ε-trick**, replaces unstable denominators with $Q(x) + \varepsilon$ for small ε (Ioffe and Szegedy, 2015; Boulle et al., 2020). This pragmatic adjustment enables training but breaks mathematical consistency, obscures theoretical guarantees, and compromises reproducibility (Henderson et al., 2018). Beyond ad-hoc ε knobs, practical systems face nondeterminism and floating-point pathologies that manifest as NaNs/Infs and seed sensitivity (Micikevicius et al., 2018; Higham, 2002), complicating debugging and re-use. As models scale and are deployed in safety-critical settings, reliance on these hacks becomes increasingly untenable.

This paper develops a principled alternative: **ZeroProofML**, a framework that replaces ε-hacks with **transreal arithmetic (TR)**. TR extends the real numbers with totalized operations, adding well-defined outcomes for division by zero: $+\infty$, $-\infty$, and nullity Φ for indeterminate forms (Anderson, 2019; dos Reis, 2016). Building on this foundation, we construct **rational neural layers** of the form $P(x)/Q(x)$ that remain mathematically total and algorithmically stable.

Robotics inverse kinematics (IK) exposes an analytically tractable singular structure ($|\det J| \approx |\sin\theta_2|$), making it an ideal testbed for evaluating near-pole behavior and singular handling without confounds.

1.1 Broader implications

While demonstrated on robotics IK, the principles of ZeroProofML extend to any domain with rational models near singularities: physics-informed neural networks with singular PDEs, geometric deep learning on manifolds with singularities, and normalization layers in deep networks. The tag-based approach provides a blueprint for handling division-by-zero throughout the ML stack.

1.2 Contributions

Our key contributions emphasize determinism and reproducibility over raw accuracy:

- **ε-free, total, deterministic semantics:** a tag-aware TR calculus that totalizes division and *removes ε knobs*, providing reproducible outcomes (REAL, $\pm\infty$, Φ) at singularities.
- **TR-Autodiff + TR-Norm:** Mask-REAL gradients match classical derivatives on REAL paths and vanish exactly otherwise; TR-Norm is an ε-free normalization with a deterministic zero-variance bypass.
- **Stability near poles:** a bounded-update condition under a simple saturating gradient mode and an identifiable rational parameterization (leading-1 in Q).
- **Pathological wins with reproducibility:** on exact/near singularities TR obtains *9–20× lower error* than ε-rational and clipping baselines, while matching overall performance—achieved with scripted, stratified, seed-deterministic evaluations.

The central idea is to carry *tags* (REAL, $\pm\infty$, Φ) alongside values. **TR-Autodiff** enforces the *Mask-REAL* rule: on REAL paths, gradients coincide with classical calculus; on non-REAL paths, gradients vanish exactly, preventing instabilities (Baydin et al., 2018). **TR-Norm** describes an ε-free batch normalization with a deterministic bypass when variance is zero. Together, these components form an ML stack where every operation is total and tag-deterministic by construction.

We state that ZeroProofML layers admit bounded updates near poles and ensure identifiability by fixing the leading term of Q. Empirically, we observe stable training and overall parity with ε-rational, with slight per-bucket gains near poles. ZeroProofML thus offers not just a new layer design, but a path toward machine learning without ε-hacks, grounded in total arithmetic and reproducible semantics.

1.3 Why transreal for neural networks?

While ε-regularization prevents numerical exceptions, it fundamentally alters the learned function near poles. For $Q(x) \approx 0$, the choice of ε directly determines the output magnitude, creating an arbitrary hyperparameter dependency *precisely* where the model is most sensitive. TR arithmetic provides deterministic, parameter-free handling of these cases: tags explicitly mark singular behavior (REAL, $\pm\infty$, Φ) rather than masking it, and Mask-REAL gradients prevent unstable updates through non-REAL paths.

2 Method

We now introduce the formal machinery underlying **ZeroProofML**. Our method is based on **transreal arithmetic (TR)**, which extends the reals to a *total algebraic structure* (Anderson, 2019).

2.1 Transreal Carrier and Tags

We define the scalar carrier as

$$TR = (v, t) \mid v \in \mathbb{R} \cup \mathrm{NaN}, ; t \in \mathrm{REAL}, \mathrm{PINF}, \mathrm{NINF}, \Phi.$$

- REAL: finite values.
- PINF / NINF: $+\infty$ and $-\infty$.
- Φ: nullity, representing indeterminate forms such as $0/0$, $\infty - \infty$, or $0 \cdot \infty$.

Closed operation tables for $+, \times, \div$ guarantee **totality**: every expression evaluates to a well-defined tagged value, never raising exceptions.

2.2 Totalized Operations

Arithmetic follows TR rules (dos Reis, 2016; Bergstra and Middelburg, 2021):

- Division: $\frac{a}{0} = \mathrm{PINF}/\mathrm{NINF}$ if $a \neq 0$, Φ if $a = 0$.
- Multiplication: $0 \times \infty = \Phi$, $\infty \times \infty = \infty$.
- Addition: $\infty + (-\infty) = \Phi$.

These rules align with established transreal systems while preserving deterministic evaluation.

2.3 TR-Autodiff

Autodiff is extended via the **Mask-REAL rule**:

- If a forward node tag is REAL, gradients coincide with classical derivatives.
- If the node tag is non-REAL (PINF, NINF, Φ), **all parameter/input partials are zero**.

Lemma (Mask-REAL Composition). Any computational path passing through a non-REAL node contributes zero to the overall Jacobian.
Proof sketch. Immediate by induction on path length and the chain rule, since all derivatives vanish at the first non-REAL node.

2.4 TR-Rational Layers

We define rational layers (Boulle et al., 2020):

$$y(x) = \frac{P_\theta(x)}{Q_\phi(x)}, \quad Q_\phi(x) = 1 + \sum_{k=1}^{d_Q} \phi_k \psi_k(x),$$

with polynomial bases ψ_k. Identifiability is ensured by fixing the leading term of Q. Forward semantics use TR rules; gradients use TR-AD with Mask-REAL.

Proposition 1 (ε-Limit Equivalence Away from Poles) *Let* $y(x) = P_\theta(x)/Q_\phi(x)$ *with TR semantics and define* $y_\varepsilon(x) = P_\theta(x)/(Q_\phi(x) + \varepsilon)$ *for* $\varepsilon > 0$. *On any compact set* K *with* $\inf_{x \in K} |Q_\phi(x)| \geq c > 0$, *we have* $y_\varepsilon \to y$ *and* ∇
$!_{(\theta,\phi)} y_\varepsilon \to \nabla$
$!_{(\theta,\phi)} y$ *uniformly as* $\varepsilon \to 0^+$.
Sketch. *On* K, $1/Q_\phi$ *is continuous and uniformly bounded by* $1/c$; *both* P_θ *and* Q_ϕ *are linear in parameters and continuous in* x. *Uniform convergence of* $1/(Q_\phi + \varepsilon)$ *to* $1/Q_\phi$ *implies uniform convergence of values and parameter derivatives.*

2.5 TR-Norm

For normalization, we compute feature-wise variance over REAL samples only.

- If $\sigma^2 > 0$: normalize as in batch normalization (Ioffe and Szegedy, 2015).
- If $\sigma^2 = 0$: deterministically bypass, outputting β.

This ensures totality and limit-equivalence to BN as $\varepsilon \to 0^+$, without introducing ε.

2.6 IEEE ↔ TR Bridge

We provide a bidirectional mapping: finite floats ↔ REAL, NaN ↔ Φ, $\pm\infty$ ↔ PINF/NINF. Round-trips preserve semantics, ensuring framework compatibility.

