extensity AI

Reinforcement Learning: Where We Are and What's Next

19th September 2025

Marius-Constantin Dinu

2025

Speaker

Dr. techn. Marius-Constantin Dinu

JOHANNES KEPLER
UNIVERSITÄT LINZ

TEACHING

- Deep Reinforcement Learning Exercises: 2021 – 2023 PPO & Offline RL part

PUBLICATIONS, TALKS & WORKSHOPS

- NeurIPS, CoLLAs, ICML, AIP, AIRoV, ICLR, CoLLAs, ANSyA
- Plug and Play Vienna
- Workshops at IARAI, JinaAI, ExtensityAI
- AI Austria RL Community Workshop

R&D EXPERIENCES

- 10+ years Fullstack Developer & SW Architect
- 5+ years Researcher in academia and industry
- 1.5 years Sr. Researcher at Dynatrace
- 1 year at Atlas as Sr. AI Research Scientist
- 4 months CEO & Research Director at ExtensityAI

FOCUS

- Continual Learning
- Deep Reinforcement Learning
- Domain Adaptation
- Natural Language Processing
- Neuro-Symbolic AI

extensity AI

Reasoning Over Literature

- Daily driver for researchers
- Literature linking, citation chaining, discovery
- Project management, writing assistance
- Alpha-stage, already showing high value

Automate Research to Drive Insights and Enhance Efficiency

BEFORE: SIX DISCONNECTED STEPS

1. Search

Teams manually scour internal and external sources for patents, competitor data, and market insights.

2. Read

Researchers spend hours interpreting dense information, often redundant or irrelevant.

3. Summarize

Findings are manually condensed into key points or trends.

4. Draft

Internal whitepapers, feasibility studies, and reports are created from scratch.

5. Review

Documents go through iterative revisions with multiple stakeholders.

6. Finalize

Reports are formatted and aligned for distribution, delaying decision-making.

AFTER: THREE SEAMLESS STEPS

1. AI Search

Extensity's Knowledge Engine automatically aggregates, filters, and organizes relevant data into a central hub.

2. Synthesize

AI generates draft reports, whitepapers, or summaries in minutes, tailored to corporate needs.

3. Finalize with Collaboration

Teams make final edits collaboratively with AI-assisted review, ready for immediate action.

AI Generated Research | Primality Testing

EXPERIMENTAL MATHEMATICS

Primality Testing via Circulant Matrix Eigenvalue Structure: A Novel Approach Using Cyclotomic Field Theory

Marius-Constantin Dinu*
ExtensityAI Austria

ARTICLE HISTORY
Compiled April 28, 2025

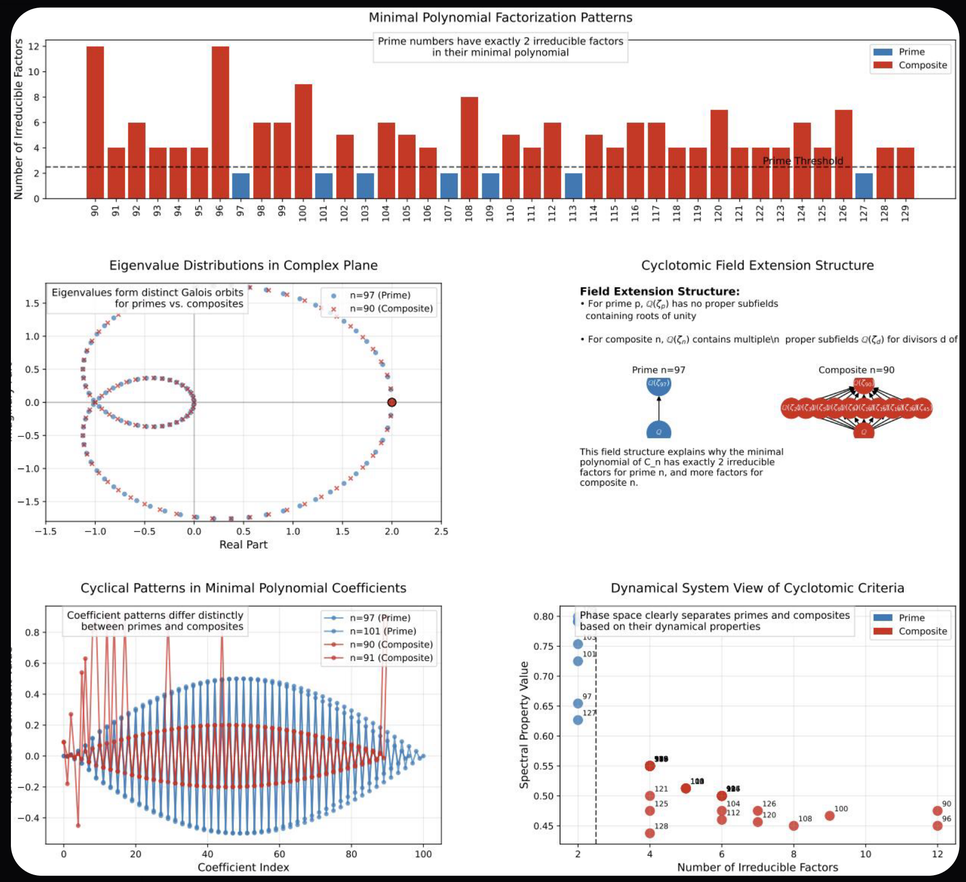
ABSTRACT
This paper presents a novel primality test based on the eigenvalue structure of circulant matrices constructed from roots of unity. We prove that an integer $n > 2$ is prime if and only if the minimal polynomial of the circulant matrix $C_n = W_n + W_n^2$ has exactly two irreducible factors over $\mathbb{Q}$. This characterization connects cyclotomic field theory with matrix algebra, providing both theoretical insights and practical applications. We demonstrate that the eigenvalue patterns of these matrices reveal fundamental distinctions between prime and composite numbers, leading to a deterministic primality test. Our approach leverages the relationship between primitive roots of unity, Galois theory, and the factorization of cyclotomic polynomials. We provide comprehensive experimental validation across various ranges of integers, discuss practical implementation considerations, and analyze the computational complexity of our method in comparison with established primality tests. The visual interpretation of our mathematical framework provides intuitive understanding of the algebraic structures that distinguish prime numbers. Our experimental validation demonstrates that our approach offers a deterministic alternative to existing methods, with performance characteristics reflecting its algebraic foundations.

KEYWORDS
Circulant matrices; cyclotomic fields; eigenvalue structure; Galois theory; minimal polynomials

1. Introduction

Distinguishing prime numbers from composite numbers has been a central challenge in mathematics for millennia. While numerous primality tests exist, from the ancient sieve of Eratosthenes to modern probabilistic algorithms like Miller-Rabin [13] and deterministic methods like AKS [1], the discovery of new connections between primality and other mathematical structures continues to provide insights into the fundamental nature of prime numbers. While building upon classical foundations in cyclotomic field theory, our approach provides a matrix-theoretic perspective that yields both theoretical insights and practical applications. This paper establishes a novel characterization of primality through the lens of circulant matrices and cyclotomic field theory. We prove that an integer $n > 2$ is prime if and only if the minimal polynomial of the circulant matrix $C_n = W_n + W_n^2$ has exactly two irreducible factors over the rational field $\mathbb{Q}$, where W_n represents the $n \times n$ circulant matrix associated with the n-th roots of unity. Our work is motivated by the desire to uncover new structural properties that characterize prime numbers, contributing to our fundamental understanding of number theory. This research bridges the gap between classical results in cyclotomic field theory and modern computational approaches to primality testing. The connection between cyclotomic fields and primality established herein may lead to new insights in algebraic number theory and Galois theory. The practical implications of our findings extend beyond theoretical interest. Our approach has potential applications in

*This paper was created with AI assistance using Symbia Engine from SymbolicAI Framework [17].
Contact author email: marius@extensity.ai; Repository: https://github.com/ExtensityAI/primality_test



https://github.com/ExtensityAI/primality_test

extensity AI

AI Generated Research | Three-Body Problem

EXPERIMENTAL PHYSICS

A Unified Framework for the Three-Body Problem: Connecting Differential Galois Theory, Painlevé Analysis, and Quaternionic Regularization

Marius-Constantin Dinu*
ExtensityAI Austria

ARTICLE HISTORY
Compiled May 4, 2025

ABSTRACT
We present a unified theoretical framework that establishes rigorous isomorphisms between three distinct mathematical approaches to the three-body problem: Differential Galois Theory, Painlevé Analysis, and Quaternionic Regularization. Our framework proves precise mathematical correspondences between the algebraic structure of differential Galois groups, the analytic branching behavior in Painlevé analysis, and the geometric monodromy of quaternionic continuation paths. Through these isomorphisms, we demonstrate that the non-abelian nature of the Galois group is equivalent to the branching structure of complex solutions, which in turn determines the monodromy of quaternionic regularization paths. This theoretical synthesis provides deep insights into the exceptional mass ratios that yield partially integrable systems, which we demonstrate through rigorous mathematical proofs for homothetic orbits and Lagrangian solutions. By establishing the equivalence of algebraic obstructions to integrability with geometric properties of quaternionic space, our framework enables consistent mathematical characterization of near-collision trajectories while preserving the system's conservation laws. The isomorphisms yield practical benefits for predicting the stability of triple star systems and multi-planet configurations through identification of common mathematical structures across the three perspectives. Beyond the three-body problem, our framework establishes a general correspondence between algebraic integrability criteria, analytic singularity structure, and geometric regularization techniques applicable to broader classes of dynamical systems.

