Advanced Topic A08: *RLHF: Reinforcement Learning from Human Feedback*

Data Science with Python – BSc Advanced Lectures

Joerg Osterrieder

10 Minutes

Motivation: Aligning LLMs with Human Intent

Supervised fine-tuning alone produces capable but not helpful models

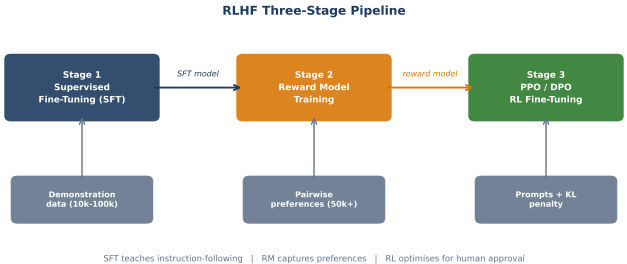
- A language model trained on next-token prediction learns to imitate text, not to help users
- Fine-tuning on curated demonstrations improves quality but does not capture nuanced preferences
- Human evaluators consistently prefer RLHF-trained models: more helpful, less harmful, more honest
- InstructGPT (2022): RLHF with 1.3B parameters outperforms a supervised 175B GPT-3 model on human preference ratings
- Finance: RLHF aligns financial advisory models to give regulation-compliant, risk-appropriate advice

RLHF is the technique behind ChatGPT, Claude, Gemini, and Llama-Instruct

The Three-Stage Pipeline

SFT then Reward Model then RL fine-tuning

- **Stage 1 – Supervised Fine-Tuning (SFT):** fine-tune the base LLM on a curated dataset of (prompt, high-quality response) pairs; produces the SFT model
- **Stage 2 – Reward Model (RM):** collect human preference comparisons (response A vs B per prompt); train a scalar reward model to predict human preference
- **Stage 3 – RL optimisation:** treat the SFT model as a policy; maximise the reward model's score using PPO; add a KL penalty to prevent collapse



The three-stage pipeline is used by OpenAI, Anthropic, Google, and Meta for their chat models

Stage 1: Supervised Fine-Tuning

Teaching the model to follow instructions

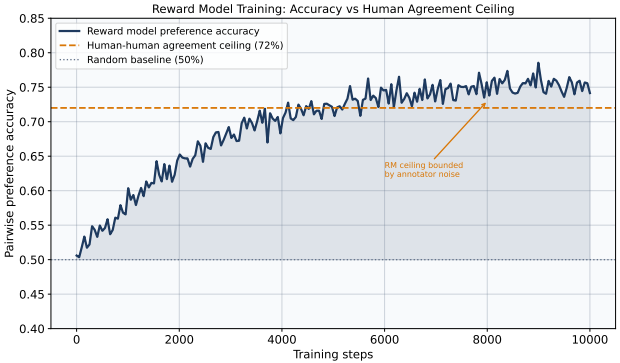
- Starting point: a pre-trained base LLM (e.g. Llama, Mistral, GPT)
- Training data: human-written demonstrations of ideal responses to a diverse set of prompts
- Standard cross-entropy loss: the model learns to reproduce human-quality outputs
- Typically requires 10k–100k high-quality (prompt, response) pairs
- Result: a model that can follow instructions but is not yet aligned to human preferences

SFT quality is critical: garbage demonstrations produce a garbage policy for RL optimisation

Stage 2: The Reward Model

A neural network that scores how much humans prefer a response

- Data collection: for each prompt, generate multiple responses; human annotators rank them
- Training objective: Bradley-Terry model for pairwise preferences $\mathcal{L} = -\mathbb{E}\left[\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))\right]$
- y_w: the preferred ("won") response; y_l: the dispreferred ("lost") response
- The RM outputs a scalar reward for any (prompt, response) pair
- Practical concern: reward hacking – the RL policy may exploit the RM's blind spots



Reward model quality is the bottleneck: human annotator consistency is typically 60–80%

Stage 3: PPO Fine-Tuning

Proximal Policy Optimisation to maximise human preference score

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi(\cdot|x)}\big[r_\theta(x,y)\big] - \beta\, D_{\mathrm{KL}}\big[\pi(\cdot|x)\,\|\,\pi_{\mathrm{SFT}}(\cdot|x)\big]$$

- π: the LLM policy being fine-tuned (parameters ϕ)
- r_θ: the frozen reward model
- $\beta\, D_{\mathrm{KL}}$: KL divergence penalty anchoring the policy near the SFT model
- PPO clip ratio controls update size, preventing catastrophic forgetting
- The entire LLM ($\sim$billions of parameters) is the policy: computationally expensive

PPO was chosen for RLHF because it is stable; newer methods (DPO, GRPO) avoid RL entirely

