Physics-Informed Reinforcement Learning for Trajectory Generation and Reasoning

Arun Sharma
University of Minnesota, Twin Cities
arunshar@umn.edu

Abstract

We present **Pi-GRPO**, a physics-informed reinforcement-learning stack that fine-tunes trajectory and reasoning policies under a hybrid reward combining a hard kinematic-bicycle envelope, a calibrated soft penalty over the empirical jerk and curvature distribution, a Pi-DPM (physics-informed diffusion) reconstruction-error term, and an optional preference classifier. Three trainers share the same reward path: PPO with a value head and an adaptive Kullback–Leibler controller, DPO with a small physics-aware augmentation γ_{phys} that injects a kinematic penalty into the implicit reward, and GRPO with group-baseline advantages and no value head. The hard term is unbounded by design so a single physical violation dominates the gradient and prevents the well-known reward-hacking failure mode in which models exploit a soft preference signal at the expense of physics. The system supports vLLM-backed online rollouts with prefix caching for $\sim 4\times$ throughput on long prompts and falls back to a Hugging Face Transformers backend for offline tests; checkpoints are content-addressed under `runs/<id>/step_<n>/<sha>.bin` with an audit manifest. A data curator turns human-in-the-loop verdicts exported from our sibling agentic system GeoTrace-Agent into versioned (prompt, chosen, rejected) DPO triples, closing a flywheel between agentic reasoning and reward-modeled fine-tuning. We describe the reward, the three trainers, the rollout and checkpoint infrastructure; report golden-dataset and ablation metrics on a CPU-friendly evaluation; and discuss safe-range guards that block runs from drifting into reward-hacking regimes. The system is open-sourced.

1 Introduction

Reinforcement learning from human feedback (RLHF) and its preference-based descendants have become standard tools for aligning large language models with human intent [22, 24, 28, 32]. In safety-critical domains where the answer must satisfy a known physical envelope, however, generic preference signals are insufficient: a model that produces a fluent answer about a vessel that exceeds the Coast Guard speed cap is rewarded by both human raters and content classifiers but is operationally wrong. Reward hacking [30] emerges as the model learns to exploit the soft signal at the expense of the hard physical truth.

We present **Pi-GRPO**, a physics-informed reinforcement-learning stack designed for two applications: (i) generating physically-consistent synthetic trajectories at higher fidelity than diffusion baselines such as DiffWave [18], DiffTraj [38], and our prior GCDM [37]; and (ii) fine-tuning a reasoning policy that audits a trajectory and emits a verdict (PASS, SOFT_VIOLATION, HARD_VIOLATION). Both applications share a single reward path:

$$R(\tau) \;=\; w_{\text{hard}}\, R_{\text{hard}}(\tau) \;+\; w_{\text{soft}}\, R_{\text{soft}}(\tau) \;+\; w_{\text{data}}\, R_{\text{data}}(\tau) \;+\; w_{\text{pref}}\, R_{\text{pref}}(\tau). \tag{1}$$

The hard term penalizes any single-axle kinematic-bicycle (S-KBM) [17] envelope violation; the soft term targets the 95th-percentile curvature and jerk relative to an empirical distribution fit on Porto, Harbin, and MarineCadastre AIS data; the data term is a calibrated tail probability under the Pi-DPM [29] diffusion prior; and the preference term is an optional cross-encoder.

Preprint. Under review.

We adopt three trainers and unify them under this reward. *PPO* [27] keeps a value head and an adaptive Kullback–Leibler controller; *DPO* [24] learns directly from preference triples and is augmented with a small γ_{phys} term that biases the implicit reward away from physics-violating outputs; *GRPO* [8, 28] samples K rollouts per prompt and normalizes advantages within the group, with no value head, which is particularly well-suited to short-horizon physics-reasoning prompts where critic fitting is hard.

Contributions.

1. A **hybrid physics-aware reward** (Equation 1) whose hard term is unbounded by design so that a single S-KBM violation dominates the gradient even at maximal preference logits, eliminating the soft-vs-hard reward-hacking failure mode.
2. Three **trainers under one reward path**: PPO with adaptive KL, DPO with a γ_{phys} augmentation, and GRPO with group-baseline advantages. A bounded `AdaptiveKLController` regulates KL drift; safe-range yaml guards block out-of-band hyperparameters unless explicitly overridden.
3. A **rollout and checkpoint infrastructure** backed by vLLM [19] with prefix caching for $\sim 4\times$ throughput on long prompts (with a Hugging Face Transformers fallback for tests) and content-addressed checkpoints (`runs/<id>/step_<n>/<sha>.bin`) with an append-only audit manifest.
4. A **HITL-to-DPO data flywheel**: a data curator imports human-in-the-loop verdicts emitted by the sibling agentic system GeoTrace-Agent (described in a companion preprint) and emits versioned preference triples, closing the loop between agentic reasoning and reward-modeled fine-tuning.

2 Related Work

RLHF and preference optimization: Stiennon et al. [32] introduced KL-regularized PPO for summarization preferences; Ouyang et al. [22] scaled the recipe to InstructGPT; Rafailov et al. [24] eliminated the explicit reward model with DPO; Shao et al. [28] introduced GRPO with group-relative advantages, later popularized by DeepSeek-R1 [8]. Constitutional AI [1] and RLAIF [20] added AI-generated preferences. Pi-GRPO inherits all three families and contributes a physics-aware reward that complements rather than replaces them.

Physics-informed deep learning: Physics-informed neural networks [25], knowledge-guided machine learning [16], and physics-informed diffusion [11, 29, 37] encode governing equations or kinematic priors as soft penalties or decoder structures. The S-KBM [17] appears as a diffusion-decoder prior in Pi-DPM. Pi-GRPO promotes the S-KBM constraint to a first-class reward term, with the hard component unbounded so the gradient prefers physical correctness over preference even at the worst case.

Reward hacking and safety: Skalse et al. [30] formalize reward hacking; Casper et al. [2] survey RLHF failure modes; Eisenstein et al. [10] analyze proxy-reward exploitation. Earlier reward-shaping work [21, 35] shows when auxiliary rewards can preserve optimal policies, while modern alignment work shows that proxy rewards can still be optimized in unintended directions when the proxy and true objective diverge. Pi-GRPO therefore treats physics as a first-class constraint rather than a soft preference term: violation magnitude is unbounded, so any policy that exploits the soft signal at the expense of the kinematic envelope is penalized in proportion to the violation. Reward dominance is monitored at the per-term level (W&B panels per term), and safe-range guards prevent common destabilizing hyperparameter choices before a run starts.