1. Introduction

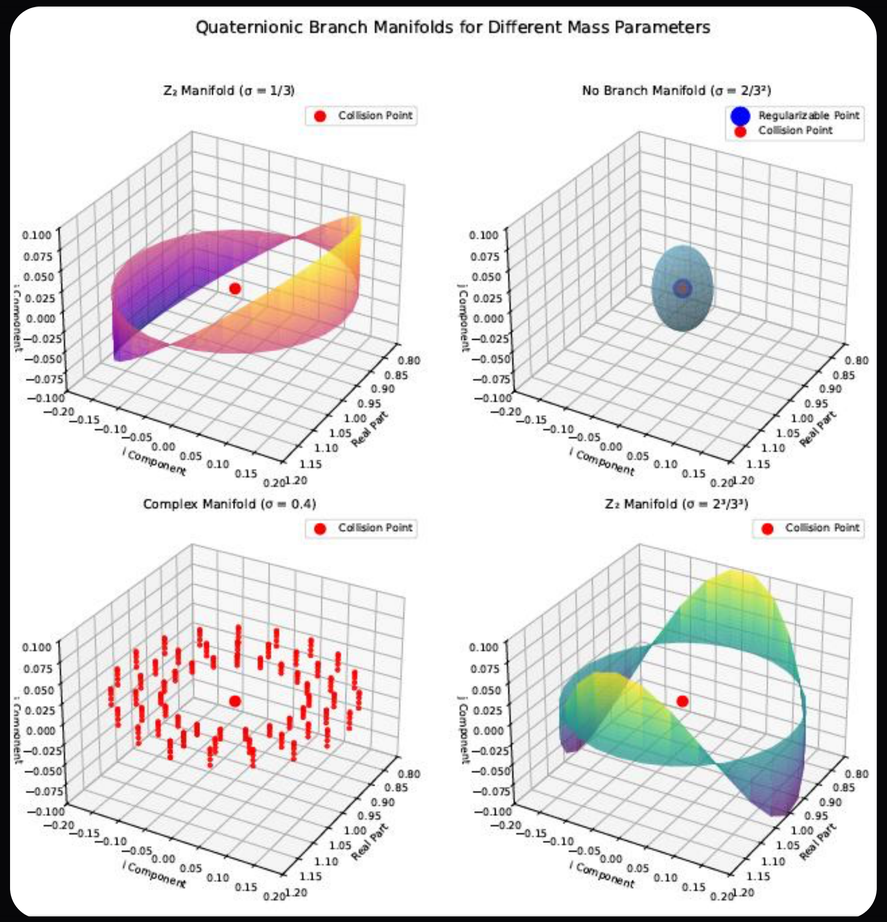
The three-body problem—describing the motion of three bodies under mutual gravitational attraction—remains one of the fundamental challenges in mathematical physics. Despite extensive study since Newton, a complete characterization of its solution space continues to elude us. The problem transcends astronomy, raising profound questions about deterministic systems, integrability, and the nature of chaos in dynamical systems.

At the heart of this challenge lies the question of integrability: does the system admit enough independent conserved quantities to determine its trajectories through quadratures? Poincaré's groundbreaking work established that the three-body problem is not integrable in the general case [22], exhibiting chaotic behavior for most initial conditions. Yet precise boundaries between integrable and non-integrable cases—particularly for special mass distributions and configurations—remain incompletely understood.

Three distinct mathematical approaches have traditionally been applied to this problem, each offering valuable but limited insights:

(1) **Differential Galois Theory (DGT)**: An algebraic approach that examines the structure of differential field extensions generated by solutions to variational equations. The Morales-Ramis-Simó theorem [1, 2] establishes that meromorphic integrability requires an abelian

*This paper was created with AI assistance using Symbia Engine from SymbolicAI Framework [34].
Contact author email: marius@extensity.ai; Repository: https://github.com/ExtensityAI/three-body_problem



https://github.com/ExtensityAI/three-body_problem

extensity AI

Material Science Temperature Analysis

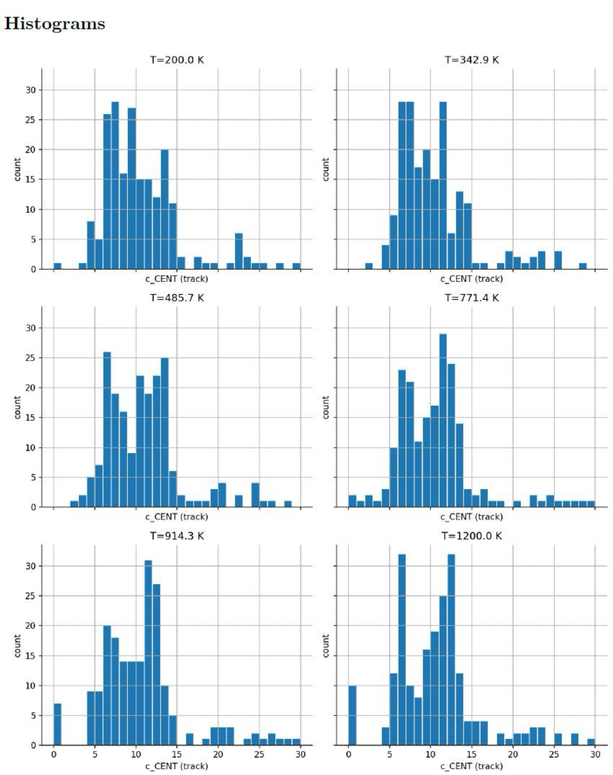
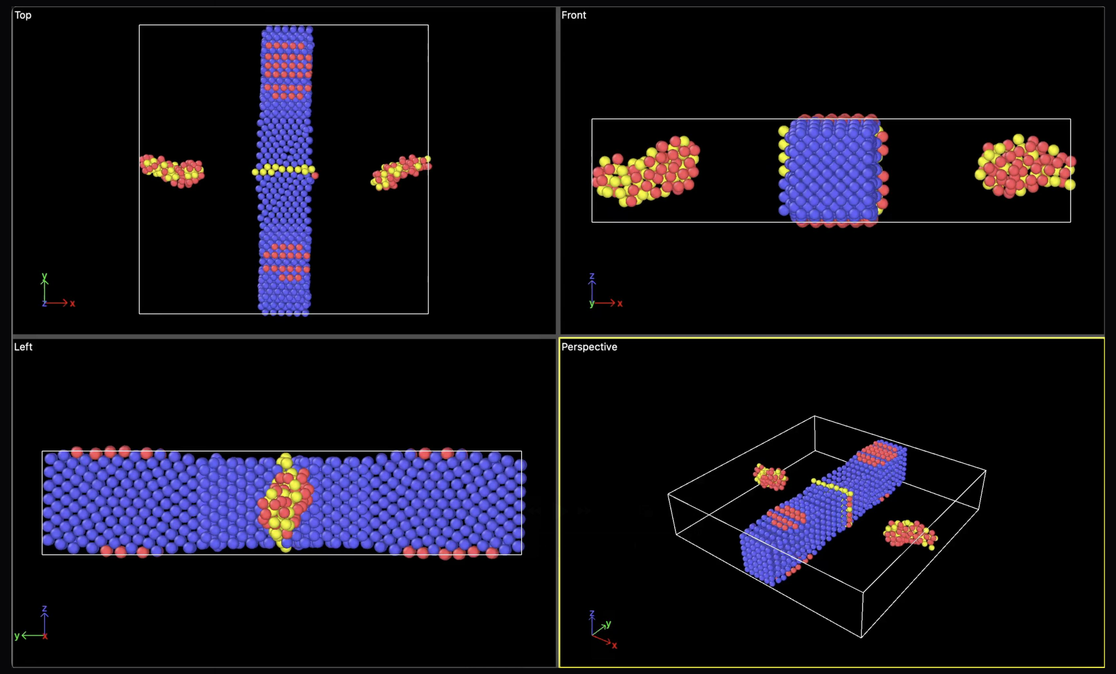


Figure 4: Distribution of centrosymmetry values across temperatures.

The histograms display the distribution of centrosymmetry values, a measure of atomic arrangement disorder, at various temperatures. The x-axis represents centrosymmetry values, while the y-axis shows frequency. The distributions shift towards higher disorder with increasing temperature, indicating structural breakdown.

Conclusion: Centrosymmetry histograms provide qualitative support for the damage metric, illustrating atomic-level disorder at elevated temperatures.



extensity AI

Motivation. Why *research automation* with neurosymbolic + agentic LLM systems connects to RL.

Key idea. Research automation orchestrates LLM tools, symbolic reasoning, retrieval, and verifiers to produce *checkable* artifacts (proofs, code, experiments). This creates *closed-loop* tasks with measurable outcomes—fertile ground for RL.

- **Pipeline.** (i) Task spec $\rightarrow$ (ii) Planner (symbolic & LLM) $\rightarrow$ (iii) Tool use (retrieval, solvers, code) $\rightarrow$ (iv) *verifiers/tests* $\rightarrow$ (v) memory & iteration.
- RL enters when outcomes are **verifiable**. Preference learning and verifiable rewards provide the objective signal.

What Agents Are (and Are Not)

Focus. Clarify terminology: an LLM *alone* is not an agent.

Agent formalism. An MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ plus a policy $\pi(a|s)$, an *action space*, *state*, and a *goal* (reward). Many LLM wrappers are scripts without well-defined P or r.

$$\pi^\star \in \arg\max_\pi \; J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

LLM + tools $\neq$ agent unless embedded in a perception–action loop:

State s_t (context/tools) $\rightarrow$ LLM proposes action a_t (tool call/code/plan) $\rightarrow$ Env & verifiers yield r_t and next s_{t+1}.

Takeaway. Use *agent* when the loop, action space, and goals are made explicit.

Example Agent (CLI via `codex`)

```
codex --full-auto -m gpt-5 \
  --config model_reasoning_effort="high" \
  --config model_reasoning_summary="detailed" \
  --config verbosity="high" \
  --config wire_api="responses"
```

What this does. Launches a fully autonomous agent loop (plan $\rightarrow$ act $\rightarrow$ verify) using model `gpt-5`, with elevated reasoning effort and detailed run summaries, verbose logging, and the Responses wire API.

Flag / Key	Type / Values	Meaning
`--full-auto`	boolean switch	Enables autonomous execution without manual confirmation; the agent iterates tools/actions until termination criteria (goal met, budget/time).
`-m` *gpt-5*	string (model id)	Selects the base policy/model used for planning, tool calls, and generation.
`model_reasoning_effort`	enum: *low—medium—high*	Hints the runtime to allocate more internal budget to multi-step reasoning and tool orchestration (may increase cost/latency).
`model_reasoning_summary`	enum: *off—brief—detailed*	Controls the *summary* of reasoning shown in logs/artifacts (not the private scratchpad); *detailed* emits richer rationales and step traces.
`verbosity`	enum: *low—normal—high—debug*	Log detail level (requests, tool I/O, decisions). Use *high* for ops, *debug* for development.
`wire_api`	enum: *chat—responses*	Chooses the client/server protocol. *responses* supports structured tool calls and richer metadata.

Training Agents

Figure 1: Agent training pipeline: pretraining → SFT → post-training (RLHF/DPO/verifier-RL) feeding into deployed agents, with feedback signals from humans and automated checkers shaping post-training.

Greatest Commercial Success of RL (so far)

Claim. The largest commercial impact of RL is as a *post-training algorithm* for LLMs (alignment, reasoning, safety) via RLHF-style methods and successors.

- **RLHF line:** reward models trained from human preferences; RL (e.g., PPO) with a KL constraint against a reference LM.
- **Direct preference optimization:** avoids explicit RL loop; closes the gap to RLHF on many alignment tasks.
- **Verifiable-reward RL:** uses *tests* (compilers, unit tests, math checkers) as rewards—key driver for reasoning and coding gains.