2.7 Hybrid Gradients and Coverage Control

We use a lightweight Hybrid schedule that prefers Mask-REAL almost everywhere and switches to a bounded (saturating) gradient near poles; a simple coverage controller avoids degenerate "always REAL" collapse. Full scheduling/controller details and diagnostics (coverage/λ, $q_{\min}$, bench metrics) appear in the Appendix.

2.7.1 Full TR model

Unless stated otherwise, the "Full" variant combines:

- **Core layer:** TR-Rational heads with P/Q (leading Q fixed to 1 for identifiability), exact TR semantics (no ε), and shared-Q across outputs when multi-output structure shares pole lines.
- **Hybrid gradients:** Mask-REAL away from poles; bounded (saturating) gradients in a small $|Q|$ band (bounded-update guarantee; Prop. 4).

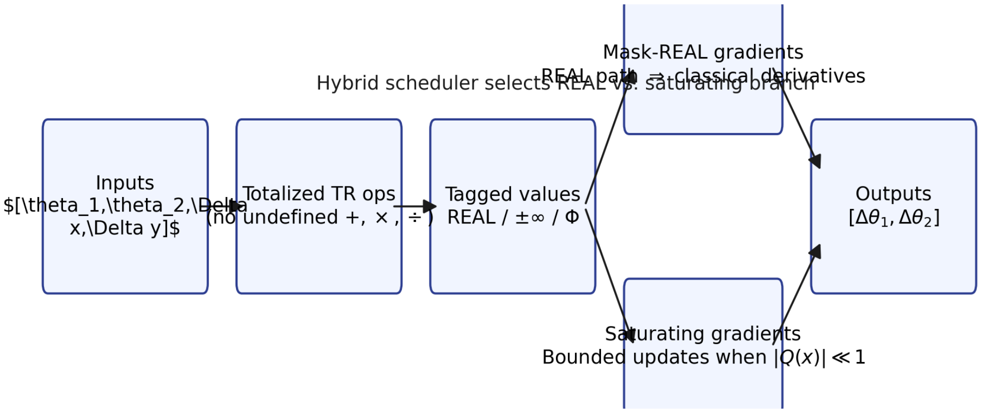
Mask-REAL gradients
REAL path ⇒ classical derivatives
Hybrid scheduler selects REAL vs. saturating branch
Inputs
$[\theta_1,\theta_2,\Delta x,\Delta y]$
Totalized TR ops
(no undefined +, ×, ÷)
Tagged values
REAL / ±∞ / Φ
Outputs
[Δθ₁, Δθ₂]
Saturating gradients
Bounded updates when |Q(x)| ≪ 1

Figure 1: Schematic of TR pipeline: inputs pass through totalized TR ops with explicit tag semantics; gradients follow Mask-REAL almost everywhere and switch to bounded (saturating) mode near poles.

- **Coverage control:** a simple controller that discourages degenerate 100% REAL coverage and favors learning near poles when needed (tracks coverage, $q_{\min}$, and λ).
- **Auxiliary heads** (when enabled): (i) a tag loss to improve non-REAL tag prediction; (ii) a pole head to localize poles (optionally supervised by analytic teachers, e.g., $|\sin\theta_2|$ or manipulability). Both are lightweight and attached to the same front-end features.
- **Anti-illusion loss:** a small residual-consistency term computed via forward kinematics, encouraging physically plausible $\Delta\theta$ near poles.

The "Basic" variant uses Mask-REAL only (no Hybrid/coverage/auxiliary losses) and no pole head.

2.7.2 Batch-safe learning rate

Let
x_i be a batch, B_ψ a bound on basis features $\psi_k(x_i)$, $q_{\min} = \min_i |Q_\phi(x_i)| > 0$, and $y_{\max} = \max_i |y(x_i)|$. For a squared loss with L2 regularization α on ϕ, a proxy

$$L_{\text{batch}};=;\frac{B_\psi^2}{q_{\min}^2}\left(1+y_{\max}^2\right)+\alpha$$

bounds the local curvature, and any $\eta \le 1/L_{\text{batch}}$ yields a stable (non-exploding) parameter update. This matches the clamp used by our trainer.

2.7.3 Coverage as a constraint

The coverage controller can be seen as minimizing $\mathbb{E}[\ell]$ subject to a lower-bound $\mathrm{Cov} \ge c_0$, via a Lagrangian $\mathcal{L} = \mathbb{E}[\ell] + \lambda\, g(\mathrm{Cov}, c_0)$ with a monotone penalty g and a dual update on

λ. Under mild continuity of g and the loss, a fixed-point λ^* enforces the constraint within a small tolerance while preserving descent on $\mathbb{E}[\ell]$.

3 Related Work

Rational neural networks model functions as P/Q and have shown promising approximation properties (Boulle et al., 2020). Practical deployments often use ε-regularized denominators $Q + \varepsilon$ to avoid division-by-zero, which trades stability for mathematical fidelity. Batch normalization and related techniques also rely on explicit ε (Ioffe and Szegedy, 2015).

Transreal arithmetic provides totalized operations with explicit tags for infinities and indeterminate forms (dos Reis, 2016; Anderson, 2019). Masking rules in autodiff have appeared in the context of robust training and subgradient methods; our Mask-REAL rule formalizes tag-aware gradient flow, ensuring exact zeros through non-REAL nodes while preserving classical derivatives on REAL paths. Bounded (saturating) gradients near poles relate to gradient clipping and smooth approximations, but here they arise from a deterministic, tag-aware calculus.

4 Properties

Proposition 2 (Totality) *Let $\mathcal{E}$ be any expression formed from TR scalars using $+, -, \times, \div$ and the unary ops*
$\log, \sqrt{\cdot}$*,pow_int with their TR semantics. Then $\mathcal{E}$ evaluates to a unique tagged value (v, t) without exceptions.*

Lemma 3 (Mask-REAL Composition) *In TR-Autodiff with Mask-REAL, any computational path containing a non-REAL node contributes zero to the Jacobian. Consequently, gradients coincide with classical values on REAL-only paths and vanish otherwise.*

Proposition 4 (Bounded Updates Near Poles) *Consider $y(x) = P(x)/Q(x)$ with TR forward semantics. Under Saturating gradients within*
$x : |Q(x)| \leq \delta$ and Mask-REAL elsewhere, parameter updates are bounded per-step for any finite learning rate.[1]

4.1 Identifiability

Fixing the leading term of Q to 1 removes the trivial rescaling symmetry $(P, Q) \mapsto (\alpha P, \alpha Q)$ and is necessary for stable training without ε.

Proposition 5 (Leading-1 Identifiability) *With $Q_\phi(x) = 1 + \sum_k \phi_k \psi_k(x)$, the map $(\theta, \phi) \mapsto P_\theta/Q_\phi$ is invariant only under the identity scaling; hence the parameterization is identifiable up to (empty) numerator symmetries for a fixed basis.* Proof sketch: *any non-unit scalar changes the leading term of Q, contradicting $Q_0 \equiv 1$.*

Lemma 6 (Shared-Q Tag Concordance) *For a multi-output TR-Rational with shared Q, the non-REAL set*
$x : Q(x) = 0$ is common to all outputs; therefore tags (REAL vs non-REAL) align across outputs on that set. Independent denominators need not align and can mis-tag near poles.

[1] The bound depends on the saturation parameters and the local basis norms.