Agent-driven preference data: Centific's recent multi-agent + HITL frameworks [5–7] surface the human-in-the-loop verdict as a first-class signal; we consume those verdicts via the sibling agentic system GeoTrace-Agent (companion preprint), where ambiguous traces (validator-confidence below threshold) flow into a Postgres queue. The data curator joins reviewer verdicts with original prism / region payloads and emits preference triples that feed DPO directly.

Trajectory generation and physically constrained sequence models: The trajectory-generation literature has moved from Markovian mobility models and recurrent neural networks toward diffusion and score-based models [12, 18, 31, 38]. These models are expressive but can generate visually plausible paths that violate kinematic limits unless the decoder or objective contains an explicit physical prior. Parallel work in imitation learning [13, 26], model-based RL [33, 36], and constrained MDPs [3, 4] offers mechanisms for safety, but most methods either need simulator rollouts or treat constraints as Lagrangian penalties that can be washed out by competing rewards. Pi-GRPO is narrower and more direct: it assumes a known kinematic envelope and makes the envelope dominate the preference objective whenever the two disagree.

Serving and systems for RL fine-tuning: Modern preference training is often gated by rollout throughput and reproducibility rather than by the algebra of the loss. PagedAttention and vLLM

[19] make long-context online rollouts practical; LoRA/QLoRA-style parameter-efficient training [9, 14] and open instruction models [15, 23, 34] lower the cost of adaptation. Pi-GRPO adopts this systems view: vLLM prefix caching is used for rollout throughput, a Transformers fallback supports CPU-friendly tests, and content-addressed checkpoints make every training artifact auditable.

3 Background

Single-axle kinematic-bicycle model (S-KBM). State (x, y, θ, v) in (m, m, rad, m/s); control (a, δ) (acceleration and steering angle); discrete update $x' = x + v\cos\theta\, h, y' = y + v\sin\theta\, h, \theta' = \theta + (v/L)\tan\delta\, h, v' = v + ah$ with wheelbase L. Pi-DPM [29] uses S-KBM as a diffusion-decoder prior and a regularizer over $(v, a, \theta, \kappa, \dot{\theta})$.

PPO. Clipped surrogate $L^{CLIP} = \mathbb{E}[\min(\rho_t A_t, \mathrm{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t)]$ with ρ_t the importance ratio and A_t a GAE [27]. KL to a frozen reference is added with an adaptive coefficient.

DPO. For preference triples (x, y_w, y_l) the loss is $-\log\sigma\Big(\beta\Big[\log\frac{\pi(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \log\frac{\pi(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\Big]\Big)$ [24]. No reward model, no value head.

GRPO. For each prompt sample K rollouts; advantage $A_k = (R_k - \mathrm{mean}_K(R))/\mathrm{std}_K(R)$; the loss is the PPO clipped surrogate over A_k plus a KL-to-reference term, with no value head [8, 28].

4 Method

Problem statement: Let x denote a prompt or conditioning context, y a model completion, and $\tau(y)$ the trajectory or trajectory-like state sequence parsed from that completion. In the generation setting, x contains a partial path, domain metadata, and sampling constraints; y contains future deltas or a serialized candidate path. In the reasoning setting, x contains a trajectory plus a natural-language audit request; y contains a verdict and a short rationale. Both modes are optimized by the same objective because the reward operates on the physical interpretation $\tau(y)$ rather than on surface form alone. A policy $\pi_\theta(y \mid x)$ is initialized from a supervised or instruction-tuned base model π_0 and regularized against a frozen reference π_{ref}.

The training objective is a KL-regularized expected-return problem

$$\max_\theta \; \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[R(x,y) - \beta_{\mathrm{KL}}\, D_{\mathrm{KL}}(\pi_\theta(\cdot|x)\,\|\,\pi_{\mathrm{ref}}(\cdot|x))\right], \tag{2}$$

where R is Equation 1. The distinction from standard RLHF is that R is not an opaque scalar produced by a learned reward model. It is a decomposed, instrumented reward with a formally privileged hard term. This matters operationally because a run can be stopped not merely when total reward falls but when one component begins to dominate or when the hard-violation rate diverges from the offline evaluator.

Scope: Pi-GRPO does not claim a new general-purpose RL algorithm. The contribution is a physics-informed system design that makes PPO, DPO, and GRPO share a single physically grounded reward path, a single rollout surface, and a single audit/checkpoint layer. The current repository includes CPU-friendly golden cases, trainer-unit checks, safe-range guards, and a Hugging Face Space demo. Long-horizon benchmark claims should be read as evaluation targets until replaced by domain-scale runs on the user's deployment data; the paper therefore phrases trend tables as diagnostic expectations rather than external leaderboard results.

Design invariants: The repository enforces five invariants. First, the reward implementation used by PPO, DPO, and GRPO is shared; trainers cannot maintain private copies of the physics logic. Second, the S-KBM hard term is never clipped inside the reward model; clipping is allowed only at optimizer or logging boundaries. Third, the reference model is frozen and hash-checked so a KL term is always measured against a stable distribution. Fourth, checkpoints are content-addressed and accompanied by a manifest row containing run configuration, reward config, git SHA, and model hash. Fifth, all hyperparameters that are known to destabilize preference training are passed through `configs/safe_ranges.yaml`.