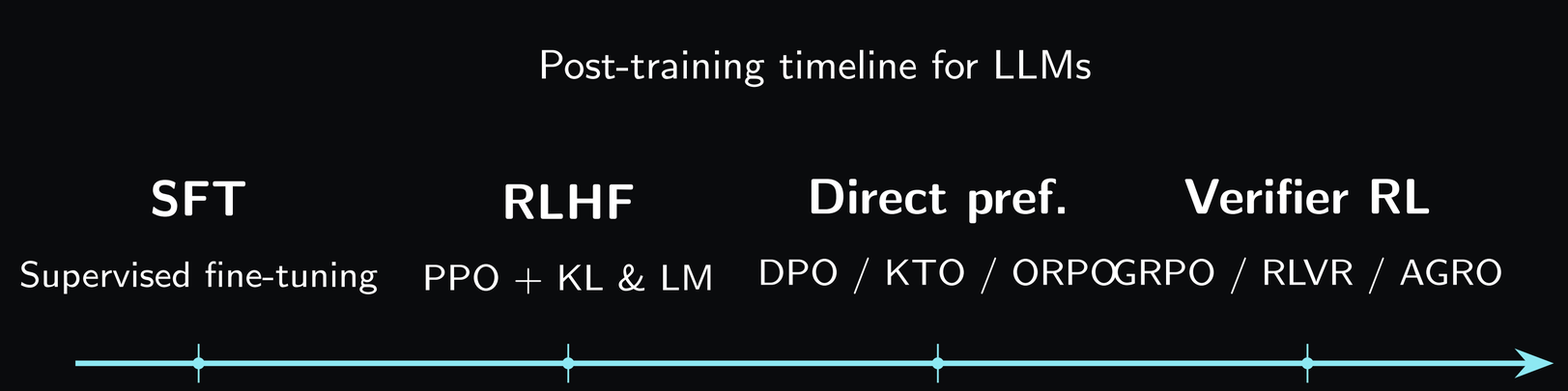


Figure 2: Preference- and verifier-driven RL have become core to modern LLM training.

RL Fine-Tuning for LLMs: The Basics

Focus. How RL is used to fine-tune an LM policy π_θ.

Objective (KL-regularized):

$$\max_\theta \underbrace{\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[\, r_\phi(x,y)\,]}_{\text{reward model / verifier}} - \beta\, \mathbb{E}_{x\sim\mathcal{D}}\Big[\mathrm{KL}\big(\pi_\theta(\cdot\mid x)\,\|\,\pi_{\mathrm{ref}}(\cdot\mid x)\big)\Big]$$

Interpretation: Improve reward while staying close to a safe base model π_{ref}.

PPO-style update (token-level weighting):

$$\theta \leftarrow \arg\max_\theta\ \mathbb{E}\Big[\,\min\big(\rho_t(\theta)\hat{A}_t,\ \mathrm{clip}(\rho_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t\big)\ -\ \beta\,\mathrm{KL}_t\Big]$$

with $\rho_t = \frac{\pi_\theta(y_t|h_t)}{\pi_{\theta_{\mathrm{old}}}(y_t|h_t)}$ and $\hat{A}_t$ from reward model rollouts.

Notes. KL is essential (prevents divergence/degeneracy); reward shaping (length penalties) often required for LLMs.

KL-Regularized Objective

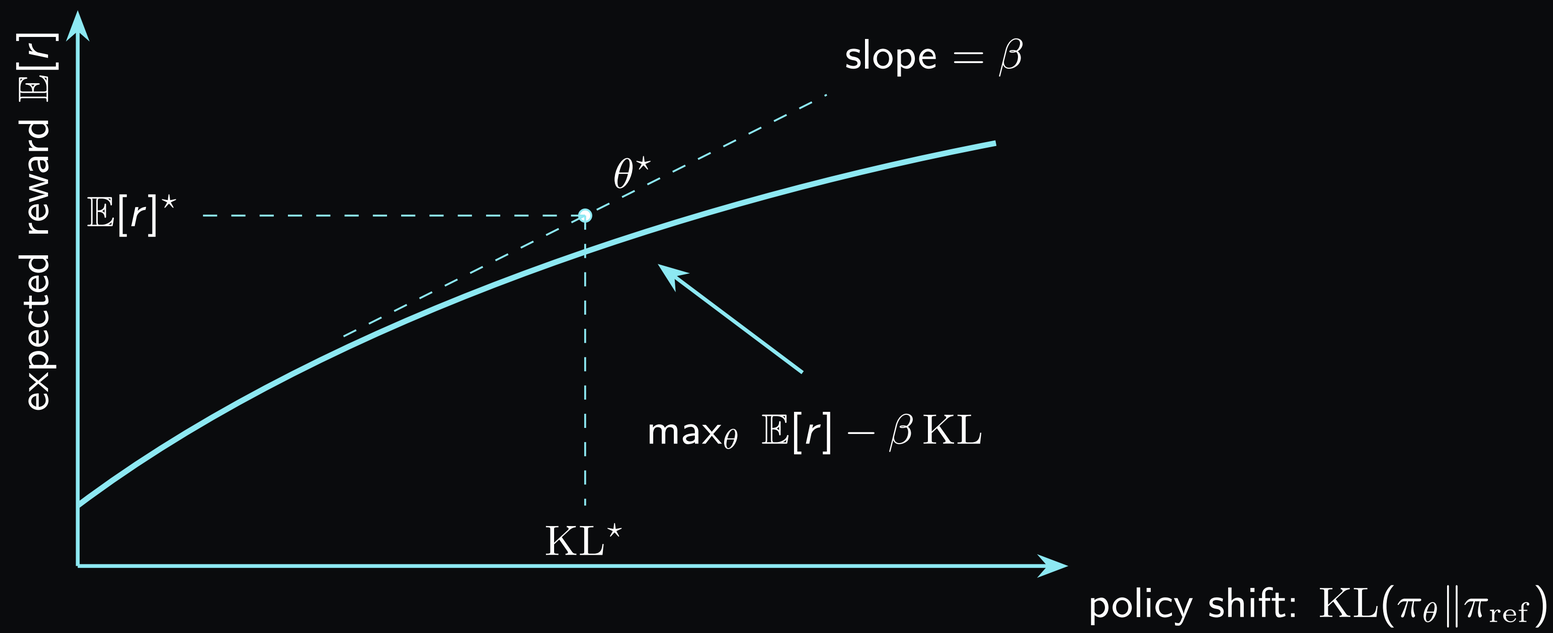


Figure 3: KL-regularized objective: balance task reward and divergence from a reference model. The optimum $\theta^\star$ occurs where the frontier's tangent has slope β, yielding $(\mathrm{KL}^\star, \mathbb{E}[r]^\star)$.

Pseudocode: PPO with KL for LLMs

Focus. Minimal recipe many teams actually run.

Algorithm 1 KL-regularized PPO for LLM post-training

Initialize π_θ, reference π_{ref}, reward model r_ϕ
for iter $= 1..T$ **do**
    Sample prompts $x \sim \mathcal{D}$; generate K responses $y^{(k)} \sim \pi_\theta(\cdot \mid x)$
    Compute scalar rewards $R^{(k)} \leftarrow r_\phi(x, y^{(k)})$ (or from verifiers/tests)
    Compute token-level advantages $\hat{A}_t$ (e.g., GAE on per-token deltas)
    Compute per-token KL $d_t = \log \frac{\pi_\theta(y_t|h_t)}{\pi_{\mathrm{ref}}(y_t|h_t)}$
    Optimize θ w.r.t. clipped PPO surrogate $-\beta \sum_t d_t$
    Optionally anneal β to control exploration vs. stability
end for

RL from Human Feedback (RLHF)

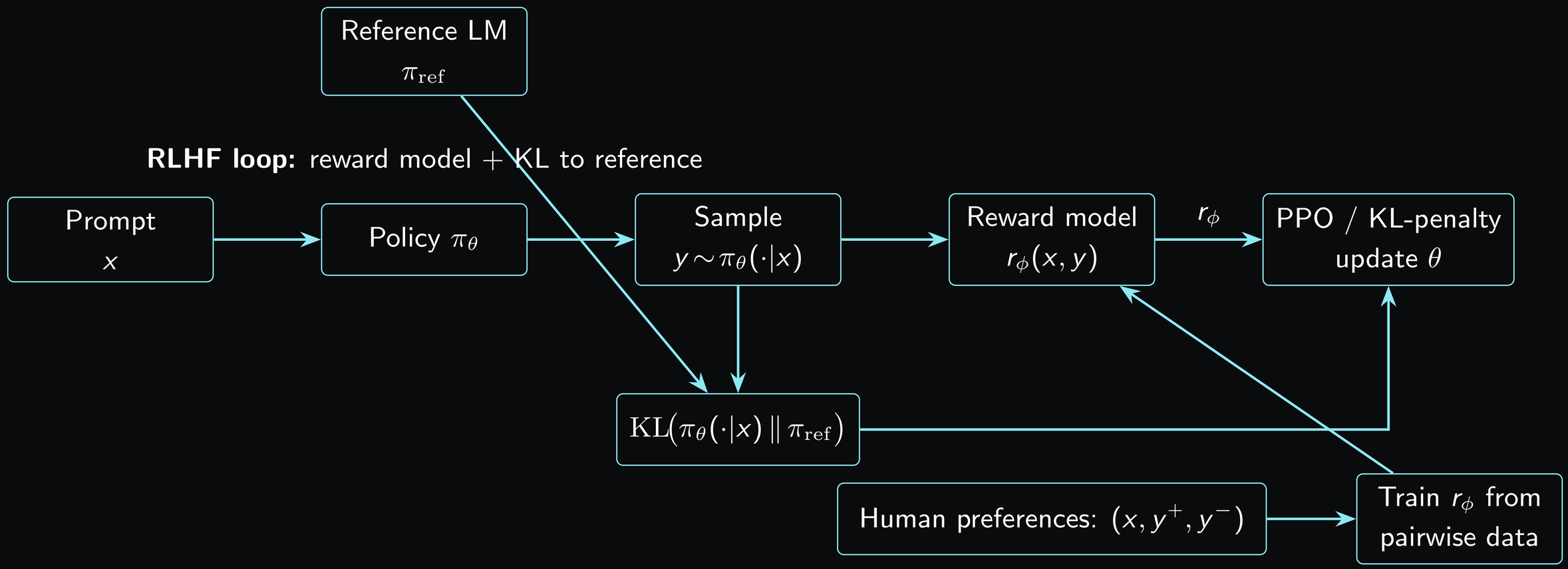


Figure 4: RLHF training loop: sample from π_θ, score with r_ϕ, penalize deviation from π_{ref} via KL, and update with PPO (while r_ϕ is trained from human pairwise preferences).

DPO: Direct Preference Optimization (no RL loop)

Focus. Replace RLHF with a simple, stable preference objective.

Setup. Given pairs $\{(x, y^+, y^-)\}$ where $y^+ \succ y^-$, DPO trains π_θ so that preferred responses are more likely *relative* to a reference π_{ref}.

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}\Big[\log \sigma\Big(\beta\big(\log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x) - \log \pi_{\text{ref}}(y^+ \mid x) + \log \pi_{\text{ref}}(y^- \mid x)\big)\Big)\Big]$$

Pros/Cons.

- *Pros:* No on-policy sampling; stable and simple; strong alignment quality.
- *Cons:* Less direct control of exploration; sometimes lags RL on hard reasoning unless augmented.