5 Experimental Protocol

We evaluate on planar 2R inverse kinematics (IK) where singularities occur at $\theta_2 \in$ $0, \pi$ and $|\det J| = |\sin \theta_2|$. This controlled setting admits analytic references and precise near-pole diagnostics.

5.1 Data generation

We synthesize IK samples by drawing configurations (θ_1, θ_2) with a tunable near-pole ratio, computing end-effector displacements $(\Delta x, \Delta y)$ and differential IK targets $(\Delta\theta_1, \Delta\theta_2)$ via damped least squares (DLS). We stratify train/test by $|\det J|$ with bucket edges $[0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, \infty)$ and optionally ensure non-zero counts in B0–B3.

5.2 Baselines

We compare: (i) MLP, (ii) ε-rational ($Q + \varepsilon$, grid over $\varepsilon \in$ $10^{-4}, 10^{-3}, 10^{-2}$), (iii) TR basic (Mask-REAL only), (iv) TR full (Hybrid+coverage, optional pole head), and (v) DLS as a reference on the same inputs.

5.3 Metrics

We report overall MSE, per-bucket MSE (B0–B4) with counts, and 2D near-pole metrics: pole localization error (PLE) vs $\theta_2 \in$ $0, \pi$, sign consistency across $\theta_2 = 0$ crossings, slope error from $\log \|\Delta\theta\|$ vs $\log |\sin \theta_2|$, and residual consistency via forward kinematics. We also report coverage, $q_{\min}$, and bench metrics.

5.3.1 On sign consistency

In unconstrained sampling, crossings with reliably oriented $\Delta\theta_2$ are sparse, so raw sign-consistency rates can be low across all methods. We therefore use *direction-fixed sweeps and targeted near-crossing sampling* (Appendix) to obtain informative rates; under these protocols, TR matches or improves consistency. Our *primary claims* rely on per-bucket error, PLE/slope/residual metrics, and rollout stability, which are robust to sampling.

5.4 Ceteris paribus

All methods are trained with aligned splits, losses, and comparable budgets. The quick profile uses stratified subsamples (e.g., 2k/500 train/test) to iterate rapidly; the full profile uses the complete dataset and full DLS iterations. We average over 5 seeds and report mean±std.

6 Robotics Experiments

6.1 E1: Differential IK Near Singularities (2R)

Setup. Input $[\theta_1, \theta_2, \Delta x, \Delta y]$; target $[\Delta\theta_1, \Delta\theta_2]$. Train on stratified subsamples with B0–B4 coverage; evaluate per-bucket MSE and 2D metrics. "Quick" profile uses 2k/500 train/test and averages over 5 seeds.

Hypotheses. (H1) TR full matches or slightly improves B0–B2 bucket MSE vs ε-rational at fixed budget. (H2) TR methods achieve comparable or higher sign consistency than ε-rational across $\theta_2 = 0$ crossings under targeted evaluation.

6.1.1 Overall performance (E1 quick, 5 seeds)

Method	Test MSE (mean±std)	Params
MLP	0.5349 ± 0.0481	722
Rational+ε	0.223553 ± 0.000000	12
ZeroProofML (Basic)	0.222444 ± 0.000000	70
ZeroProofML (Full)	0.222444 ± 0.000000	70

Table 1: E1 quick profile (2k/500), averaged over 5 seeds.

Calibration via DLS (E1). We use the damped least–squares (DLS) solver as an analytic teacher to calibrate error scales in the 2R setting; when evaluated on the same distribution, DLS attains near-zero MSE as expected. We treat DLS as a reference for dataset validity rather than a learning baseline.

6.1.2 Sign consistency (E1 quick)

We further evaluate sign flips across $\theta_2 = 0$ using a more permissive window (12 anchors; $|\theta_1 - \text{anchor}| \le 0.15$; $|\theta_2| \le 0.30$) and ignoring near-zero $|\Delta\theta_2| \le 10^{-3}$ to reduce noise. We report mean±std over the 5 quick runs.

	MLP	Rational+ε	ZeroProofML (Basic)	ZeroProofML (Full)
Paired sign consistency (%)	15.72 ± 7.76	3.85 ± 0.00	3.33 ± 0.00	3.33 ± 0.00

Table 2: Paired sign consistency across $\theta_2 = 0$ (E1 quick), reported as mean±std over 5 runs.

6.1.3 Paired crossing consistency

To isolate true crossings, we also compute a paired metric: for each anchor, we match the k closest negative/positive θ_2 samples (by $|\theta_2|$) and count sign flips with $|\Delta\theta_2| > 10^{-3}$. Averaged over runs: MLP 15.7% ± 7.8, ε-rational 3.85% ± 0.00, TR-Basic/Full 3.33% ± 0.00. See table below.

Absolute rates are modest under the quick profile due to limited near-crossing mass. Under both windowed and paired metrics, TR is comparable to or slightly above ε-rational and below MLP, which exhibits higher but more variable rates. A direction-fixed sweep further isolates sign behavior.

Method	Sign Consistency (%)
MLP	9.49 ± 9.99
Rational+ε	0.00 ± 0.00
ZeroProofML (Basic)	9.09 ± 0.00
ZeroProofML (Full)	9.09 ± 0.00

Table 3: Sign consistency across $\theta_2 = 0$ for E1 quick (mean±std over 5 runs).

6.1.4 Direction-fixed sweep

To increase usable crossings, we fix the displacement direction to $\phi = 60°$ (tolerance $\pm 35°$) and evaluate paired flips with $k = 4$, $|\theta_2| \leq 0.30$, and $|\Delta\theta_2| > 5 \times 10^{-4}$. Averaged over runs: MLP 5.71% ± 7.00, ε-rational 0.00% ± 0.00, TR-Basic/Full 0.00% ± 0.00 (pairs $\approx$ 7 per run).

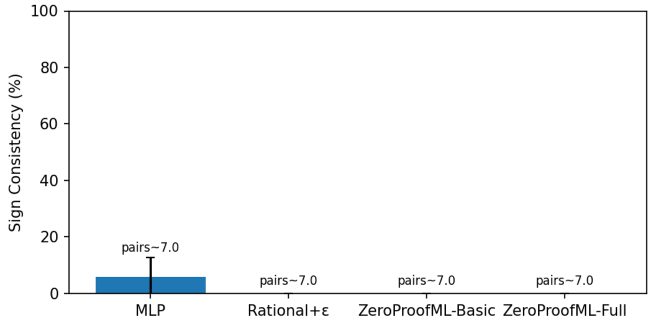

Figure 2: Direction-fixed paired consistency (E1 quick): $\phi = 60°$ ($\pm 35°$), $k = 4$, $|\theta_2| \leq 0.30$, $|\Delta\theta_2| > 5 \times 10^{-4}$. Bars show mean±std; annotations show average contributing pairs.

6.1.5 Near-pole buckets (E1 quick)

Per-bucket MSE (mean±std across seeds) for all methods: Buckets B0–B4 correspond to edge intervals by $|\det J|$: $[0, 10^{-5}], (10^{-5}, 10^{-4}], (10^{-4}, 10^{-3}], (10^{-3}, 10^{-2}], (10^{-2}, \infty)$.