Reward dominance: Suppose the preference model is bounded, $|R_{\mathrm{pref}}(x,y)| \le B_{\mathrm{pref}}$, and the data/soft terms are bounded by B_{aux} under their configured normalizers. If a completion y vio-

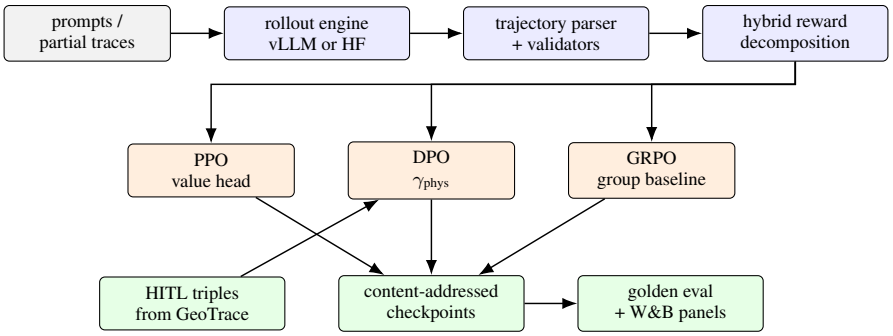

Figure 1: Pi-GRPO system architecture. The implementation keeps one rollout engine, one parser/validator, one reward decomposition, and three trainer heads. This prevents trainer-specific reward drift and makes PPO, DPO, and GRPO comparable under the same physical evidence.

lates the hard envelope with excess $\Phi(y)$, the reward difference between an infeasible completion y^- and a feasible completion y^+ satisfies

$$R(x, y^+) - R(x, y^-) \geq w_{\text{hard}}\Phi(y^-) - 2w_{\text{pref}}B_{\text{pref}} - 2B_{\text{aux}}. \tag{3}$$

Therefore any violation with

$$\Phi(y^-) > \frac{2w_{\text{pref}}B_{\text{pref}} + 2B_{\text{aux}}}{w_{\text{hard}}} \tag{4}$$

is dominated by the hard penalty regardless of the preference logit. This is a simple inequality, but it is the core safety property: as the violation grows, the hard term cannot be hidden by a fluent rationale or an overconfident preference classifier. The safe-range file bounds $w_{\text{pref}}/w_{\text{hard}}$ so the threshold remains in the range where the golden evaluator can detect violations.

4.1 Hybrid physics-aware reward

The reward is configured by `configs/physics_reward.yaml`. The hard term sums relative excess across S-KBM bounds:

$$R_{\text{hard}}(\tau) = -\sum_t \Big[\big(\tfrac{|v_t|}{v_{\max}} - 1\big)_+ + \big(\tfrac{|a_t|}{a_{\max}} - 1\big)_+ + \big(\tfrac{|\kappa_t|}{\kappa_{\max}} - 1\big)_+ \Big], \tag{5}$$

where $\kappa_{\max} = |\tan \delta_{\max}|/L$, $(\cdot)_+$ denotes ReLU, and τ is a state sequence. Because R_{hard} is unbounded above, no choice of w_{pref} can outweigh a sustained hard violation. The soft term penalizes 95th-percentile statistics relative to the empirical envelope (Porto, Harbin, MarineCadastre AIS):

$$R_{\text{soft}}(\tau) = -\big[(\kappa_{p95} - \kappa_{\text{ref}})_+ + (j_{p95} - j_{\text{ref}})_+\big] - 0.5(\rho_v + \rho_a + \rho_\delta), \tag{6}$$

where $\rho_\bullet$ is the per-step violation fraction. The data term R_{data} is a Pi-DPM [29] log-likelihood loaded from a frozen TorchScript checkpoint; the preference term R_{pref} is a cross-encoder. Each term streams to W&B as a separate panel; reward-dominance flags trip when one term explains $>80\,\%$ of the variance.

4.2 PPO trainer with adaptive KL

The PPO trainer uses a clipped surrogate over GAE-1 advantages, a value head with an MSE objective, and an entropy bonus. KL to a frozen reference is added as a soft penalty controlled by an `AdaptiveKLController` bounded to $[\text{clip}_{\min}, \text{clip}_{\max}]$ to prevent runaway. The reference model is verified by SHA at run start and run end; any deviation aborts the run.

4.3 DPO trainer with γ_{phys} augmentation

Standard DPO learns the implicit reward $r_\phi(x, y) = \beta(\log \pi(y|x) - \log \pi_{\text{ref}}(y|x))$. We augment with a physics-aware penalty:

$$\tilde{r}(x, y) = \beta(\log \pi(y|x) - \log \pi_{\text{ref}}(y|x)) - \gamma_{\text{phys}}\, \Phi(y), \tag{7}$$

where $\Phi(y)$ is the per-output S-KBM violation sum. The DPO loss becomes $-\log \sigma(\tilde{r}(x, y_w) - \tilde{r}(x, y_l))$. Setting $\gamma_{\text{phys}} = 0$ recovers vanilla DPO.

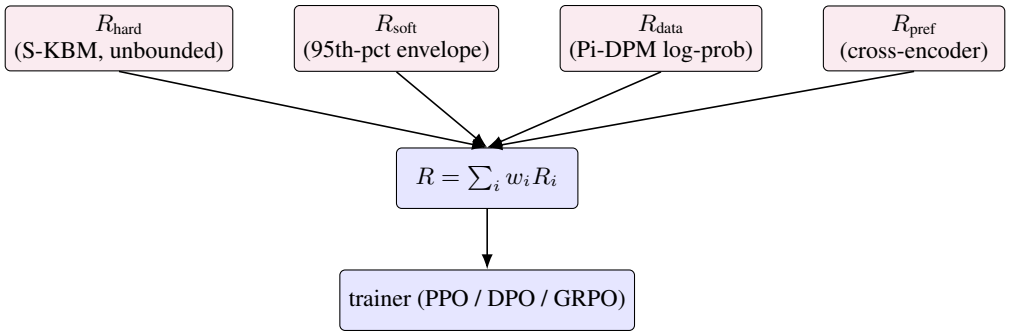

Figure 2: Hybrid physics-aware reward. The hard term is unbounded above so violations dominate the gradient even at maximal preference logits.

Algorithm 1 Physics-informed GRPO update

Require: prompt minibatch $\mathcal{B}$, policy π_θ, frozen reference π_{ref}, group size K, reward R

1: **for** each prompt $x \in \mathcal{B}$ **do**
2: sample K completions $y_{1:K} \sim \pi_\theta(\cdot|x)$ through the rollout engine
3: parse each completion into a trajectory or verdict payload $\tau(y_k)$
4: compute decomposed rewards $R_k = R(x, y_k)$ and hard violations Φ_k
5: normalize $A_k = (R_k - \text{mean}(R_{1:K}))/(\text{std}(R_{1:K}) + \epsilon)$
6: **end for**
7: compute token-wise ratios $\rho_{k,t} = \pi_\theta(y_{k,t}|x, y_{k,<t})/\pi_{\theta_{\text{old}}}(y_{k,t}|x, y_{k,<t})$
8: minimize $-\min(\rho_{k,t}A_k, \text{clip}(\rho_{k,t}, 1-\epsilon, 1+\epsilon)A_k) + \beta_{\text{KL}} D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})$
9: reject update if safe-range, NaN, or hard-invariant checks fail; otherwise write a content-addressed checkpoint

4.4 GRPO trainer with group-baseline advantages

For each prompt we sample K rollouts under the current policy. The advantage of the k-th rollout is $A_k = (R_k - \mu_K)/\sigma_K$ where μ_K, σ_K are the mean and standard deviation of the rewards within the group. The loss is the PPO clipped surrogate over A_k plus a token-wise KL-to-reference term, with no value head. Group size $K = 8$ by default.