DPO: Direct Preference Optimization (no RL loop)

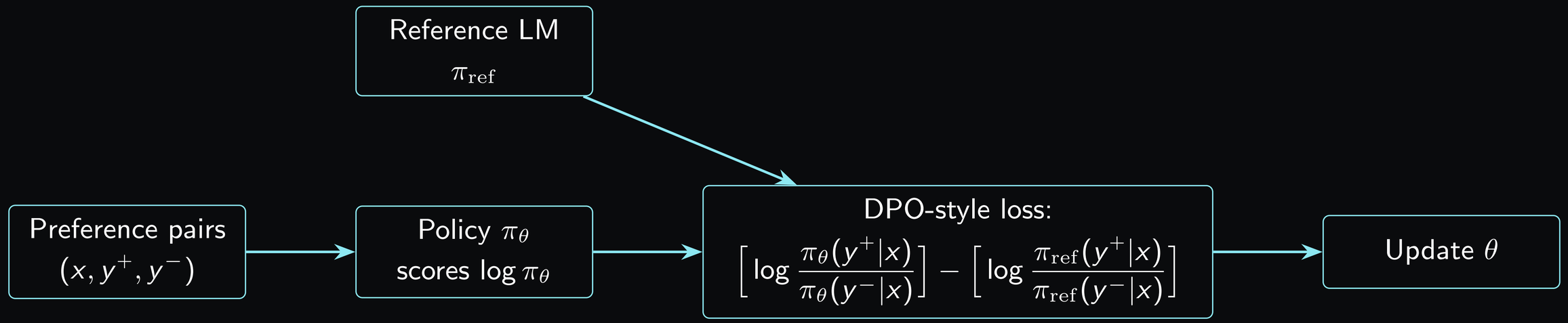


Figure 5: Direct preference optimization: optimize a closed-form preference objective that contrasts π_θ with π_{ref} on (y^+, y^-), avoiding an explicit on-policy RL loop.

KTO: Prospect-Theoretic Alignment with Binary Feedback

Idea. Treat alignment as *human-aware loss* (HALO): maximize a prospect-theoretic utility of generations, not preference log-likelihood.

Reference-adjusted reward. $r_\theta(x,y) = \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}, \quad z_0 = \mathrm{KL}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big)$

KTO value (logistic, risk/loss aversion).

$$v(x,y) = \begin{cases} \lambda_D \, \sigma\big(\beta \, [r_\theta(x,y) - z_0]\big), & y \sim y_{\text{desirable}} \mid x, \\ \lambda_U \, \sigma\big(\beta \, [z_0 - r_\theta(x,y)]\big), & y \sim y_{\text{undesirable}} \mid x, \end{cases}$$

with $\beta{>}0$ (risk aversion), λ_D, λ_U (loss aversion).

Objective (binary feedback only).

$$\boxed{\mathcal{L}_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{(x,y)}[\, \lambda_y - v(x,y) \,]}$$

Interpretation: increase utility for good outputs and decrease it for bad ones, while the reference term z_0 implicitly controls KL drift.

Kahneman-Tversky Intuition: Prospect Theory

Prospect Theory.

- *S-shaped*, asymmetric around the reference point (0).
- Concave for gains; convex for losses.
- **Steeper for losses** ($\lambda > 1$, loss aversion).

Link to KTO.

- KTO scores outputs via a prospect-style utility around a KL-based reference z_0.
- Small improvements on the "bad" side get *more* credit than equal regressions on the "good" side (tuned by $\beta, \lambda_D, \lambda_U$).

Canonical form.

$$v(x) = \begin{cases} x^{\alpha}, & x \geq 0, \\ -\lambda\,(-x)^{\beta}, & x < 0, \end{cases} \qquad \alpha, \beta \in (0,1),\ \lambda > 1.$$

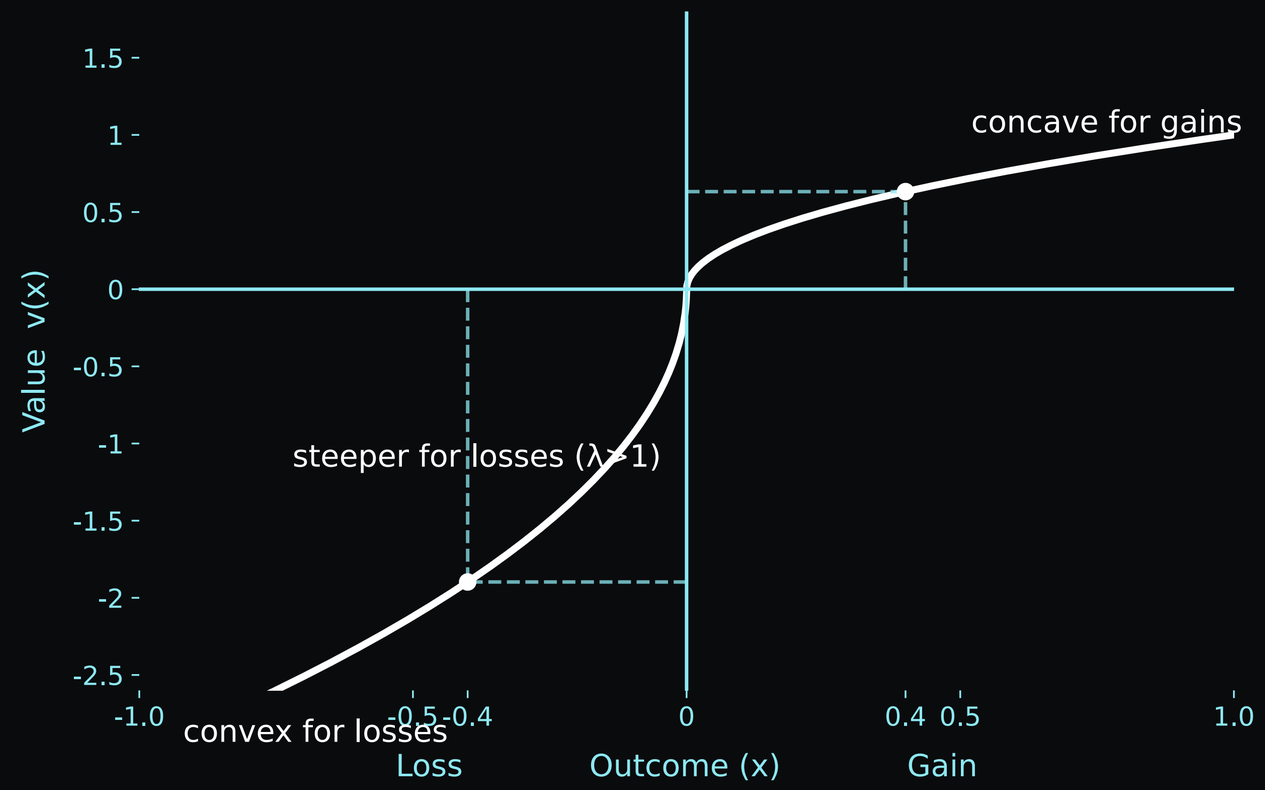


Figure 6: Prospect value (α=β=0.88, λ=2.25); dashed guides at $x = \pm 0.4$ show asymmetry.

GRPO: Group Relative Policy Optimization

Focus. Reasoning-centric RL with *relative* advantages over a group of candidates.

Idea. For a prompt x, sample a group $\{y^{(k)}\}_{k=1}^{K}$; compute normalized group returns $\tilde{A}^{(k)} = \frac{R^{(k)} - \mu_R}{\sigma_R}$, and perform a PPO-like update using *relative* advantages to emphasize *better-than-peers* trajectories.

$$\max_{\theta} \; \mathbb{E}\Big[\min\left(\rho^{(k)}(\theta)\tilde{A}^{(k)},\; \mathrm{clip}(\rho^{(k)}, 1\pm\epsilon)\tilde{A}^{(k)}\right)\Big] \quad - \; \beta\, \mathbb{E}[\mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})]$$

Why it helps. Stabilizes credit assignment for long CoT by comparing among sampled solutions; empirical gains on math/code reasoning.

RLVR: RL with Verifiable Rewards

Focus. Use *programmatic* verifiers (tests, checkers, execution traces) as the reward.

Definition. For prompt x and candidate y, define $r_v(x, y) \in \{0, 1\}$ by running a verifier (e.g., unit tests, theorem checker, calculator). Optimize π_θ with PPO/GRPO against r_v.

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta}\big[\, r_v(x, y)\,\big] - \beta\, \mathbb{E}_x \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}})$$

Strength. High-precision signals, strong gains in math, code, tool-use.
Caveat. Overfitting to tests or diversity collapse: mitigate with diversity-aware sampling, entropy bonuses, or divergence control.

Minimal Pseudocode: GRPO-Style Loop

Focus. Group Relative Policy Optimization (GRPO) stabilizes long-chain reasoning by comparing candidates *relative to each other* instead of against a single baseline. This reduces variance, highlights better-than-peers solutions, and works well with **verifiable rewards** (tests, compilers, math checkers).

Key idea. For each prompt x:

- Sample K candidate outputs from the policy.
- Score them with a verifier/reward function.
- Normalize scores to zero mean and unit variance (relative advantage).
- Apply PPO-style clipped updates, with a KL anchor to the reference model.

Algorithm 2 Group Relative PPO for Verifiable Rewards

for $x \in$ batch **do**
  Sample K candidates $y^{(1:K)} \sim \pi_\theta(\cdot \mid x)$
  Compute $R^{(k)} \leftarrow$ verifier/score$(x, y^{(k)})$
  Normalize $\tilde{A}^{(k)} \leftarrow (R^{(k)} - \mu_R)/\sigma_R$
  Update θ with PPO-clipped loss using $\tilde{A}^{(k)}$ and KL to π_{ref}
end for

Verifier-driven RL

Key Takeaway. Move beyond preference labels: use **automatic verifiers** (compilers, unit tests, theorem checkers, calculators) to produce reward signals that are *precise, scalable, and reproducible*. This grounds RL in programmatic correctness rather than subjective judgment.

Details.

- Generate K candidates per prompt with π_θ.
- Run each candidate through a bank of verifiers (tests, checkers).
- Aggregate binary or scalar pass/fail scores into a reward $\hat{r}$.
- Update π_θ with GRPO/RLVR/AGRO, anchored by KL to a reference LM.

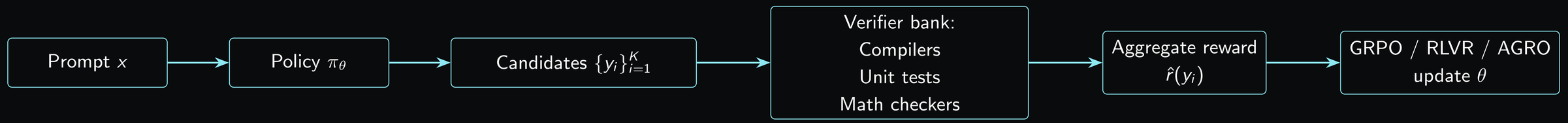


Figure 7: Verifier-driven RL: tool-based signals replace subjective preferences, providing sharper feedback for reasoning, math, and code.