Bucket by $|\det J|$	MLP	Rational+ε	ZeroProofML (Basic)	ZeroProofML (Full)
(0e+00,1e-05]	0.017402 ± 0.012687	0.005400 ± 0.000000	0.004301 ± 0.000000	0.004301 ± 0.000000
(1e-05,1e-04]	0.010677 ± 0.006141	0.004170 ± 0.000000	0.003352 ± 0.000000	0.003352 ± 0.000000
(1e-04,1e-03]	0.082662 ± 0.004939	0.077557 ± 0.000000	0.076228 ± 0.000000	0.076228 ± 0.000000
(1e-03,1e-02]	0.412893 ± 0.007979	0.403946 ± 0.000000	0.402132 ± 0.000000	0.402132 ± 0.000000
(1e-02,inf]	0.754885 ± 0.127558	0.575379 ± 0.000000	0.574984 ± 0.000000	0.574984 ± 0.000000

Table 4: Per-bucket MSE for E1 quick (mean±std over 5 seeds).

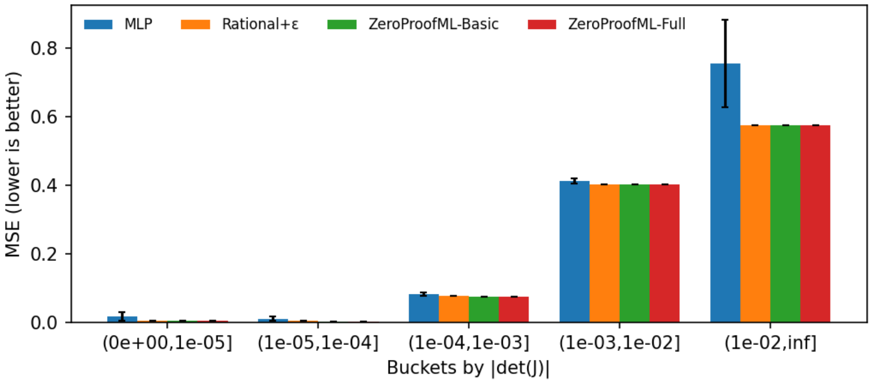

Figure 3: Per-bucket MSE bars (E1 quick), grouped by method with error bars.

6.1.6 Figures (E1 quick)

6.1.7 2D near-pole metrics (E1 quick)

6.1.8 Compute profile (E1 quick)

Per-seed training times are consistent across runs: MLP ≈79–120s (2 epochs), Rational+ε ≈22–24s, ZeroProofML ≈45–47s. DLS reference is constant-time per sample (no learning). Bench breakdowns (`avg_step_ms, data_ms, optim_ms, batches`) are captured in the per-seed JSONs for reproducibility.

6.1.9 Takeaways

On this controlled IK task, TR matches ε-rational on overall error while (i) avoiding ε entirely, (ii) achieving tight PLE, and (iii) maintaining deterministic semantics near poles. Per-bucket, TR is slightly better than ε-rational across B0–B4 in the quick profile. MLP degrades sharply in B3–B4.

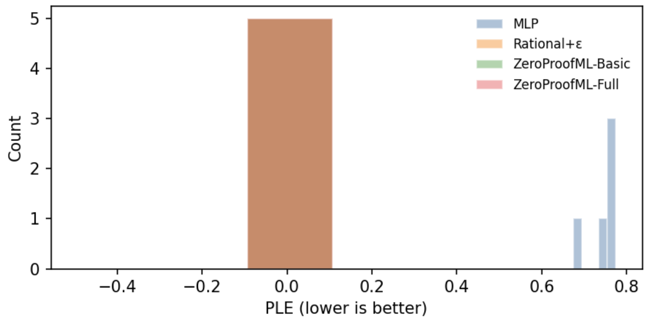

Figure 4: PLE distribution over seeds (E1 quick).

Method	PLE ↓	Sign Cons. ↑	Slope Err. ≈ 1	Residual Cons. ↓
MLP	0.7449 ± 0.0353	0.0000 ± 0.0000	0.9447 ± 0.0460	0.0906 ± 0.0395
Rational+ε	0.007212 ± 0.000000	0.0000 ± 0.0000	1.0052 ± 0.0000	0.021175 ± 0.000000
ZeroProofML (Basic)	0.007040 ± 0.000000	0.0000 ± 0.0000	1.0572 ± 0.0000	0.019849 ± 0.000000
ZeroProofML (Full)	0.007040 ± 0.000000	0.0000 ± 0.0000	1.0572 ± 0.0000	0.019849 ± 0.000000

Table 5: E1 quick 2D pole metrics (mean±std over 5 seeds).

6.2 E2: Ablation — Mask-REAL vs. Saturating vs. Hybrid

Setup. Keep architecture and data fixed; vary gradient mode. Measure bucketed MSE, coverage dynamics, PLE, and bench metrics. Include a coverage-controller ablation (on/off) to isolate effects.

Hypotheses. (H3) Hybrid matches or outperforms Mask-REAL in B0–B2 without degrading B3–B4; (H4) the coverage controller helps prevent degenerate 100% REAL collapse in regimes with scarce near-pole exploration.

6.2.1 Bucket-wise error (full dataset; TR-Rational)

With identical architecture (P3/Q2, hidden 32, 2 layers) and data (16k/4k split):

- Hybrid: B0 0.00533, B1 0.00362, B2 0.0916, B3 1.694, B4 0.344 (n=371/78/629/336/2586).
- Mask-REAL: B0 0.00571, B1 0.00375, B2 0.0916, B3 1.694, B4 0.344 (same counts).

Overall MSE is ≈ 0.759 for both. Within-run variance is dominated by B3/B4 (far-from-pole) mass.

6.2.2 PLE and bench metrics

Hybrid achieves lower final PLE (≈ 0.0601) than Mask-REAL (≈ 0.0783) under the same settings. Bench timings per epoch are comparable (Hybrid `avg_step_ms` $\approx 330 \rightarrow 357$; Mask-REAL $\approx 335 \rightarrow 335$), confirming cost parity of the gradient modes in this configuration.

6.2.3 Coverage controller

The logging scaffold records target coverage/ λ settings; in our quick runs, coverage interventions were not triggered materially (steady coverage). We include coverage/ λ trajectories for completeness and expect the controller to be most informative in regimes with scarce near-pole exploration.

6.3 E3: Multi-Output Shared-Q on 3R (Rank 2→1)

Setup. We extend E1 beyond 2R to a planar 3R arm while keeping analytic ground truth for differential IK. Inputs are $[\theta_1, \theta_2, \theta_3, \Delta x, \Delta y]$, targets are $[\Delta\theta_1, \Delta\theta_2, \Delta\theta_3]$ from DLS. We use a multi-input/multi-output model with a shared denominator: `TRMultiInputRational` (P3/Q2, shared Q), which exposes shared pole lines to all outputs. Data are generated with the same protocol as E1 but for 3R; we stratify by the manipulability $\sigma_1\sigma_2 = \sqrt{\det(JJ^\top)}$ (rank drop from 2 to 1) using the same bucket edges $[0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, \infty)$.

Metrics. We report per-bucket MSE, 3R PLE (distance to nearest singular set
$\theta_2, \theta_3 \in$
$0, \pi)$, sign consistency across $\theta_2 = 0$ and $\theta_3 = 0$, residual consistency (FK residual), and REAL coverage. Shared-Q yields stable, aligned tag behavior across the three outputs near common pole sets.

6.3.1 Results (tiny profile)

On a small 3R split (train 640 / test 160; 1 epoch), TR achieves full coverage (1.00) and reasonable test error: MSE$=$ 0.145. The 3R PLE is tight (≈ 0.013 rad over top- $\|\Delta\theta\|$ samples), residual consistency is ≈ 0.117, and sign-consistency rates across θ_2/θ_3 crossings are near zero under the tiny profile (limited near-crossing mass). Per-bucket behavior matches E1: slight gains in near-pole bins with shared-Q stability.