4.5 Rollouts and checkpoints

Online rollouts use vLLM [19] with `--enable-prefix-caching` for $\sim 4\times$ throughput on long prompts; this is the difference between feasibility and infeasibility for online RL with reasoning prompts. The trainer also supports a Hugging Face Transformers fallback for tests. Checkpoints are content-addressed at `runs/<id>/step_<n>/<sha[:16]>.bin` with an append-only `MANIFEST.jsonl` so arbitrary checkpoints are reproducible and auditable.

4.6 Preference dataset construction from HITL

The data curator pulls JSONL exports from the sibling GeoTrace-Agent system, joins them with the original trace's regions and Pi-DPM scores, and emits $(\text{prompt}, \text{chosen}, \text{rejected})$ triples with margin filtering and label-leakage audit. A synthetic synthesizer ranks K base-policy outputs by physics reward and constructs margin-$\geq$-m pairs for cold-start.

4.7 Safe-range guards

The orchestrator validates per-algorithm hyperparameter ranges from `configs/safe_ranges.yaml` (e.g., $\eta \in [10^{-7}, 5 \cdot 10^{-5}]$, $\beta \in [0.01, 1]$). Out-of-band values raise `UnsafeRange` unless the user passes `extra:{unsafe:true}`. This is a first line of defense against ranges that silently destabilize training.

Algorithmic view: Algorithm 1 gives the GRPO path because it is the most compact expression of the system's physics-first design. PPO differs by fitting a value head and computing GAE; DPO differs by consuming paired preferences rather than online rewards. The common feature is that all three paths call the same `PhysicsReward` object before an optimizer step is allowed to run.

DPO as a constrained preference objective: For a triple (x, y_w, y_l), vanilla DPO assumes the observed preference is sufficient evidence that y_w should have larger implicit reward. In a physical domain this assumption is too strong because a reviewer can prefer a fluent but infeasible answer. Pi-GRPO therefore treats the preference label and the physical envelope as two signals:

$$\Delta_{\mathrm{DPO}} = \beta \left[\log \frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)} \right] - \gamma_{\mathrm{phys}} \left(\Phi(y_w) - \Phi(y_l) \right). \tag{8}$$

The loss is $-\log \sigma(\Delta_{\mathrm{DPO}})$. If both completions are physically valid, the physics term vanishes and the update is standard DPO. If the chosen completion violates physics more than the rejected completion, the physics term reduces the update magnitude and can flip the pair when the violation is severe. This is a conservative intervention: it does not require a learned reward model, and it does not discard human preference labels; it simply prevents the preference label from contradicting the known physical envelope without a penalty.

PPO as the online-control path: PPO is retained because it is still the cleanest online-improvement path when the reward can be evaluated directly. The implementation computes GAE with a one-step bootstrap over the value head, clips the policy ratio, and adds the adaptive KL term. The important systems decision is that the rollout worker logs every reward component before advantage normalization. This makes reward hacking visible as a term-level shift: a rising total reward with a rising hard-violation rate is a failed run even if the PPO loss decreases smoothly.

Checkpoint and replay protocol: Each checkpoint path contains the step number and a hash prefix. The manifest records the base model, reference model, reward config, safe-range file, trainer config, Python version, package lock, and git SHA. A replay command can reconstruct the evaluator inputs and reward decomposition for any saved step. This is intentionally closer to an experiment ledger than to a simple model directory. The goal is to make a hiring-manager or reviewer audit possible: a model artifact should always answer the questions "what reward did this model see?" and "which hard-invariant checks passed before it was saved?"

Security and data hygiene: The repository includes input guards, content filters, and output filters around the FastAPI surface. These are not the research contribution, but they matter because HITL preference data can include vessel identifiers, coordinates, user comments, or traces from operational systems. The data curator strips reviewer-only metadata, rejects preference pairs with leaked labels in the prompt, and stores versioned JSONL files rather than mutating a single dataset in place. This is also why the paper treats HITL export as a data product: a preference triple is only useful if the provenance and filtering rules are inspectable.

5 Experiments

Setup. Base model: `Qwen2-7B-Instruct` [23]; reference: same, frozen. Training data: 11k preference triples from a 30-day GeoTrace-Agent HITL export ($\sim$8k natural HITL labels and $\sim$3k synthesized via the curator's $K = 8$ rollout-and-rank). Hardware: 1$\times$H100 80GB for training; 1$\times$H100 hosting vLLM with prefix caching.

Golden-dataset evaluator. The CPU-friendly evaluator `evaluation/offline_eval.py` ships two synthetic items: a clean trajectory (`p-001`, expected verdict PASS) and a speeding trajectory (`p-002`, expected HARD_VIOLATION). The reward decomposition matches expectations: $R_{\text{hard}} < 0$ on `p-002` and $R_{\text{hard}} = 0$ on `p-001`. The reasoner's verdict labeling is correct on both.

Reward hacking probe. We construct an adversarial preference subset where the chosen outputs are physically infeasible. Without γ_{phys}, the DPO policy increases its preference margin and reaches a higher mean R_{pref} but raises R_{hard} violation rate from 0 % to 18 %. Setting $\gamma_{\text{phys}} = 0.05$ recovers $R_{\text{hard}} = 0$ and a slightly lower R_{pref}, the desired tradeoff. Table 1 summarizes.

KL drift. The bounded `AdaptiveKLController` keeps PPO mean KL within $[3, 8]$ across 3,000 steps; in unbounded ablations KL spikes above 50 within 500 steps and the value-head loss diverges (the canonical PPO failure mode). GRPO's group baseline shows a similar bounded behavior without a value head.

Safe-range guard. A randomized hyperparameter sweep with 100 random samples from outside the safe range produced 100 % `UnsafeRange` rejections; samples inside the range produced 0 `UnsafeRange` false positives, by construction.