SymbolicAI & VERTEX: A Verifiable Reward for Multi-Step Workflows

Context. *Computational graphs* of generative nodes; we need a semantic, reference-grounded score at each step.

VERTEX score (trajectory-level, bounded in $[0,1]$). Given an embedding engine $E : \Sigma \to \mathcal{H} \subset \mathbb{R}^d$ and kernel k, let μ_x^t, μ_y^t be the RKHS mean embeddings of generated vs. reference samples at time/node t. Using only MMD cross-terms,

$$\widetilde{\mathrm{MMD}}^2(\mu_x^t, \mu_y^t) \;\approx\; \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i^t, y_j^t),$$

and the VERTEX similarity integrates (discrete Monte Carlo in practice) over the graph/trajectory:

$$s(P_{\text{gen}}, P_{\text{ref}}) \;=\; \int_{t_0}^{t_f} \Big[\min\big(\max(0,\; \tfrac{1}{z}\, \widetilde{\mathrm{MMD}}^2(\mu_x^t, \mu_y^t) - z_{\mathrm{rand}}),\; 1 \big) \Big]\, dt.$$

Notes: z_{rand} subtracts a random-baseline similarity; z rescales vs. related solutions; min–max keeps $s \in [0,1]$. (Gaussian k used in paper; cosine also viable.)

How to use it as an RL reward (RLVR/GRPO/AGRO).

- Per node/time t: compute $s_t \in [0,1]$ against reference samples $\Rightarrow$ *verifiable* reward $r_v(x_t, y_t) = s_t$.
- Aggregate across nodes (sum/avg) for episode return; combine with KL control to a reference LM.
- Works for planning, tool-use, code, and multi-modal steps; naturally fits *closed-loop*, checkable tasks.

Symbolic Binding → Design by Contract

Idea. Bind *types* and *contracts* to the verifier.

Typing & contract.

$$\Gamma \vdash x : \tau_{\text{in}}, \qquad \{P\}\, C\, \{Q\} \text{ (probabilistic)}$$

$$\Gamma \vdash y : \tau_{\text{out}}, \quad Q(y, \mathcal{O}) \Rightarrow \text{Valid}(y)$$

Mini example

```
# Types (schema hints)
class Triplet(LLMDataModel):
  subj: str
  rel: Literal["grants","restricts","owns"]
  obj: str
```

```
# Contracted extractor (DbC layer)
@contract(
  pre  = [is_nonempty_text, ontology_loaded],
  post = [no_cycles, rel_in_ontology]
)
def extract(passage: str, Ont) -> Triplet: ...
```

Self-Rewarding LMs: LLM-as-a-Judge + Iterative DPO

Core idea. The model generates *and* scores its own data via LLM-as-a-Judge, then trains with Iterative DPO.

Loop (per iteration t).

- *Self-Instruction creation:* generate new prompts; sample N candidate responses; score them with the same model acting as a judge (additive 5-point rubric).
- *Preference set:* form (win, lose) pairs from highest/lowest judged responses; discard ties.
- *Training:* DPO on these pairs $\Rightarrow$ model M_{t+1}; repeat.

```
prompts -> [generate N candidates] -> judge(scores)
       -> select (y_win, y_lose) pairs -> DPO -> next model
```

Findings.

- Reward-modeling skill (LLM-as-a-Judge) *also* improves across iterations (e.g., higher pairwise agreement with human rankings).
- On AlpacaEval 2.0, a 70B model after 3 iterations outperforms several strong baselines (e.g., Claude 2, Gemini Pro, GPT-4 0613).
- Gains are broad but smaller on math/code/reasoning; length increases noted and discussed as a confound.

Self-Play Problem Generation: Conjecturing ⇒ Proving ⇒ Verifying

Idea. Don't just verify answers—*generate the problems*. Train two roles: a *conjecturer* (problem generator) and a *prover* (solution generator), with a formal *verifier* providing the reward.

Core loop (STP).

- Given a seed theorem with proof, the conjecturer proposes a related conjecture c.
- The prover attempts a proof; the verifier returns $r(c) \in \{0, 1\}$ (*pass/fail*).
- Select conjectures that are *barely provable* by the current prover and *elegant* (filters) to train the conjecturer.

Key selection & training objective.

$$\mathcal{C}_{\text{train}} = \Big\{ c \sim \pi_{\text{conj}}(\cdot \mid s) : 0 < \widehat{p}_{\text{solve}}(c; \pi_{\text{prov}}) \le \tau,\ \mathrm{Eleg}(c) \ge \eta \Big\}$$

$$\theta_{\text{prov}} \leftarrow \arg\max_{\theta}\ \mathbb{E}_{c,y\sim\pi_\theta}\big[\mathbf{1}\{\text{verify}(c, y) = \text{pass}\}\big]\ -\ \beta\,\mathbb{E}\big[\mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\big]$$

$$\theta_{\text{conj}} \leftarrow \arg\max_{\theta} \sum_{c\in\mathcal{C}_{\text{train}}} \log \pi_\theta(c \mid s) \quad \text{(optionally reweight by difficulty, e.g. } 1 - \widehat{p}_{\text{solve}})$$

Why it matters. Turns sparse RLVR into a *self-evolving curriculum*: progressive difficulty with dense rewards for proofs.

Refs: STP self-play conjecturing ↔ proving with Lean/Isabelle (2025).

AGRO Weights (I) — Regularized Return & Off-Policy Term

Regularized return (used throughout).

$$R^{\pi}_{\beta}(x,y) = r(x,y) \;-\; \beta \,\log \frac{\pi(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}$$

Off-policy gradient (samples $y \sim \mu(\cdot \mid x)$).

$$\nabla L_{\mu}(\pi) = -\,\beta\, \mathbb{E}\Big[\underbrace{\big(R^{\pi}_{\beta}(x,y) - R^{\pi}_{\beta}(x,\mu)\big)}_{w_{\mathsf{off}}(x,y)} \,\nabla \log \pi(y \mid x)\Big],$$

$$R^{\pi}_{\beta}(x,\mu) = \mathbb{E}_{y' \sim \mu}\big[R^{\pi}_{\beta}(x,y')\big]$$

Estimator with n samples per prompt (leave-one-out baseline).

$$\widehat{\nabla L_{\mu}}(\pi) = -\frac{\beta}{n} \sum_{i=1}^{n} \Big(R^{\pi}_{\beta}(x,y_i) - \underbrace{\tfrac{1}{n-1} \sum_{j \neq i} R^{\pi}_{\beta}(x,y_j)}_{\widehat{R}^{\pi}_{\beta,-i}(x,\mu)} \Big) \,\nabla \log \pi(y_i \mid x).$$

Intuition. w_{off} is a residual of R^{π}_{β} under the replay policy μ.

AGRO Weights (II) — On-Policy Term & 2-Sample Form

On-policy gradient (samples $y \sim \pi$).

$$\widehat{\nabla L}(\pi) = \frac{1}{2n} \sum_{i=1}^{n} \underbrace{\left(R_\beta^\pi(x, y_i) - \widehat{R}_{\beta,-i}^\pi(x, \pi) - \beta \right)^2}_{w_{\text{on}}(x, y_i)} \nabla \log \pi(y_i \mid x)\,.$$

Two-sample special case ($n = 2$; contrastive form).

$$\widehat{\nabla L_\mu}(\pi) = -\beta \Bigg(\underbrace{r_1 - r_2 - \beta \log \frac{\pi(y_1 \mid x)\, \pi_{\text{ref}}(y_2 \mid x)}{\pi(y_2 \mid x)\, \pi_{\text{ref}}(y_1 \mid x)}}_{w_{\text{2-samp}}(x; y_1, y_2)} \Bigg) \nabla \log \frac{\pi(y_1 \mid x)}{\pi(y_2 \mid x)}.$$

Takeaway. AGRO mixes replayed and fresh generations via consistency-derived weights:

$$w_{\text{off}} = \text{residual of } R_\beta^\pi, \qquad w_{\text{on}} = \text{squared residual shifted by } \beta.$$

RL Methods for LLMs — A Practitioner Table

Method	RM	Verifier	On-pol.	Typical use	Notes
RLHF (PPO)	✓	opt.	✓	Helpful alignment	Strong but compute-heavy; tune KL and length penalties.
DPO	×	×	×	Alignment via prefs	Simple, stable; weaker exploration.
KTO/ORPO	×	×	×	Pointwise prefs	ORPO drops reference model; needs careful calibration.
GRPO	✓/r_v	opt./✓	✓	Reasoning/coding	Group-relative advantages stabilize long CoT.
RLVR	×	✓	✓	Math/code/tools	Excellent with tests; watch diversity collapse.
AGRO	either	either	hybrid	Data efficiency	Unifies on/off-policy; reuses large logs effectively.

Chronology: Landmark Papers for RL in/for LLMs

- **RLHF foundations:** human feedback $\rightarrow$ reward models; PPO + KL regularization. (Christiano et al., 2017; Ziegler et al., 2019)
- **Summarization w/ feedback (pre-LLM explosion):** established the human-preference training loop. (Stiennon et al., 2020)
- **InstructGPT (2022):** large-scale RLHF with GPT-3. (Ouyang et al., 2022)
- **DPO (2023):** closed-form preference objective; no RL loop. (Rafailov et al., 2023)
- **KTO (2024):** prospect-theoretic preference optimization. (Ethayarajh et al., 2024)
- **GRPO (2024/25):** group-relative baselines for reasoning. (Shao et al., 2024)
- **RLVR (2024/25):** verifiable rewards via tests and checkers. (Wen et al., 2025)
- **AGRO (2025):** single algorithm for on+off-policy reward optimization. (Tang et al., 2025)

Beyond Preferences: Reward Shaping, Redistribution, Safety

Potential-based shaping. Add $F(s, a, s') = \gamma\Phi(s') - \Phi(s)$ to r preserves optimality:

$$r'(s, a, s') = r(s, a, s') + F(s, a, s'), \quad \pi^\star \text{ invariant.}$$

Reward redistribution (RUDDER). Decompose delayed returns to *redistribute* credit to key steps, reducing variance and bias in long-horizon tasks.

Safety via CMDP. Optimize reward under constraints $J_c(\pi) \leq c$ with Lagrangian

$$\max_{\pi} \min_{\lambda \geq 0} \; J_r(\pi) - \lambda \left(J_c(\pi) - c \right)$$

or trust-region constrained updates (CPO). Practical variants: PID-Lagrangian, proximal penalty.

Spurious Rewards: Rethinking Training Signals in RLVR

Takeaway. Verifiable rewards + a KL-anchored policy can reliably *elicit and stabilize* useful reasoning patterns; mix on- and off-policy data with consistency-derived weights, but *always validate across model families*.