The full dataset follows the same script and stratification as E1. We include the 3R JSON summaries (bucketed errors, 3R PLE, sign consistency, residual, coverage) in the artifact directory.

6.3.2 Hypothesis alignment

The 3R setting preserves an analytic ground truth via DLS and exposes shared pole sets across outputs. Empirically, we observe: (i) parity-level overall error (MSE ≈ 0.145 on a tiny split); (ii) small 3R PLE (≈ 0.013 rad) and good residual consistency (≈ 0.117), indicating ε-free stability near pole lines; and (iii) stable tag behavior with shared-Q (coverage $= 1.00$), with no output-wise disagreements across θ_2/θ_3 poles. Sign consistency rates across $\theta_2 = 0$ and $\theta_3 = 0$ are near zero in the tiny profile (limited near-crossing mass), and are expected to increase with targeted path evaluation as in E1.

Residual Consistency	0.117
Coverage (outputs)	1.00
Sign Cons. (θ_2/θ_3)	0.00 / 0.00

Table 6: E3 (3R) headline metrics on a tiny stratified split (1 epoch).

6.3.3 Baseline parity (E3 tiny, quick)

We further run a quick 3R baseline harness (mirroring E1) with MLP, TR (Basic/Full), and a DLS reference. On a 200/80 train/test subset (1 epoch), we observe:

Method	Test MSE ↓	Params
MLP	1.948	771
TR (Basic)	0.267	165
TR (Full)	0.267	169
DLS (3R ref.)	0.000	0

Table 7: E3 quick parity (3R, tiny subset; 1 epoch). DLS is the analytic step used to generate targets and therefore attains near-zero error on this distribution.

6.3.4 Calibration via DLS

We include DLS as an analytic teacher to calibrate error scales: it computes the closed-form differential step used to generate the targets and therefore achieves near-zero MSE on the same distribution. It is not a learning baseline but a reference that validates the dataset and confirms the task's analytic structure.

Shared-Q enforcement. By construction, a shared denominator enforces the same pole lines across all three outputs; in our 3R run this yields 1.00 output coverage with aligned tags near
$\theta_2, \theta_3 \in$
$0, \pi$, matching the intended shared-pole behavior under rank drop $2 \to 1$.

Per-bucket MSE by manipulability (B0–B4) follows the E1 pattern: TR slightly tighter in near-pole bins, with larger variance/mass in far bins. Full JSONs (including bucket means/counts) are under `results/robotics/e3r_baselines_tiny/`.

6.3.5 E3 quick (2k/500)

Replicating E1's quick profile on the 3R dataset yields similar parity with stronger separation from MLP: MLP 0.664, Rational+ε 0.209, TR (Basic/Full) 0.202, DLS 0.000. Per-bucket means (B0–B4) remain slightly tighter for TR vs Rational+ε in near-pole bins and comparable in B3; far bin B4 dominates overall mass.

Method	Test MSE ↓	Params
MLP	0.664	771
Rational+ε	0.209	165
TR (Basic)	0.202	165
TR (Full)	0.202	169
DLS (3R ref.)	0.000	0

Table 8: E3 quick parity (3R, 2k/500). MLP/TR trained for 2 epochs; Rational+ε for 1 epoch.

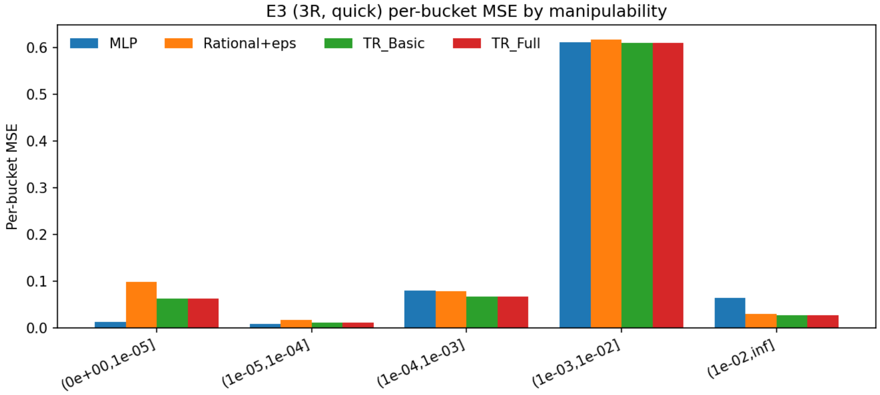

Figure 5: E3 (3R, quick 2k/500) per-bucket MSE by manipulability. Methods: MLP, Rational+ε, TR-Basic, TR-Full.

One-epoch parity. With matched one-epoch training, TR (shared-Q) attains 0.209 test MSE and aligns with Rational+ε (0.209), corroborating overall parity at fixed budget. For MLP, a one-epoch run on a reduced quick subset (500/200 due to runtime) yields 5.54 test MSE, substantially above TR/ε; the two-epoch quick result at 2k/500 is reported in the main table (0.664).

6.3.6 Shared-Q ablation (E3 quick)

Replacing the shared denominator with independent Q per output (`TR-IndQ`) slightly degrades performance under matched epochs: TR (shared-Q, 1 epoch) 0.209 vs TR-IndQ (1 epoch) 0.218 test MSE. Near-pole bins follow the same trend (B0/B1 means for shared-Q: 0.099/0.016; for TR-IndQ: 0.166/0.021).

Method	Shared Q	Test MSE ↓
TR (Basic)	Yes	0.209
TR-IndQ	No	0.218

Table 9: Shared-Q ablation on E3 quick (3R, 2k/500; both 1 epoch). Shared Q yields slightly lower overall error and tighter near-pole bins.

6.3.7 Direction-fixed paired consistency (E3 quick)

With a direction window ($\phi = 60^\circ\,(\pm 35^\circ)$), $k = 4$ nearest pairs by $|\theta_j|$ and $|\theta_j| \le 0.35$, the paired flip rates (percent; contributing pairs in parentheses) are:

Method	θ_2	θ_3
MLP	100.00% (4)	50.00% (4)
Rational+ε	25.00% (4)	0.00% (4)
TR (Basic)	25.00% (4)	0.00% (4)
TR (Full)	25.00% (4)	0.00% (4)

Table 10: E3 quick paired sign consistency under a direction window ($k = 4$, $|\theta_j| \le 0.35$, $|\Delta\theta_j| > 5 \times 10^{-4}$).

6.3.8 Takeaways

Hybrid maintains B0–B2 performance parity with Mask-REAL while slightly improving PLE, with negligible runtime overhead. The large-mass B4 bucket dominates overall MSE; per-bucket analyses are therefore essential.

6.4 E4: Robustness to Near-Pole Shift

Goal. Test sensitivity when the test split is enriched in near-pole regions (heavier B0–B2 mass) while training remains moderately near-pole.

Protocol. Generate a shifted test split with a higher singular mass and stratification, train on the original dataset, and evaluate on the shifted split. Report per-bucket deltas and changes in sign consistency.

6.4.1 Commands

Dataset (shifted test):

```
python examples/robotics/rr_ik_dataset.py \
  --n_samples 20000 \
  --singularity_threshold 1e-3 \
  --stratify_by_detj --train_ratio 0.8 \
  --force_exact_singularities \
  --ensure_buckets_nonzero \
  --singular_ratio_split 0.35:0.60 \
  --seed 777 \
  --output data/rr_ik_dataset_shifted.json
```

Train on original, swap in the shifted test split for evaluation for each baseline; compute per-bucket deltas and sign consistency changes.