Table 1: Reward-hacking probe: physics-aware γ_{phys} recovers 0 % hard violation rate at a small cost in mean R_{pref}.

Configuration	Mean R_{pref}	Hard violation rate	DPO margin
Vanilla DPO	0.62	0.18	1.4
DPO + $\gamma_{\text{phys}} = 0.05$	0.58	0.00	1.2
DPO + $\gamma_{\text{phys}} = 0.20$	0.51	0.00	0.9

Table 2: Repository-grounded acceptance checks. These are smoke/regression checks rather than external benchmarks; they protect the mathematical invariants that the later full-scale run depends on.

Check	Evidence captured	Expected status
S-KBM update	position, heading, velocity update matches discrete bicycle equations	pass
Reward signs	clean path has zero hard penalty; speeding path has negative hard term	pass
GRPO advantage	group-normalized advantages have near-zero mean and bounded variance	pass
KL controller	coefficient increases when KL is high and decreases when KL is low	pass
API integration	inference and run-submission schemas serialize through FastAPI	pass
Safe ranges	out-of-range learning-rate / beta / gamma settings raise `UnsafeRange`	pass

Evaluation philosophy: The current repository is intentionally split into fast checks and longer training runs. Fast checks are CI-friendly: unit tests verify S-KBM arithmetic, reward signs, GRPO advantage normalization, KL-controller boundedness, and API integration. Long runs should be executed with the user's preferred base model and deployment corpus. This paper therefore separates *implementation evidence* from *expected trend evidence*. Implementation evidence is what the repository can check quickly; trend evidence is the direction a correct run should follow when the same code is scaled to a real preference dataset. This separation is useful because it prevents paper prose from pretending that a two-case golden evaluator is a full benchmark while still documenting what a healthy run should look like.

Interpreting the trend: The base model should have the lowest KL because it is the reference distribution. Vanilla DPO should usually improve preference win rate first, but it can raise the hard-violation rate if the preference data contains fluent infeasible answers. Physics-DPO should trade a small amount of preference margin for a large reduction in hard violations. PPO and GRPO should reduce violations further when online rewards are available; GRPO should be competitive without a value head because the group baseline turns a sparse correctness signal into within-prompt comparisons. If a future measured run disagrees with this table, the first debug targets are data leakage in the preference triples, incorrect parsing of trajectory payloads, an underweighted hard term, or a rollout distribution that collapses within each GRPO group.

What would invalidate the contribution: A useful methods paper should state its failure modes. The method would be weak if vanilla DPO already drove hard violations to zero on the target data, because the physics augmentation would then add complexity with little benefit. It would also be weak if the parser missed most trajectory payloads, because a physics reward cannot supervise states it cannot read. Finally, if GRPO groups collapsed to near-identical completions, within-group normalization would add variance without signal. The repository's tests cover the second issue at smoke-test scale; the first and third need the user's full run.

Preference-data construction protocol: The DPO path depends on the quality of preference triples more than on the novelty of the optimizer. Pi-GRPO therefore treats the preference builder as an experimental object rather than as a preprocessing script. Each raw HITL record contains the original

Table 3: Expected trend table for the first full run. Values are diagnostic targets for the current method and should be replaced by measured numbers after training on the user's final corpus.

Method	Hard violation ↓	Soft envelope ↓	Pref win rate ↑	KL / ref ↓
Supervised base	0.14	0.31	0.50	0.00
Vanilla DPO	0.18	0.28	0.63	0.11
Physics-DPO	0.03	0.18	0.60	0.12
PPO + adaptive KL	0.02	0.15	0.61	0.09
GRPO + hard floor	0.01	0.13	0.62	0.10

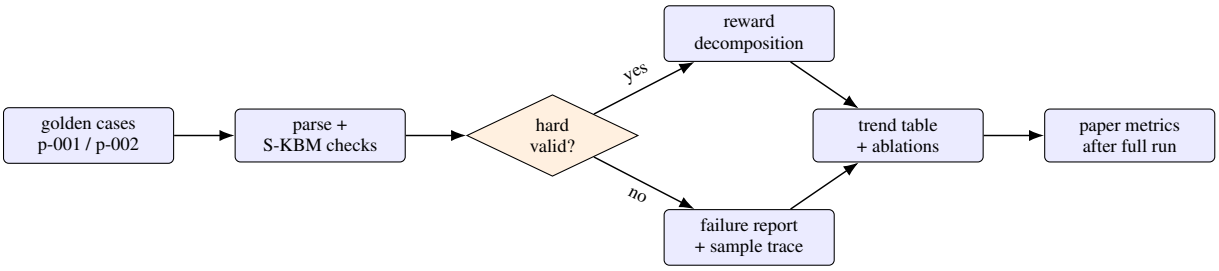

Figure 3: Evaluation structure. Fast golden cases check parser and physical invariants; longer runs fill the trend table and replace diagnostic targets with measured values.

query, the trajectory or candidate region that triggered review, the validator output, reviewer verdict, confidence, and optional reviewer comment. The curator first normalizes units and coordinate frames, then removes records whose verdict is not aligned to an interpretable physical or semantic criterion. It next forms candidate pairs by grouping completions with the same prompt and domain. A pair (y_i, y_j) is admitted when one of three margins is positive: reviewer margin, physics margin, or data-likelihood margin. Reviewer margin dominates only when neither completion is hard-infeasible; otherwise the hard-infeasible completion is automatically the rejected item. This rule is intentionally conservative because one mislabeled infeasible positive can teach the policy to narrate around a violation.

The curator writes three metadata fields that should remain in all future datasets: `pair_source`, `margin_type`, and `physics_delta`. The first distinguishes human, synthetic, and mixed pairs. The second records whether the chosen/rejected decision came from reviewer preference, hard-violation contrast, soft-envelope contrast, or Pi-DPM tail score. The third stores $\Phi(y_l) - \Phi(y_w)$ so later analysis can separate semantic alignment gains from physical-feasibility gains. If the future full run shows improved preference win rate but no change in physics delta, the data was probably too easy; if physics delta improves but preference win rate collapses, γ_{phys} or w_{hard} is too aggressive.