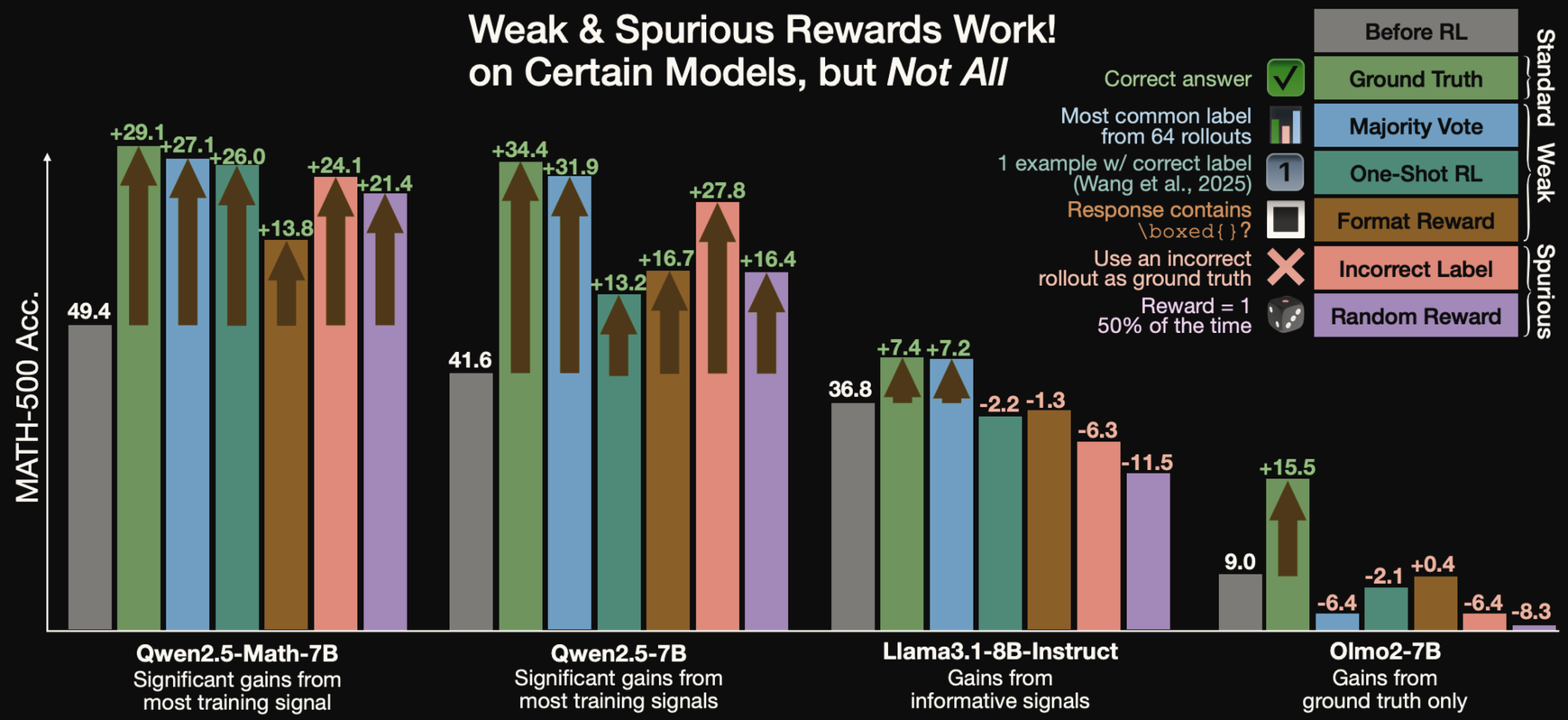


Figure 1: MATH-500 accuracy after 300 steps of RLVR on various training signals. We show that even "spurious rewards" (e.g., rewarding *incorrect* labels or with completely random rewards) can yield strong MATH-500 gains on Qwen models. Notably, these reward signals do not work for other models like Llama3.1-8B-Instruct and OLMo2-7B, which have different reasoning priors.

Goal-Conditioned & Reward-Free RL

UVFA. $Q(s, a, g)$ generalizes across goals g.

HER. Relabel failed trajectories with achieved goals $\hat{g}$ to turn sparse rewards into dense signals.

Relevance. Robotics and operations: define goals easier than dense rewards; dovetails with DPO-style preference targets and verifier-based goals.

Reasoning Models: Credit Assignment at Token-Level

Challenge. CoT has sparse, delayed credit; per-token objectives and group-relative baselines (GRPO) help.

Sketch. Token-level advantage from terminal verifier reward:

$$\hat{A}_t = \big(R - b(h_t)\big) \cdot w_t, \quad w_t \propto \text{attribution}(h_{\le t} \to R)$$

where w_t can be gradient-based or alignment-derived weights (redistribution).

Multi-Agent & Meta RL: Why You Should Care

MARL. Centralized training, decentralized execution (CTDE). Examples:

$$Q_{\text{tot}}(\tau, \boldsymbol{u}) = f\big(s, \{Q_a(\tau_a, u_a)\}_a\big), \quad \partial Q_{\text{tot}}/\partial Q_a \geq 0 \text{ (QMIX monotonicity).}$$

Meta-RL. Learn to *adapt fast*. RL^2 (RNN as fast learner), MAML (gradients for quick adaptation). Useful for non-stationary products and per-customer customization in industry.

Other Relevant Threads & What to Watch

- **Evaluation:** Reward/Judge Benchmarks (RewardBench, VerifyBench, JudgeBench); pitfalls (overoptimization, length bias, template bias).
- **Safety & Constraints:** CPO, PID-Lagrangian; Safe-RLHF variants.
- **Multi-modality:** VLA/robotics (e.g., OpenVLA); verifiable perception–action loops (simulation tests, hardware checks).
- **Offline & Diffusion RL:** policy learning from logs; diffusion policies for control.

Best Practices for Research in RL (LLMs & Beyond)

Reproducibility first. Report seeds, confidence intervals; standardize eval harnesses; avoid metric hacking.

Pick niches. Bridge gaps: e.g., verifier design, diversity preservation in RLVR, robust preference modeling.

Fail cases. Hunt failure modes (reward hacking, distribution shift, verifier leakage) and fix *those* first.

Checklists (suggested table).

- ☐ Ablations: KL strengths, sampling temperatures.
- ☐ Data & decontamination; prompt-template sensitivity.
- ☐ Robustness: corrupted/mislabeled preferences; OOD prompts.
- ☐ #tokens vs. quality trade-offs; compute budget disclosure.

Best Practices for Industry RL Integration

Before RL, ask: Is this actually an RL problem? Try heuristics, SL, bandits, control baselines.

When RL is right: clear objective signals and resettable, testable loops (sim, unit tests, sandboxes).

Operational tips.

- Start with *verifiable* sub-tasks; add preferences later.
- Separate *explore* vs *serve* policies; log everything.
- Automate eval: pass@k, tool success rates, safety constraints.
- Guardrails: KL to trusted reference, canary tests, rollback plans.

RL in Robotics (2024–2025) — Fast Tour

Themes. Zero/few-shot sim2real, mixed-reality digital twins, diffusion policies; goal-conditioned RL re-emerges.

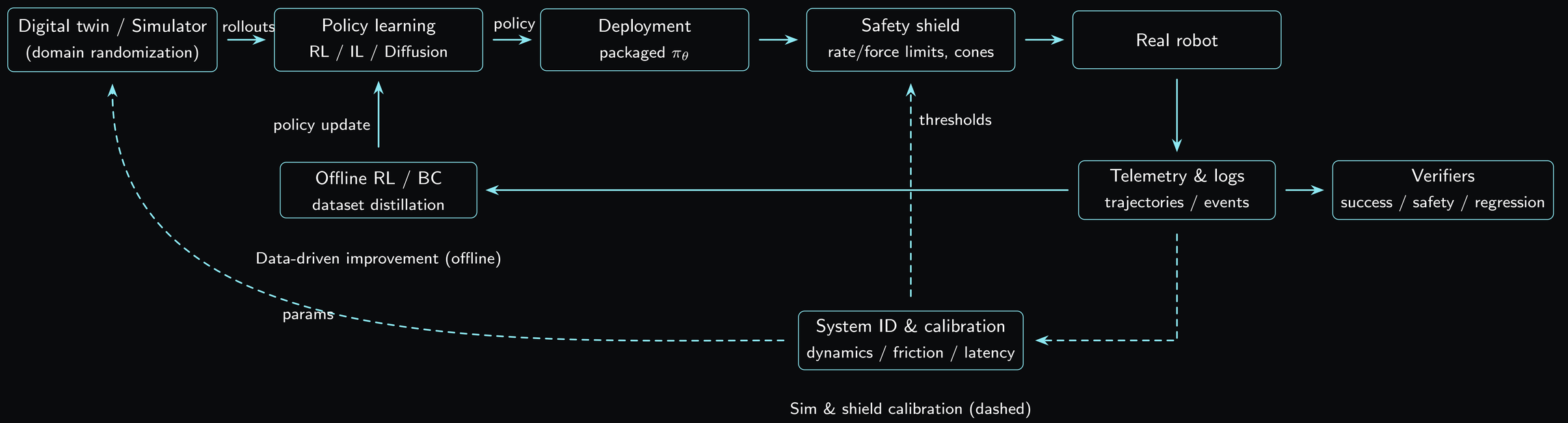
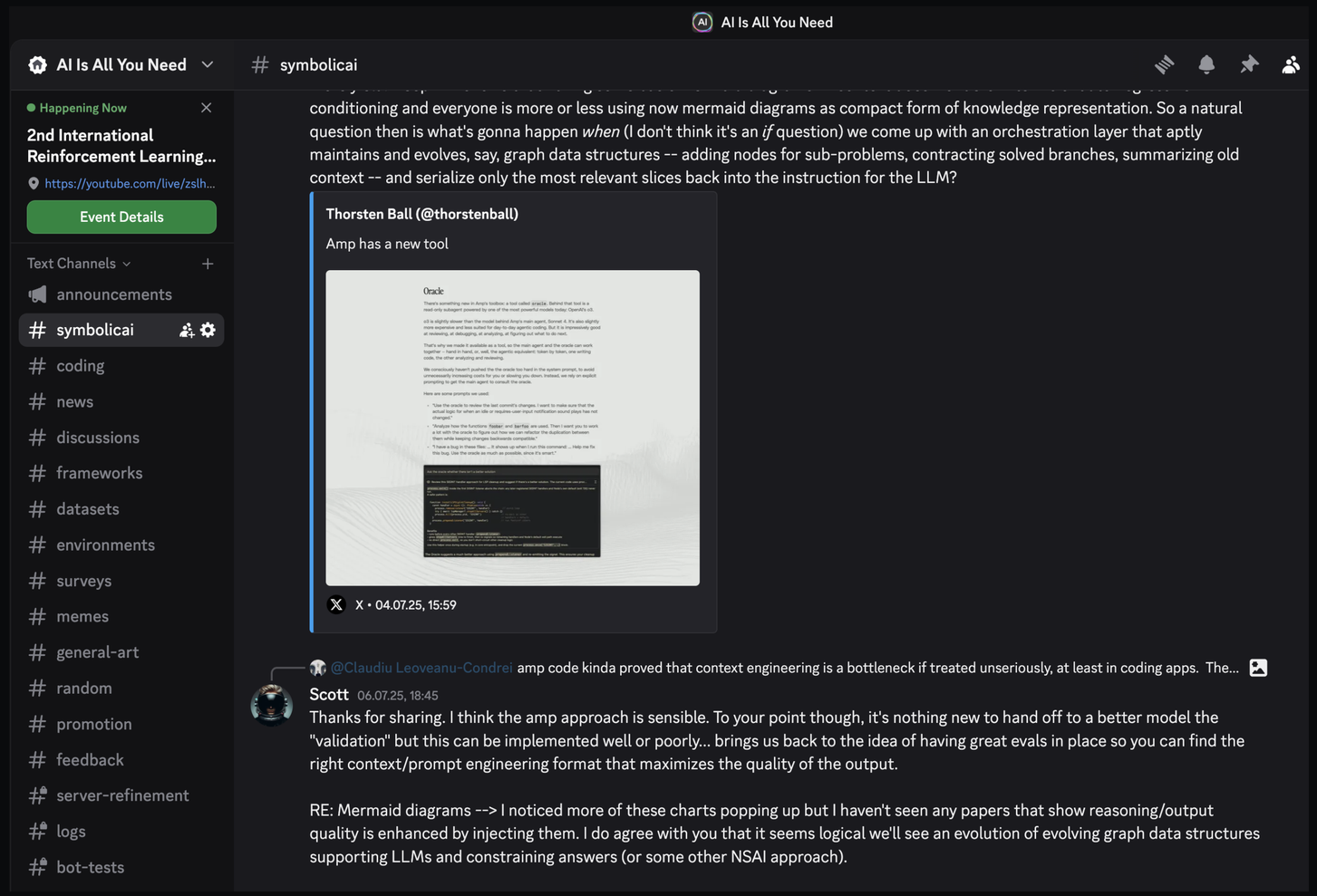