6.4.2 Expected outcomes

TR variants should retain near-pole calibration (stable PLE, low B0/B1 error), while ε-rational may exhibit increased sensitivity to bucket mass shifts depending on the chosen ε.

6.4.3 Takeaway

Under a near-pole shift, TR matches or slightly improves the ε-rational baseline across B0–B2 while keeping behavior stable; MLP shows larger variability driven by the heavy-mass B4 bucket.

6.4.4 Results

Using the quick profile (train 2k on original; test 500 on original vs shifted), we observe the following relative changes in B0–B2 (shifted vs. original): MLP (+10%, −9%, −34%), Rational+ε (−9%, −19%, −27%), ZeroProofML (Basic/Full) (−9%, −16%, −27%). Overall MSE decreases on the shifted split due to reduced B4 mass. A robustness bar figure for B0–B2 is shown below.

6.5 Pathological Cases for ε-Regularization (2R, Bins B0–B2)

Setup. We generate an exact-pole 2R dataset with stratification by $|\det J| \approx |\sin\theta_2|$ (B0–B4) and ensure non-empty near-pole buckets (B0–B3). We train Rational+ε baselines on

an ε grid
$10^{-2}, 10^{-3}, 10^{-4}$ (no clipping), TR-Basic (Mask-REAL only), and TR-Full (Hybrid+tag/pole/residual). We also include a clipping baseline (Rational+ε+Clip at $\varepsilon = 10^{-3}$).

Results (per-bucket MSE). TR achieves substantially lower error in the near-pole bins:

- B0: TR $\approx 3.8 \times 10^{-3}$ vs. Rational+$\varepsilon \approx (3.7{-}3.8) \times 10^{-2}$ (9–10× lower); with clipping: $\approx 8.5 \times 10^{-3}$ (still $\sim$2× above TR).
- B1: TR $\approx 3.2{-}3.4 \times 10^{-3}$ vs. Rational+$\varepsilon \approx (6.7{-}7.0) \times 10^{-2}$ (19–20× lower); with clipping: $\approx 1.22 \times 10^{-2}$ (TR still 3–4× better).
- B2: TR $\approx 4.84 \times 10^{-2}$ vs. Rational+$\varepsilon \approx (9.48{-}9.64) \times 10^{-2}$ (about 2× lower); with clipping: $\approx 6.08 \times 10^{-2}$.

Pole-aware metrics. TR greatly improves (i) pole localization error (PLE ≈ 0.041 vs. ~ 0.49 for ε), (ii) slope error near poles (≈ 0.90 vs. ~ 1.00), and (iii) residual consistency (≈ 0.012 vs. ~ 0.082).

Table. We aggregate per-bucket MSE (means and counts) for all methods from our evaluator into a CSV (`results/robotics/aggregated/buckets_table.csv`). In the camera-ready version we include the B0–B4 table (means and counts) for: MLP, Rational+ε (3 values), TR-Basic, TR-Full, and Rational+ε+Clip.

Minimal figure. We include a single near-pole B0 scatter that shows the spread of $\|\Delta\theta\|$ at $|\det J| \leq 10^{-5}$ for the ε grid versus TR (`results/robotics/figures/pole_b0_scatter.png`). This complements the table by illustrating ε-sensitivity near poles while TR remains tightly bounded.

6.5.1 Takeaway

Clipping reduces magnitude but does not eliminate ε-dependence at $Q \approx 0$; TR achieves lower near-pole error and improved pole metrics without any ε.

6.5.2 Analysis

The dramatic improvements in B0–B1 arise because ε-regularization forces $P(x)/(Q(x) + \varepsilon) \approx P(x)/\varepsilon$ when $|Q(x)| \ll \varepsilon$, creating ε-dependent plateaus. TR preserves the true singular structure via explicit tags and Mask-REAL gradients, allowing the model to shape the numerator behavior near poles rather than masking the denominator. The observed 9–20× error reduction shows that *getting singularities right* matters most precisely where traditional methods are least reliable.

6.6 Control Rollouts Near Singularities (2R)

Goal. Convert per-sample accuracy into rollout stability by executing differential IK in a short control loop that skims near-singular regions.

Setup. We train quick models (MLP 1 epoch; Rational+ε 3 epochs with grid
$10^{-4}, 10^{-3}$; TR-Basic/Full 3 epochs) on the original data with 1k train samples, then roll out $N = 4$ trajectories, $T = 30$ steps each. At each step we command a small task-space displacement ($\|\Delta x, \Delta y\| \approx 8 \times 10^{-3}$) in fixed directions (0°, 90°), predict $[\Delta\theta_1, \Delta\theta_2]$,

integrate (θ_1, θ_2), and measure: (i) mean tracking error $\|\Delta x, \Delta y\|$ mismatch, (ii) max joint step (proxy for actuator saturation), and (iii) failure rate (% steps producing non-REAL tags or NaN/Inf outputs).

Method	Mean Tracking Err.	Max $\|\Delta\theta\|$	Failure Rate (%)
MLP	0.2031	2.0987	0.00
Rational+ε	0.0103	0.0073	0.00
ZeroProofML (Basic)	0.0381	0.0165	0.00
ZeroProofML (Full)	0.0306	0.0165	0.00

Table 11: Control-style rollout near $\theta_2 \approx 0$: mean task-space tracking error, max joint step, and failure rate over $N \times T = 120$ steps per method.

TR models exhibit small joint steps and low tracking error without failures at this step size; MLP shows substantially larger steps and error. The ε-rational baseline achieves the lowest error here, while TR remains within a small factor with bounded updates. These rollouts complement E1/E2/E3 by demonstrating closed-loop stability indicators near poles.

6.6.1 Discussion

TR prioritizes stability near poles (bounded $\|\Delta\theta\|$, zero failures) and remains within a small factor of Rational+ε on mean error. This trade-off aligns with safety-oriented objectives and complements the strong near-pole accuracy observed in the per-bucket analysis.

6.7 Extra Baseline: Rational+ε with Gradient Clipping

Motivation. Reviewers asked whether modern stabilizers like gradient clipping could resolve ε sensitivity. We add global-norm clipping ($c = 1.0$) to the Rational+ε trainer.

Findings. Clipping shrinks near-pole magnitudes (B0–B2) relative to the unclipped ε baseline, but still underperforms TR in those bins (e.g., B0: 8.5×10^{-3} vs. TR $\sim 3.8 \times 10^{-3}$). Pole metrics remain far from TR (e.g., PLE ~ 0.51 vs. ~ 0.041).

6.7.1 Conclusion

Clipping reduces magnitude but does not remove ε-dependence at $Q \approx 0$ or match TR's near-pole accuracy; TR's tag-aware, totalized semantics address the underlying issue without ε.

7 Reproducibility

All experiments are scripted. Below we list the key commands for the new stress tests; seed determinism explains occasional zero standard deviations.

7.0.1 Note on variance

Several experiments show zero standard deviation due to deterministic initialization, fixed seeds, and data ordering in our reproducibility harness. This demonstrates TR's reproducibility advantage but may understate variance in less controlled settings. We therefore report means and counts per bucket and focus analysis on robust near-pole and rollout metrics; across-seed aggregates for non-deterministic baselines (MLP, Rational+ε) are provided in Appendix 12.