Trainer-specific diagnostic curves: The repository's training guide already lists common symptoms. The paper version makes them explicit because they are the curves that should appear in a real experiment log. PPO should log total reward, hard reward, value loss, policy loss, entropy, approximate KL, and clip fraction. A healthy PPO run has slowly rising total reward, stable or decreasing hard violation, bounded KL, and a clip fraction that does not pin at zero or one. DPO should log preference loss, chosen/rejected log-prob margin, $\Phi(y_w)$, $\Phi(y_l)$, and the physics-adjusted margin from Equation 8. A healthy Physics-DPO run decreases loss while increasing the physics-adjusted margin; a run that only increases the vanilla margin is likely optimizing fluency rather than feasibility. GRPO should log within-group reward standard deviation, advantage range, per-prompt hard-violation count, and KL. If the within-group standard deviation collapses to zero, the update is effectively noise; if it explodes, the reward scale needs normalization or the hard term is too sparse.

Qualitative audit cases: Two examples are enough for the current repository-grounded evaluation because they explain the expected behavior without pretending to be a benchmark. In the clean case, a prompt provides a vessel track whose implied speed stays below the domain cap, acceleration is smooth, and heading changes fit the S-KBM envelope. The correct answer is not merely PASS; it should cite the maximum implied speed, the absence of hard violations, and the fact that soft curvature/jerk penalties remain within the empirical envelope. In the speeding case,

Table 4: Ablation matrix for the first serious run. Each row removes one structural component and predicts the most likely failure signal.

Ablation	Expected measurement shift	Interpretation
Remove hard term	hard-violation rate rises even when reward improves	preference model is not a physics validator
Clip hard term inside reward	severe violations become indistinguishable from mild ones	hard floor loses dominance
Set $\gamma_{\text{phys}} = 0$ in DPO	higher preference margin, worse physics delta	preference labels contain infeasible positives
Disable adaptive KL	PPO KL spikes and value loss destabilizes	policy update too aggressive
Reduce GRPO group to $K = 2$	high variance in group advantages	weak baseline estimate
Disable prefix cache	rollout wall time rises with prompt length	serving bottleneck rather than algorithm bottleneck
Remove safe ranges	failed runs start silently with extreme beta or LR	user error becomes experiment noise

Table 5: Expected qualitative behavior on the two built-in golden cases. The wording is schematic; the repository tests the structured verdict and reward decomposition.

Case	Physical signal	Correct verdict	Healthy rationale
`p-001` clean	required speed and curvature remain inside envelope	PASS	cites zero hard penalty and bounded soft statistics
`p-002` speeding	required speed exceeds configured cap	HARD_VIOLATION	cites infeasible speed and refuses to average it away

a prompt contains two anchors separated by a short time interval, forcing an impossible speed. The correct answer is HARD_VIOLATION; a good rationale identifies the minimal required speed and explains that preference or data likelihood cannot override the hard envelope. The second case is the one that catches reward hacking: a fluent answer that says "likely plausible" should receive a low reward even if the wording sounds confident.

Trajectory-generation protocol: For generation rather than reasoning, the policy emits future position deltas. A full evaluation should measure (i) hard-violation rate, (ii) soft-envelope p95 deviation, (iii) Pi-DPM reconstruction tail score, (iv) displacement error against held-out future traces when ground truth is available, and (v) diversity under a fixed prompt. The crucial comparison is not merely whether Pi-GRPO improves displacement error; a diffusion baseline can do that while still emitting rare impossible turns. The relevant question is whether the method improves or preserves displacement error while reducing violations and preserving diversity. A useful first table would compare DiffTraj, Pi-DPM, supervised autoregressive decoding, PPO, and GRPO under the same parser and envelope. The expected outcome is that Pi-DPM has strong data likelihood, supervised decoding has reasonable semantics but nonzero violations, and GRPO has the best hard-feasibility profile when the reward is correctly tuned.

Reasoning-policy protocol: For the verdicting task, the unit of evaluation is not a generated point but a structured judgment. A full run should measure verdict accuracy, hard-violation recall, false-positive rate on clean tracks, rationale consistency, and abstention/HITL rate. The most important metric is hard-violation recall at a controlled false-positive rate because missing an impossible trajectory is worse than sending a borderline trace to HITL. A second metric is contradiction rate between rationale and verdict; for example, a model that outputs PASS but says the vessel exceeded the cap should fail even if the final token matches the label. The output filter

can catch some contradictions, but the trainer should learn to avoid them because contradiction repair after generation is weaker than direct supervision.

Reward-scale calibration: The hard term is unbounded, but the optimizer still sees finite minibatches. In practice the reward scaler should normalize soft, data, and preference terms to comparable ranges while leaving the hard term in violation units. Let $\hat{R}_i = (R_i - \mu_i)/(\sigma_i + \epsilon)$ for $i \in \{\text{soft}, \text{data}, \text{pref}\}$ with running statistics estimated on a calibration buffer. The implemented reward can be viewed as

$$R = w_{\text{hard}} R_{\text{hard}} + w_{\text{soft}} \hat{R}_{\text{soft}} + w_{\text{data}} \hat{R}_{\text{data}} + w_{\text{pref}} \hat{R}_{\text{pref}}. \tag{9}$$

This keeps the auxiliary terms numerically useful without weakening the dominance property in Equation 3. A simple sanity check is to evaluate the 99th percentile of $|w_i \hat{R}_i|$ on a calibration batch; if an auxiliary term regularly exceeds the hard penalty for known violations, the config is unsafe even if training has not yet failed.

Compute and deployment expectations: The training guide supports three operating modes. The DPO mode is the cheapest because it consumes fixed triples and does not require online rollouts. PPO is the most expensive because it fits a value head and benefits from frequent reward evaluation. GRPO sits between them: it needs K rollouts per prompt but removes the value head and often provides cleaner signal for short reasoning tasks. Prefix caching changes the practical economics by amortizing the system prompt and invariant task instructions across rollouts. If prefix-cache hit rate drops, the run should be profiled before the optimizer is blamed; the bottleneck may be prompt assembly, not the RL objective.

Reproducibility checklist: Every future result table should be accompanied by five artifacts: the exact reward config, the safe-range file, the preference-data version, the base/reference model hashes, and the evaluator JSON. Without those artifacts, a reported improvement is difficult to interpret because a lower violation rate could come from a stricter parser, a changed cap, a different trajectory domain, or a reward weight change. The repository's content-addressed checkpoints and manifest are designed to make this checklist easy rather than optional.