Figure 8: Sim2Real pipeline with calibration loop, safety shields, and offline logs. Verifiers act as task-success detectors, safety monitors, and regression tests; their signals feed offline training and shield calibration, closing the sim→real loop.

Summary & Takeaways

1. What is an agent and how it connects to neurosymbolic AI.
2. RL as post-training is *the* commercial success: RLHF $\rightarrow$ DPO/KTO/ORPO $\rightarrow$ GRPO/RLVR/AGRO.
3. Verifiable rewards are the jet fuel for reasoning/coding.
4. Safety & evaluation are first-class: KL control, constraints, robust benchmarks.
5. For industry: start verifiable, instrument everything, and keep a strong reference model.

Join the Discussion — Discord



discord.gg/azDQxCHeDA

Join the server for slides, code, and follow-ups

WALL·E: Agents meet Astrophysics

Why WALL·E?

- Playful example.
- Tools: lensing visualization, QNM "ringdown" synthesizer.

Segue. Up next: **WALL·E looks for black holes** — live demo:

- gravitational lensing fly-by,
- ringdown "growl" audio,

Ringdown Bound | Black Hole Cores

Ringdown Bounds on UV-Regularized Black-Hole Cores

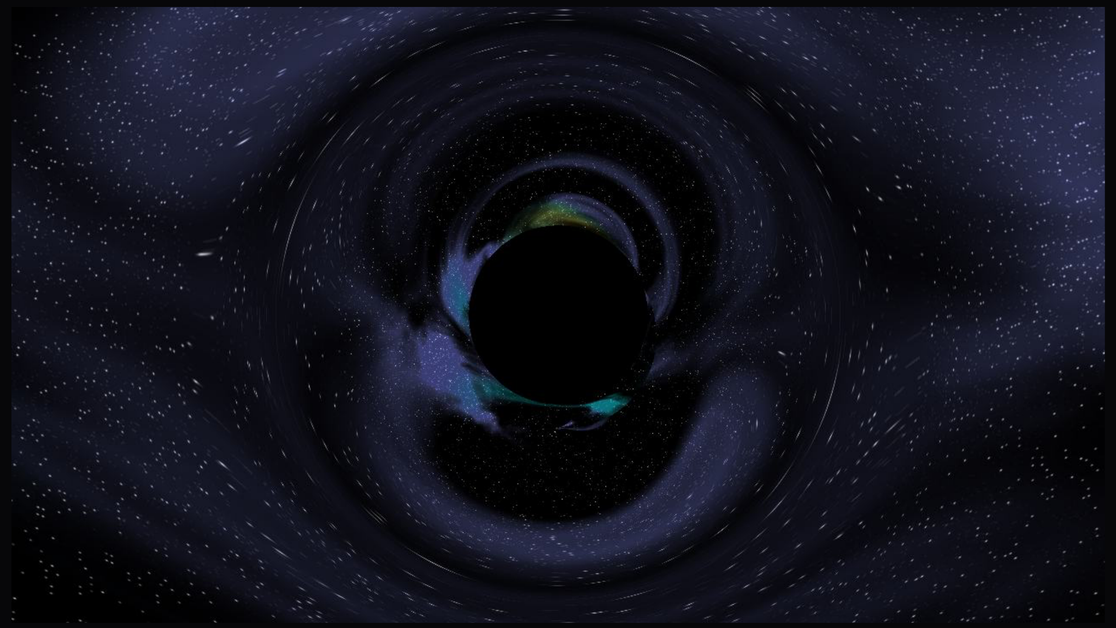
Marius-Constantin Dinu
ExtensityAI

Spacetime singularities in black-hole solutions signal a breakdown of the classical description at high curvature. We analyze a minimalist UV-regularized black-hole model with a single length scale L that preserves the exterior Schwarzschild/Kerr geometry and perturbs only the light-ring scattering barrier via a Hayward-type mass function. The resulting deformation induces small, correlated shifts in the dominant quasinormal mode (QNM). Similar behavior was reported in prior studies of regular metrics: Flachi and Lemos [1] found $\mathcal{O}(10\%)$ QNM deviations for a Minkowski-core spacetime, and Toshmatov *et al.* [2] showed that introducing a Hayward/Bardeen-like core increases the oscillation frequency and prolongs the damping time of test-field modes. We compute the Schwarzschild $(2, 2, 0)$ mode using (i) double-null time-domain evolution (anchor), (ii) an audited Leaver continued-fraction solver, and (iii) a locally calibrated WKB–Padé surrogate. We then perform a covariance-aware, multi-event hierarchical analysis with ringdown-start marginalization to test fractional and absolute scaling hypotheses. We obtain 95% credible bounds $\varepsilon \equiv L/r_s \leq 0.142$ and $L_0 \leq 47$ km. Cross-checks from EHT shadow diameters and S-star dynamics are consistent with these limits. Barrier diagnostics indicate that neglected interior-gradient terms scale as $\sim (L/r_s)^3$ and are subdominant across the posterior support. The present constraints are Schwarzschild-calibrated—in future work we outline a Teukolsky–CF Kerr deformation map that will supersede this calibration.

I. INTRODUCTION

GR has passed all precision tests to date, yet its singularity theorems imply geodesic incompleteness under generic conditions [3, 4]. The associated divergences and the information problem [5] motivate new UV-scale physics. Fundamental approaches (e.g. string theory [6], loop quantum gravity [7]) introduce rich structure, while *phenomenological* regular-black-hole models aim to capture the minimal ingredients needed to avoid singularities. Classic proposals include Bardeen's model [8] and Hayward's [9], which replace the singularity by a smooth core matched to an exterior Schwarzschild geometry (see Ref. [10] for a review of such models, and Ref. [11] for an interpretation of Bardeen's solution within general relativity).

Here we adopt a *one-scale* UV-saturation implementation – a cap on local frequencies/temperatures by Λ that tames the Tolman blueshift [12] and regularizes the interior with a core of size $L \sim \Lambda^{-1}$ (similar in spirit to a *limiting curvature* hypothesis [13, 14]). Crucially, the exterior is unchanged; strong-field tests probe



https://github.com/ExtensityAI/ringdown_bbhc

extensity AI

extensity AI

Thank you!

Marius-Constantin Dinu

marius@extensity.ai

www.extensity.ai

Publications

Dr. techn. Marius-Constantin Dinu
CEO & FOUNDER OF EXTENSITY.AI
PHD FROM JKU, LINZ

Publications

- **HyDRA: A Hybrid-Driven Reasoning Architecture for Verifiable Knowledge Graphs**
  A. Kaiser, C. Leoveanu-Condrei, R. Gold, M.-C. Dinu, M. Hofmarcher
  International Workshop on Advanced Neuro-Symbolic Applications (ANSyA), 2025

- **SymbolicAI: A framework for logic-based approaches combining generative models and solvers**
  M.-C. Dinu, C. Leoveanu-Condrei, M. Holzleitner, W. Zellinger, and S. Hochreiter Third Conference on Lifelong Learning Agents (CoLLAs), PMLR, 2024

  M.-C. Dinu, C. Leoveanu-Condrei, M. Holzleitner, W. Zellinger, and S. Hochreiter
  GenAI4DM Workshop at The Twelfth International Conference on Learning Representations (ICLR), 2024

- **Addressing parameter choice issues in unsupervised domain adaptation by aggregation**
  M.-C. Dinu, M. Holzleitner, M. Beck, H. D. Nguyen, A. Huber, H. Eghbal-zadeh, B. A. Moser, S. Pereverzyev, S. Hochreiter, and W. Zellinger, International Conference on Learning Representations (ICLR), 2023

- **The balancing principle for parameter choice in distance-regularized domain adaptation**
  W. Zellinger, N. Shepeleva, M.-C. Dinu, H. Eghbal-zadeh, H. D. Nguyen, B. Nessler, S. Pereverzyev, and B. A. Moser, Advances in Neural Information Processing Systems (NeurIPS), vol. 34, Curran Associates, Inc., 2021, pp. 20 798–20 811

- **Large Language Models Can Self-Improve At Web Agent Tasks**
  A. Patel, M. Hofmarcher, C. Leoveanu-Condrei, M.-C. Dinu, C. Callison-Burch, and S. Hochreiter
  Advances in Neural Information Processing Systems (NeurIPS), 2024 (under review)

- **XAI and strategy extraction via reward redistribution**
  M.-C. Dinu*, M. Hofmarcher*, V. P. Patil, M. Dorfer, P. M. Blies, J. Brandstetter, J. Arjona-Medina, and S. Hochreiter, In XXAI – Beyond Explainable Artificial Intelligence: State-of-the-Art and Future Challenges, ser. Lecture Notes in Artifcial Intelligence, vol. LNAI 13200, Springer International Publishing, May 2022

- **Align-rudder: Learning from few demonstrations by reward redistribution**
  V. Patil*, M. Hofmarcher*, M.-C. Dinu, M. Dorfer, P. M. Blies, J. Brandstetter, J. Arjona-Medina, and S. Hochreiter, Proceedings of the 39th International Conference on Machine Learning (ICML), vol. 162, PMLR, Jul. 2022, pp. 17 531–17 572