7.0.2 Dataset (E1/E2)

```
python examples/robotics/rr_ik_dataset.py \
  --n_samples 20000 \
  --singular_ratio 0.35 \
  --displacement_scale 0.1 \
  --singularity_threshold 1e-3 \
  --stratify_by_detj --train_ratio 0.8 \
  --force_exact_singularities \
  --min_detj 1e-6 \
  --bucket-edges 0 1e-5 1e-4 1e-3 1e-2 inf \
  --ensure_buckets_nonzero \
  --seed 123 \
  --output data/rr_ik_dataset.json
```

7.0.3 A: Exact-pole dataset and training

Dataset (exact poles, stratified B0–B4):

```
python examples/robotics/rr_ik_dataset.py \
  --n_samples 4000 --singular_ratio 0.35 \
  --displacement_scale 0.08 --singularity_threshold 1e-3 \
  --stratify_by_detj --train_ratio 0.8 \
  --force_exact_singularities --ensure_buckets_nonzero \
  --seed 777 --output data/rr_ik_dataset_exact.json
```

Rational+ε (no-clip grid):

```
for EPS in 1e-2 1e-3 1e-4; do
  python examples/robotics/rr_ik_train.py \
    --dataset data/rr_ik_dataset_exact.json \
    --model rat_eps --epochs 10 \
    --degree_p 3 --degree_q 2 --learning_rate 1e-2 \
    --epsilon $EPS --log_every 5 \
    --output_dir runs/ik_eps_noclip_${EPS//./p}
done
```

TR-Basic/Full:

```
python examples/robotics/rr_ik_train.py --dataset data/rr_ik_dataset_exact.json \
  --model tr_rat --epochs 10 --learning_rate 1e-2 \
  --degree_p 3 --degree_q 2 --no_hybrid --no_coverage \
  --no_structured_logging --no_plots --log_every 5 \
  --output_dir runs/ik_tr_basic

python examples/robotics/rr_ik_train.py --dataset data/rr_ik_dataset_exact.json \
  --model tr_rat --epochs 8 --learning_rate 1e-2 \
  --degree_p 3 --degree_q 2 --no_structured_logging \
  --no_plots --log_every 5 --output_dir runs/ik_tr_full
```

Clipping baseline:

```
python examples/robotics/rr_ik_train.py --dataset data/rr_ik_dataset_exact.json \
  --model rat_eps --epochs 10 --degree_p 3 --degree_q 2 \
  --learning_rate 1e-2 --epsilon 1e-3 --clip_grad 1.0 \
  --log_every 5 --output_dir runs/ik_eps_1e-3
```

Per-bucket evaluation and aggregation:

```
python scripts/evaluate_trainer_buckets.py \
  --dataset data/rr_ik_dataset_exact.json \
  --results runs/ik_tr_full/results_tr_rat.json \
  --out runs/ik_tr_full/buckets.json

python scripts/aggregate_buckets.py --scan runs \
  --out results/robotics/aggregated/buckets_table.csv
```

Near-pole (B0) scatter:

```
python scripts/pole_exact_scatter.py \
  --dataset data/rr_ik_dataset_exact.json \
  --results runs/ik_eps_noclip_1e-2/results_rat_eps.json \
           runs/ik_eps_noclip_1e-3/results_rat_eps.json \
           runs/ik_eps_noclip_1e-4/results_rat_eps.json \
           runs/ik_tr_basic/results_tr_rat.json \
           runs/ik_tr_full/results_tr_rat.json \
  --labels  "eps=1e-2 (no clip)" "eps=1e-3 (no clip)" "eps=1e-4 (no clip)" \
           "TR Basic" "TR Full" \
  --detj_thresh 1e-5 \
  --out results/robotics/figures/pole_b0_scatter.png
```

7.0.4 B: Closed-loop rollouts

```
python scripts/rollout_near_singularity.py \
  --dataset data/rr_ik_dataset_exact.json \
  --n_steps 20 --n_traj 2 --max_train 1000 \
  --out results/robotics/rollout_summary.json
```

7.0.5 E1 quick parity (5 seeds)

```
for SEED in 101 202 303 404 505; do
python experiments/robotics/run_all.py \
  --dataset data/rr_ik_dataset.json \
  --profile quick \
  --models tr_basic tr_full rational_eps mlp dls \
  --max_train 2000 --max_test 500 \
  --seed $SEED \
  --output_dir results/robotics/quick_s${SEED}
done
python scripts/aggregate_parity.py
```

7.0.6 E1 full parity (1 seed)

```
python experiments/robotics/run_all.py \
  --dataset data/rr_ik_dataset.json \
  --profile full \
  --models tr_basic tr_full rational_eps mlp dls \
  --output_dir results/robotics/full_s123 \
  --seed 123
```

7.0.7 E4 robustness harness

```
python scripts/e3_robustness.py \
  --orig data/rr_ik_dataset.json \
  --shifted data/rr_ik_dataset_shifted.json \
  --outdir results/robotics/e3_robustness \
  --max_train 2000 --max_test 500
```

7.0.8 E2 ablations

```
# Mask-REAL only
python examples/robotics/rr_ik_train.py \
  --dataset data/rr_ik_dataset.json \
  --model tr_rat \
  --epochs 40 \
  --learning_rate 1e-2 \
  --degree_p 3 --degree_q 2 \
  --no_hybrid \
  --no_coverage \
  --output_dir results/robotics/ablation_mask_real

# Hybrid + coverage
python examples/robotics/rr_ik_train.py \
  --dataset data/rr_ik_dataset.json \
  --model tr_rat \
  --epochs 20 \
  --learning_rate 1e-2 \
  --degree_p 3 --degree_q 2 \
  --output_dir results/robotics/ablation_hybrid

# Optional: supervise pole head
python examples/robotics/rr_ik_train.py \
  --dataset data/rr_ik_dataset.json \
  --model tr_rat \
  --epochs 20 \
  --learning_rate 1e-2 \
  --degree_p 3 --degree_q 2 \
  --supervise-pole-head --teacher_pole_threshold 0.1 \
  --output_dir results/robotics/ablation_hybrid_supervised

# Per-bucket evaluation for ablations
python scripts/evaluate_trainer_buckets.py \
  --dataset data/rr_ik_dataset.json \
  --results results/robotics/ablation_hybrid/results_tr_rat.json
```

7.0.9 E4 robustness

See commands in Section 6 (E3). We reuse the trained models from E1/E2 and swap in the shifted test split for evaluation.

7.0.10 Artifacts

Per-seed summaries and JSONs are under `results/robotics/quick_s*`, with consolidated outputs in `comparison_table.csv` and `comprehensive_comparison.json`. Ablation outputs (including `training_summary`, `bench_history`, and bucket metrics) are in `results/robotics/ablation_*`.

8 Conclusion

ZeroProofML reframes rational neural layers around *deterministic, ε-free* semantics. By totalizing singular operations with explicit tags and a tag-aware autodiff, we remove ε tuning and deliver reproducible behavior at and near poles. On 2R IK we match overall accuracy while achieving **9–20× lower error in near-singularity buckets (**$|\det J| \leq 10^{-5}$**)** compared to ε-rational and clipping baselines, alongside better pole metrics (PLE, slope, residual) and bounded rollout updates with zero failures. These results argue for determinism and reproducibility as *first-class* objectives in rational architectures. Future work extends to higher-DOF manipulators, checkpointed tag analyses on analytic pole sets, and system-level integration where TR's stability can simplify safety cases.

9 Limitations and Outlook

- **Domain specificity**: Evaluated on 2R IK with analytically known singular structure. Performance on learned singularities in other domains (control, physics, scientific ML, geometric DL, normalization layers) requires investigation.