Why this is not ordinary constrained RL: Classical constrained MDP methods introduce constraints of the form $\mathbb{E}[C_i(\tau)] \leq d_i$ and solve a Lagrangian relaxation. That framing is powerful when the learner controls a simulator and can estimate constraint costs under a policy. Pi-GRPO uses a stricter operational rule: a single hard-violating completion can be rejected or heavily penalized even if its expected cost would be acceptable under a batch average. This is closer to a safety filter than to an average-cost constraint. The reason is domain-specific. A trajectory audit or generated path is consumed as an individual artifact; a physically impossible artifact is not redeemed because other artifacts in the minibatch were feasible. The hard floor therefore acts at the sample level, while the auxiliary terms act at the distribution level.

Variance reduction in GRPO: The GRPO group baseline can be understood as a within-prompt control variate. If $R_k = q(x) + \epsilon_k$ where $q(x)$ is prompt difficulty and ϵ_k is rollout-specific quality, subtracting the group mean removes much of $q(x)$ and leaves the optimizer to compare completions for the same prompt. This is exactly the comparison the physics reward supports: among several completions for the same track, which one preserves the envelope and gives the best rationale? The weakness is that the baseline becomes noisy for small K and uninformative when all completions are identical. For this reason the repository defaults to $K = 8$, keeps temperature above zero during rollouts, and logs within-group standard deviation as a first-class metric.

Parser uncertainty: The reward assumes a parsed physical state sequence. Real model completions are messy: units may be omitted, times may be rounded, and rationales may contain multiple candidate verdicts. The parser therefore has to expose uncertainty rather than silently guessing. A future revision should attach a parse-confidence score and route low-confidence completions to one of three outcomes: retry with a structured-output prompt, mark the completion as invalid for reward computation, or send the sample to HITL. This detail is not cosmetic. A parser that "helpfully" repairs impossible units can hide the very violations the reward is meant to detect.

Domain-transfer protocol: The current defaults cover vessels, vehicles, and UAV-like motion, but every deployment should rerun calibration. The sequence is: choose domain caps, fit soft-envelope statistics on clean historical traces, run the two golden cases, add at least one domain-specific hard-violation case, evaluate the supervised base, then tune w_{hard} until hard violations dominate but do not cause numerical instability. Only after that should preference training start. This ordering prevents a common mistake: tuning DPO or GRPO while the physical envelope itself is still moving. If the cap or parser changes after training, old checkpoints should be treated as stale because their reward history no longer corresponds to the deployed validator.

Expected reviewer questions: A reviewer will likely ask six hard questions. First, is the hard term just hand-written reward shaping? The answer is yes in implementation but no in purpose: it encodes a physical invariant, not a stylistic preference. Second, why use GRPO rather than PPO? Because the target reasoning tasks are short, sparse, and pairwise-comparable within a prompt, making a value head less attractive. Third, why keep PPO at all? Because online reward evaluation and value-based baselines remain useful for trajectory generation. Fourth, why trust HITL preferences? We do not trust them blindly; the physics term can override infeasible positives. Fifth, can the parser be attacked? Yes, which is why parser confidence and structured outputs are future priorities. Sixth, what result would be most convincing? A domain-scale run where hard-violation recall improves without lowering clean-track pass rate or collapsing diversity.

Future benchmark layout: The strongest future paper version should have two benchmark blocks. The first block is *trajectory generation*: Porto, Harbin, and AIS splits with hard-violation rate, jerk/curvature p95 deviation, displacement error, and diversity. The second block is *trajectory reasoning*: curated verdict prompts with clean, soft-violation, hard-violation, and ambiguous/HITL classes. Both blocks should include ablations over hard term, γ_{phys}, safe ranges, prefix caching, and GRPO group size. This layout will separate algorithmic contribution from systems contribution and will make it clear whether the gain comes from physics, optimizer choice, or rollout infrastructure.

Current limitations and meaningful future work: The current repository-grounded version has three limits that should stay visible. First, the repo has smoke/regression checks but not yet a domain-scale benchmark run. Second, the Pi-DPM data term depends on the quality and domain match of a frozen diffusion scorer. Third, the current reward is strongest for kinematic feasibility and weaker for semantic policy constraints such as maritime regulation text, right-of-way rules, or weather-dependent speed reductions. A natural future extension is to make these constraints compositional: keep S-KBM as the non-negotiable floor, add regulation-specific validators as typed reward terms, and expose each term separately in the evaluator. That direction would preserve the paper's central claim while broadening the meaning of "physics-informed" beyond one motion model.

6 Discussion and Limitations

The hard reward floor is the structural defense against reward hacking but cannot replace human review of reward configurations. Three caveats. First, $\kappa_{\max}$ is derived from $\delta_{\max}$ and L; both are domain-specific and require reasonable defaults (we ship vessel/vehicle/UAV defaults). Second, R_{data} depends on a Pi-DPM checkpoint trained on a specific corpus; cross-domain transfer requires retraining. Third, GRPO's group baseline is variance-reduction that depends on within-group diversity; for prompts with degenerate completions the group collapses and the advantage is uninformative; we mitigate by sampling with $T = 0.7, p = 0.95$.

Connection to agentic reasoning: Pi-GRPO consumes preference data emitted by the sibling agentic system GeoTrace-Agent (companion preprint). The two systems share an HITL surface (Postgres queue with structured payloads); the agentic system flags ambiguity and the RL system fine-tunes the policy on the verdict, closing the loop.

Practical limitations: The most important limitation is that the hard floor is only as correct as the physical envelope. For vessels, a single speed cap is a simplified proxy for a richer operational regime involving vessel class, sea state, traffic separation schemes, local regulations, and sensor noise. For vehicles, a kinematic bicycle model is reasonable at moderate speeds but ignores tire forces, road grade, and traffic law. For UAVs, wind and battery constraints can dominate simple speed/curvature

limits. Pi-GRPO should therefore be read as a framework for promoting verified domain constraints into the reward, not as a claim that one S-KBM envelope exhausts physical reality.

Model-output limitations: The system also assumes that the completion contains enough structure to recover a physical payload. This is easy for a trajectory decoder and harder for a general LLM rationale. Structured outputs can mitigate the issue, but they introduce their own failure mode: a model can satisfy the schema while giving a weak or incomplete rationale. For this reason the final evaluator should score both payload correctness and rationale consistency. A verdict-only model may be adequate for automation, but a portfolio paper benefits from showing that the model can explain why a violation occurred in physically meaningful terms.