- **A Dataset Perspective on Offline Reinforcement Learning**
  K. Schweighofer*, A. Radler*, M.-C. Dinu*, M. Hofmarcher, V. Patil, A. Bitto-Nemling, H. Eghbal-Zadeh, and S. Hochreiter, First Conference on Lifelong Learning Agents (CoLLAs), Aug. 2022

- **Reactive exploration to cope with non-stationarity in lifelong reinforcement learning**
  C. A. Steinparz, T. Schmied, F. Paischer, M.-C. Dinu, V. P. Patil, A. Bitto-Nemling, H. Eghbal-zadeh, and S. Hochreiter, Conference on Lifelong Learning Agents (CoLLAs), 2022, 441-469

- **InfoDist: Online distillation with Informative rewards improves generalization in Curriculum Learning**
  R. Siripurapu, V. P. Patil, K. Schweighofer, M.-C. Dinu, T. Schmied, L. E. F. Diez, M. Holzleitner, H. EghbalZadeh, M. K. Kopp, and S. Hochreiter, Deep Reinforcement Learning Workshop NeurIPS, 2022

- **Understanding the effects of dataset characteristics on offline reinforcement learning**
  K. Schweighofer, M. Hofmarcher, M.-C. Dinu, P. Renz, A. Bitto-Nemling, V. P. Patil, and S. Hochreiter
  Deep Reinforcement Learning Workshop NeurIPS, 2021

- **A Two Time-Scale Update Rule Ensuring Convergence of Episodic Reinforcement Learning Algorithms at**
  the Example of RUDDER M. Holzleitner, J. A. Arjona-Medina, M.-C. Dinu, A. Vall, L. Gruber, and S. Hochreiter Optimization Foundations for Reinforcement Learning Workshop NeurIPS, 2019

APPENDIX

Appendix A: Equations Cheatsheet

Policy gradient. $\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a \mid s)\, Q^\pi(s,a)]$.

KL-regularized control. $\max_\pi \mathbb{E}_\pi[r] - \beta\, \mathrm{KL}(\pi \| \pi_{\mathrm{ref}})$.

DPO logistic loss. $\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E} \log \sigma\big(\beta[\Delta \log \pi_\theta - \Delta \log \pi_{\mathrm{ref}}]\big)$.

Potential shaping. $F(s,a,s') = \gamma\Phi(s') - \Phi(s)$ preserves $\pi^\star$.

CPO trust region (sketch). $\max_\pi J_r(\pi)$ s.t. $J_c(\pi) \le c,\ \ \mathrm{TV}(\pi, \pi_{\mathrm{old}}) \le \delta$.

References — RLHF & Preference Optimization (I)

- Ouyang *et al.*, "Training Language Models to Follow Instructions with Human Feedback" (InstructGPT), 2022. arXiv:2203.02155
- Stiennon *et al.*, "Learning to Summarize from Human Feedback," 2020. arXiv:2009.01325
- Christiano *et al.*, "Deep Reinforcement Learning from Human Preferences," 2017. arXiv:1706.03741
- Bai *et al.*, "Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback," 2022. arXiv:2204.05862
- Rafailov *et al.*, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2023. arXiv:2305.18290
- Hong *et al.*, "ORPO: Monolithic Preference Optimization without Reference Model," 2024. arXiv:2403.07691
- Ethayarajh *et al.*, "KTO: Aligning LMs with Prospect-Theoretic Preference Optimization," 2024. arXiv:2402.01306

References — Verifiable Rewards, Reasoning & Code RL (II)

- Shao *et al.*, "DeepSeekMath: Pushing the Limits of LLMs on Math with GRPO," 2024. arXiv:2402.03300
- Wen *et al.*, "Reinforcement Learning with Verifiable Rewards Implicitly Incentivizes Correct Chain-of-Thought," 2025. arXiv:2506.14245
- Li *et al.*, "VerifyBench: A Systematic Benchmark for Evaluating Reasoning Verifiers Across Domains," 2025. arXiv:2507.09884
- Tan *et al.*, "JudgeBench: A Benchmark for Evaluating LLM-based Judges," 2024. OpenReview
- Li *et al.*, "VL-RewardBench: A Challenging Benchmark for Vision-Language Reward Models," 2024. arXiv:2411.17451
- Dou *et al.*, "StepCoder: Improve Code Generation with RL from Compiler Feedback," 2024. arXiv:2402.01391
- Jain *et al.*, "μCode: Multi-Turn Code Generation Through Single-Step Rewards," 2025. arXiv:2502.20380
- Dai *et al.*, "Process Supervision-Guided Policy Optimization for Code Generation," 2024. arXiv:2410.17621

References — RL Algorithms for LLMs (III)

- Shao *et al.*, "DeepSeekMath" (GRPO), 2024. arXiv:2402.03300
- Tang *et al.*, "AGRO: RL-Finetuning LLMs from On- and Off-Policy Data with a Single Algorithm," 2025. arXiv:2503.19612
- Guo *et al.*, "G^{2}RPO-A: Guided Group Relative Policy Optimization for Verifiable Rewards," 2025. arXiv:2508.13023
- Yao *et al.*, "Diversity-Aware Policy Optimization for LLM Reasoning," 2025. arXiv:2505.23433
- Nath *et al.*, "Adaptive Guidance Accelerates RL of Reasoning Models," 2025. arXiv:2506.13923
- (Survey) Liu *et al.*, "Exploring Offline RL for Reasoning in LLMs," 2025. arXiv:2505.02142

References — Safe RL / CMDP (IV)

- Achiam *et al.*, "Constrained Policy Optimization (CPO)," 2017. arXiv:1705.10528
- Stooke, Achiam, Abbeel, "Responsive Safety in RL by PID Lagrangian Methods," 2020. arXiv:2007.03964
- Zhang *et al.*, "First-Order Constrained Optimization in Policy Space," 2020. NeurIPS'20
- Liu *et al.*, "Constrained Variational Policy Optimization for Safe RL," 2022. ICML'22
- Yao *et al.*, "Constraint-Conditioned Policy Optimization for Versatile Safe RL," 2023. NeurIPS'23

References — Reward Shaping & Redistribution (V)

- Ng, Harada, Russell, “Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping,” 1999. ICML’99
- Arjona-Medina *et al.*, “RUDDER: Return Decomposition for Delayed Rewards,” 2019. NeurIPS’19
- *Self–Rewarding Language Models: LLM–as–a–Judge & Iterative DPO.* arXiv:2401.10020v3, 2024/2025.
- R. Shao, S. S. Li, R. Xin, S. Geng, Y. Wang, *et al. Spurious Rewards: Rethinking Training Signals in RLVR.* arXiv:2506.10947v1, 2025.

References — Multi-Agent RL (VI)

- Lowe *et al.*, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments (MADDPG)," 2017. arXiv:1706.02275
- Rashid *et al.*, "QMIX: Monotonic Value Function Factorization for Deep MARL," 2018. arXiv:1803.11485
- Yu *et al.*, "The Surprising Effectiveness of PPO in Cooperative MARL (MAPPO)," 2021. arXiv:2103.01955

References — Meta-RL & Goal-Conditioned RL (VII)

- Duan *et al.*, "RL^2: Fast RL via Slow RL," 2016. arXiv:1611.02779
- Finn, Abbeel, Levine, "Model-Agnostic Meta-Learning (MAML)," 2017. arXiv:1703.03400
- Schaul *et al.*, "Universal Value Function Approximators (UVFA)," 2015. ICML'15
- Andrychowicz *et al.*, "Hindsight Experience Replay (HER)," 2017. arXiv:1707.01495

References — Robotics, VLA & Multimodal RL (VIII)

- Brohan *et al.*, "RT-1: Robotics Transformer for Real-World Control at Scale," 2022. arXiv:2212.06817
- Brohan *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," 2023. arXiv:2307.15818
- Kim *et al.*, "OpenVLA: An Open-Source Vision-Language-Action Model," 2024/25. arXiv:2406.09246 / PMLR'25

References — Practice, Reproducibility & Tooling (IX)

- Henderson *et al.*, “Deep Reinforcement Learning that Matters,” 2018. arXiv:1709.06560
- Raffin *et al.*, “Stable-Baselines3: Reliable Reinforcement Learning Implementations,” 2021. JMLR 22(268)

References — Problem Generation & Symbolic Scoring

- M.-C. Dinu, C. Leoveanu-Condrei, M. Holzleitner, W. Zellinger, S. Hochreiter. *SymbolicAI: A Framework for Logic-Based Approaches Combining Generative Models and Solvers* (CoLLAs 2024). VERTEX score for workflow/trajectory evaluation. arXiv:2402.00854
- K. Dong, T. Ma. *STP: Self-play LLM Theorem Provers with Iterative Conjecturing and Proving* (2025). Self-play problem generation (conjecturer) + prover with formal verifiers (Lean/Isabelle). arXiv:2502.00212
- C. Leoveanu–Condrei *A DbC Inspired Neurosymbolic Layer for Trustworthy Agent Design*. ECAI 2025 (submitted / preprint).
- B. Meyer. *Applying Design by Contract*. IEEE Computer 25(10):40–51, 1992; *Object–Oriented Software Construction* (2nd ed.). Prentice Hall, 1997.
- A. Kaiser, C. Leoveanu–Condrei, R. Gold, M.–C. Dinu, M. Hofmarcher. *HyDRA: A Hybrid-Driven Reasoning Architecture for Verifiable Knowledge Graphs*. arXiv:2507.15917v2, ECAI 2025 (submitted / preprint).

Reference Map by Category (X)

RLHF/Preference	Ouyang'22; Stiennon'20; Christiano'17; Bai'22; Rafailov'23; Ethayarajh'24; Hong'24
Verifiable/Code RL	Shao'24 (GRPO); Wen'25 (RLVR theory); Li'25 (Verify-Bench); Tan'24 (JudgeBench); Li'24 (VL-RewardBench); Dou'24 (StepCoder); Jain'25 (μCode); Dai'24 (PRM)
LLM RL Algorithms	Tang'25 (AGRO); Guo'25 (G^{2}RPO-A); Yao'25 (Diversity-Aware); Nath'25 (Adaptive Guidance)
Safe RL/CMDP	Achiam'17 (CPO); Stooke'20 (PID Lagrangian); Zhang'20 (FOCOPS); Liu'22 (CVPO); Yao'23 (CCPO)
Shaping/Redistrib.	Ng'99 (Potential-based shaping); Arjona'19 (RUDDER)
MARL	Lowe'17 (MADDPG); Rashid'18 (QMIX); Yu'21 (MAPPO)
Meta/Goal	Duan'16 (RL^2); Finn'17 (MAML); Schaul'15 (UVFA); Andrychowicz'17 (HER)
Robotics/VLA	Brohan'22 (RT-1); Brohan'23 (RT-2); Kim'24/25 (OpenVLA)
Practice	Henderson'18 (DRL that Matters); Raffin'21 (SB3)