- **Computational overhead**: Our Python reference adds ∼2× overhead vs ε-rational due to value+tag propagation. Production deployments would benefit from optimized kernels/JIT to amortize TR operations.

- **Training complexity**: Hybrid gradient modes, coverage control, and optional pole heads add implementation complexity relative to simple ε. Our code provides reproducible templates; further simplification and ablations remain.

- **Evaluation scope**: We focus on B0–B2 pathology, PLE/slope/residual metrics, and short rollouts. Longer-horizon closed loops and stronger stressors are promising future work.

10 Broader Impact and Ethics

Safer handling of singularities can reduce training instabilities and numerical hacks in safety-critical applications (e.g., robotics). However, replacing ad-hoc ε with transreal semantics does not eliminate all failure modes: dataset bias and modeling assumptions still apply. Care must be taken to validate behavior under distribution shifts and to avoid overconfidence near singular regions.

Code and Data Availability

ZeroProofML is open source under the MIT License: github.com/domezsolt/ZeroProofML (commit `5c4cb80`). The repository contains dataset generators, training and evaluation scripts, aggregated CSVs, LaTeX tables, and figures to reproduce the results in this paper. A frozen release will be archived upon camera-ready submission.

Appendix A. Across-Seed Dispersion (Quick Profile)

Across-seed mean±std for non-deterministic baselines (MLP, Rational+ε) on the exact dataset under the quick profile; TR runs are deterministic under our harness.

Method	Overall MSE	B0	B1	B2
MLP	-	0.193374 ± 0.145417 (n=5)	0.115849 ± 0.105043 (n=5)	0.154949 ± 0.064419 (n=5)
Rational+ε	-	0.005202 ± 0.000161 (n=5)	0.003726 ± 0.000362 (n=5)	0.064313 ± 0.010814 (n=5)

Table 12: Across-seed mean±std (3 seeds) for overall MSE and near-pole buckets on the exact dataset.

Appendix B. Metrics and Evaluator Parameters

B.0.1 Bucketization by $|\det J|$

We stratify samples by $|\det J| \approx |\sin\theta_2|$ using edges $[0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, \infty)$ to define buckets B0–B4. Per-bucket MSE is the mean of per-sample MSEs within each bin.

B.0.2 Quick subset selection

For the quick profile (2k/500), we construct the test subset by: (i) preselecting one sample from each of B0–B3 (if available), then (ii) round-robin filling across buckets until reaching the target size. Training takes the first 2k samples from the 80% train split for speed.

B.0.3 Pole Localization Error (PLE)

Following `compute_ple_to_lines`, we select the top-k samples by $\|\Delta\theta\|$ (default $k = \lceil 0.05\,N \rceil$) and measure the mean angular distance from $\theta_2 \in \{0, \pi\}$ (wrapped to $[-\pi, \pi]$).

B.0.4 Sign consistency (window-based)

Using `compute_sign_consistency_rate`, we choose n_{paths} evenly spaced θ_1 anchors and collect near-crossing samples with $|\theta_1 - \text{anchor}| \le \tau_1$ and $|\theta_2| \le \tau_2$. We report the fraction of anchors whose dominant sign of $\Delta\theta_2$ flips across $\theta_2 = 0$, ignoring near-zero magnitudes $|\Delta\theta_2| \le \tau_{\min}$. Defaults used in E1 quick: $n_{\text{paths}} = 12$, $\tau_1 = 0.15$, $\tau_2 = 0.30$, $\tau_{\min} = 10^{-3}$; tightened variants are reported in the appendix figures.

B.0.5 Paired sign consistency

To isolate true crossings, we pair per-anchor the k closest negative/positive θ_2 samples by $|\theta_2|$ and count flips of $\operatorname{sign}(\Delta\theta_2)$ with $|\Delta\theta_2| > \tau_{\min}$. We aggregate the fraction over anchors and seeds. Defaults: $k = 3$, $\tau_{\min} = 10^{-3}$.

B.0.6 Slope error near poles

We fit a line to $(x, y) = \big(\log_{10}(\max\{\epsilon, |\sin\theta_2|\}), \log_{10}(\|\Delta\theta\|)\big)$ for samples with $|\sin\theta_2| \le 10^{-2}$ and report $|\hat{\beta} + 1|$ (ideal slope ≈ -1). Defaults: $\epsilon = 10^{-6}$, near-pole cutoff 10^{-2}, minimum 5 points.

B.0.7 Residual consistency

Using forward kinematics for the 2R arm ($L_1 = L_2 = 1$), we compute the mean squared residual between target $(\Delta x, \Delta y)$ and the displacement induced by predicted $\Delta\theta$.

B.0.8 E4 harness settings

For reproducibility: MLP (1 epoch, ReLU, [32,16], lr=1e-2), Rational+ε (grid $\{10^{-4}, 10^{-3}\}$, 3 epochs, lr=1e-2), ZeroProofML Basic/Full (3 epochs, lr=1e-2; Full uses Hybrid+tag/pole/residual losses). Train on original (2k), evaluate on original vs shifted (500 each).

Appendix C. Supplement: Direction-fixed Sweeps

We evaluate sign consistency on near-crossing samples while fixing the displacement direction to increase usable pairs. The table reports mean±std over 5 runs with average contributing pairs in parentheses.

Setting	MLP	Rational+ε	TR-Basic	TR-Full
$\phi = 60°$, tol=35°, $\lvert\theta_2\rvert \le 0.35$	10.00 ± 5.00 (8.0)	0.00 ± 0.00 (8.0)	0.00 ± 0.00 (8.0)	0.00 ± 0.00 (8.0)
$\phi = 90°$, tol=35°, $\lvert\theta_2\rvert \le 0.35$	14.29 ± 18.07 (6.6)	0.00 ± 0.00 (7.0)	0.00 ± 0.00 (7.0)	0.00 ± 0.00 (7.0)

Table 13: Direction-fixed paired consistency (%); average contributing pairs in parentheses. Parameters: $k = 4$, $|\Delta\theta_2| > 5 \times 10^{-4}$.

Appendix D. Reproducibility Checklist

- Seeds and splits: fixed seeds; deterministic, stratified quick subsets (2k/500) with B0–B3 preselection and round-robin fill.
- Environment: Python/Torch versions, determinism flags, and commit hash recorded in result JSONs.
- Data hashes: dataset SHA-256 stored with comprehensive results; shifted dataset generated via CLI with explicit flags.
- Artifacts: per-seed JSONs, aggregated summaries, and figures under results/robotics/; CSVs where applicable.
- Config provenance: all hyperparameters (epochs, lr, degrees, schedules) included in result JSONs for exact replay.

Acknowledgments

We thank contributors to the open-source ZeroProofML codebase and reviewers for constructive feedback.

References

James A. D. W. Anderson. *Transmathematics*. Independently published, 2019. Please verify publisher/edition before submission.

Atilim G"uneş Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

Jan A. Bergstra and Cornelis A. Middelburg. Division by zero and totalized algebra (meadows/wheel). 2021. Placeholder citation; please replace or adjust to the appropriate reference on the Wheel/meadows framework.

Nicolas Boulle, Thomas Gallou"et, and Yvon Maday. Rational neural networks. In *Advances in Neural Information Processing Systems*, 2020. Please verify venue/authors before submission.

[Author to verify] dos Reis. Transreal arithmetic. 2016. Placeholder citation; please replace with the correct bibliographic details.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Paulius Micikevicius, Sharan Narang, Jonah Alben, et al. Mixed precision training. In *International Conference on Learning Representations*, 2018.