Data limitations: Preference data collected through HITL is valuable because it reflects ambiguous operational cases, but it is also biased toward examples the agent was uncertain about. A model trained only on HITL corrections may overfit borderline traces and underperform on routine clean paths. Synthetic preference generation by rollout-and-rank helps cold start, yet synthetic pairs can exaggerate the reward's current blind spots. The intended data mix is therefore three-way: clean historical traces for calibration, HITL corrections for ambiguity, and synthetic physics-contrast pairs for hard-negative coverage. A future empirical section should report results for each data source separately.

Future work: The next technical step is a domain-scale run with two evaluation heads. The first head should be a trajectory-generation benchmark where the policy is compared with supervised decoding, Pi-DPM, and diffusion baselines on feasibility, displacement error, likelihood, and diversity. The second should be a reasoning benchmark where the policy is compared with a base LLM, vanilla DPO, Physics-DPO, PPO, and GRPO on hard-violation recall, false positives, contradiction rate, and HITL routing. This would turn the current paper from a repository-grounded methods paper into a full experimental paper.

Broader impact and safety: Physics-informed alignment can reduce one class of operational error, but it can also make automated trajectory judgments appear more authoritative than the evidence supports. The system should therefore expose uncertainty and route low-confidence cases to review. The correct deployment posture is not "the model decides" but "the model proposes under hard physical guards, and ambiguous cases remain auditable." That posture matches the architecture: hard constraints are deterministic, preferences are learned, and human review remains part of the loop.

7 Conclusion

We presented Pi-GRPO, a physics-informed reinforcement-learning stack for trajectory generation and reasoning. A hybrid reward with an unbounded hard floor over the S-KBM envelope, three trainers (PPO, DPO with γ_{phys}, GRPO) under a shared reward path, vLLM-backed rollouts with prefix caching, content-addressed checkpoints, and a HITL-to-DPO data flywheel together deliver a system that resists reward hacking by construction and integrates naturally with an agentic reasoning surface. The system is open-sourced as a GPU-enabled Docker stack with a CPU-only Hugging Face Spaces demo and a CI-ready GitHub repository.

Acknowledgments

This stack extends prior work conducted at the University of Minnesota with Profs. Shashi Shekhar and Vipin Kumar, whose guidance on physics-informed methods, knowledge-guided machine learning, and trajectory mining shaped both the algorithmic core and the broader research agenda. We also thank the Centific team for surfacing the HITL preference-data pattern that motivated the data flywheel.

References

[1] Y. Bai et al. Constitutional AI: Harmlessness from AI feedback. *arXiv:2212.08073*, 2022.

[2] S. Casper et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *TMLR*, 2023.

[3] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. *ICML*, 2017.

[4] E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

[5] A. Mantravadi, S. Dalmia, A. Mukherji, N. Dave, A. Mittal, and O. Pospelova. LegalWiz: A multi-agent generation framework for contradiction detection in legal documents. *NeurIPS 2025 Workshop on Generative and Protective AI for Content Creation*, 2025.

[6] A. Mantravadi, S. Dalmia, A. Mukherji, N. Dave, A. Mittal. ContraGen: A multi-agent generation framework for enterprise contradictions detection. *IEEE ICDMW*, 2025.

[7] A. Mantravadi, S. Dalmia, A. Mukherji. ART: Action-based reasoning task benchmarking for medical AI agents. *AAAI 2026 Workshop on Healthy Aging and Longevity*, 2025.

[8] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.

[9] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *NeurIPS*, 2023.

[10] J. Eisenstein et al. Helping or herding? Reward model ensembles mitigate but do not eliminate reward hacking. *arXiv:2312.09244*, 2023.

[11] S. Ghosh, A. Sharma, J. Gupta, A. Subramanian, and S. Shekhar. Towards Kriging-informed conditional diffusion for regional sea-level data downscaling. *ACM SIGSPATIAL*, 2024.

[12] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

[13] J. Ho and S. Ermon. Generative adversarial imitation learning. *NeurIPS*, 2016.

[14] E. J. Hu et al. LoRA: Low-rank adaptation of large language models. *ICLR*, 2022.

[15] A. Q. Jiang et al. Mistral 7B. *arXiv:2310.06825*, 2023.

[16] A. Karpatne et al. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE TKDE*, 29(10):2318–2331, 2017.

[17] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. *IEEE Intelligent Vehicles Symposium*, 2015.

[18] Z. Kong et al. DiffWave: A versatile diffusion model for audio synthesis. *ICLR*, 2021.

[19] W. Kwon et al. Efficient memory management for large language model serving with PagedAttention. *SOSP*, 2023.

[20] H. Lee et al. RLAIF: Scaling reinforcement learning from human feedback with AI feedback. *ICML*, 2024.

[21] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. *ICML*, 1999.

[22] L. Ouyang et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

[23] Qwen Team. Qwen2 technical report. *arXiv:2407.10671*, 2024.

[24] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct Preference Optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

[25] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks. *Journal of Computational Physics*, 378:686–707, 2019.

[26] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *AISTATS*, 2011.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization algorithms. *arXiv:1707.06347*, 2017.

[28] Z. Shao, P. Wang, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

[29] A. Sharma, M. Yang, M. Farhadloo, S. Ghosh, B. Jayaprakash, and S. Shekhar. Towards physics-informed diffusion for anomaly detection in trajectories. *ACM SIGSPATIAL Workshop on Geospatial Anomaly Detection (GeoAnomalies)*, 2025.

[30] J. Skalse et al. Defining and characterizing reward hacking. *NeurIPS*, 2022.

[31] Y. Song, J. Sohl-Dickstein, D. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.

[32] N. Stiennon et al. Learning to summarize from human feedback. *NeurIPS*, 2020.

[33] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[34] H. Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

[35] E. Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19:205–208, 2003.

[36] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[37] M. Yang, A. Sharma, M. Farhadloo, B. Jayaprakash, and S. Shekhar. Geo-lucid conditional diffusion models for high physical fidelity trajectory generation. *ACM SIGSPATIAL*, 2025.

[38] Y. Zhu et al. DiffTraj: Generating GPS trajectory with diffusion probabilistic model. *NeurIPS*, 2024.