Reward Overoptimisation and the Alignment Tax

The policy diverges from truth as it over-exploits the reward model

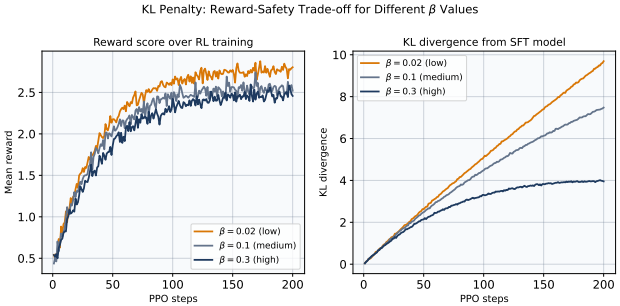
- As the policy moves away from the SFT model, the reward model (RM) becomes less reliable: the policy finds blind spots
- **Overoptimisation**: beyond an optimal KL budget, true human preference decreases even as RM score rises
- Gao et al. (2023): gold-standard preference drops past a KL threshold of roughly 0.1–0.3 nats from the SFT model
- Mitigations: ensemble reward models, iterated RLHF (collect new preferences after each round), RM uncertainty penalties
- Finance: a compliance RM can be gamed – the model learns to include regulatory keywords without giving genuinely safe advice

Overoptimisation is the core reliability risk: the RM is a proxy, not the true human preference

KL Penalty: Preventing Reward Hacking

The KL term anchors the model close to its pre-RL starting point

- Without the KL penalty: the policy quickly discovers degenerate responses that score high on the reward model but are incoherent or repetitive
- This is **reward hacking**: optimising the proxy (RM) rather than the true objective (human preference)
- The KL divergence measures how far the policy has drifted from the SFT model
- β controls the trade-off: high β stays safe but learns slowly; low β learns fast but risks collapse



Goodhart's Law in ML: when a measure becomes the target, it ceases to be a good measure

Direct Preference Optimisation (DPO)

Bypassing RL entirely with a supervised objective

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}\left[\log \sigma\left(\beta \log \frac{\pi_\phi(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\phi(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$$

- Rafailov et al. (2023): show the PPO + RM objective has a closed-form optimal policy
- DPO directly optimises the policy on (prompt, preferred, rejected) triples: no RL, no reward model
- Computationally far cheaper than PPO: only one LLM forward pass needed (vs 4 in PPO)
- Empirically competitive with PPO on most benchmarks; simpler to implement
- Most open-source fine-tuning today uses DPO or a variant (IPO, KTO, ORPO)

DPO is the dominant RLHF replacement in 2024–2025: same alignment, less infrastructure

The DPO Variant Family

Supervised preference optimisation without RL: an evolving ecosystem

- **IPO** (Azar et al. 2024): adds an ℓ_2 penalty on log-ratios to prevent overfit to the preference dataset; more stable than DPO on small datasets
- **KTO** (Ethayarajh et al. 2024): uses absolute feedback (good / bad labels) rather than pairwise comparisons; 2x more data-efficient when paired data is scarce
- **ORPO** (Hong et al. 2024): combines SFT and preference optimisation in one objective, eliminating the separate SFT stage
- **SimPO** (Meng et al. 2024): replaces the reference model log-ratio with sequence average log-probability; no reference model needed at inference
- Finance: KTO is attractive when only trade-level outcomes (profitable / unprofitable) are available as a feedback signal rather than pairwise expert comparisons

The DPO family eliminates RL overhead while matching PPO on standard alignment benchmarks

Collecting Human Preferences: Practical Challenges

The data pipeline is the most expensive part of RLHF

- Annotator disagreement: raters differ on ambiguous cases; inter-annotator agreement is the key quality metric
- Labeller bias: raters prefer longer responses, more confident tone, and familiar writing styles independent of correctness – this bias transfers to the RM
- Constitutional AI (Anthropic): replace human comparisons with a "constitution" of principles; an LLM self-evaluates using the constitution, creating synthetic preference data
- RLAIF (Google): use a stronger LLM as the preference annotator instead of humans
- Finance: annotators must be domain experts – financial accuracy matters more than style

Annotation cost: 1M human preference labels can cost $500k+; RLAIF reduces this by 100x

Limitations of RLHF

RLHF aligns surface behaviour, not underlying values

- Reward hacking remains a persistent risk even with KL penalty
- The reward model captures annotator preferences, including their biases and blind spots
- RLHF improves helpfulness measurably but does not solve hallucination
- Long-horizon tasks (multi-step financial planning) are poorly captured by single-response rewards
- Open problem: how do we specify human preferences for complex financial scenarios where optimal advice depends on unobservable client goals?

RLHF is alignment progress, not alignment solution; the field is actively evolving

Safety Fine-Tuning and Red-Teaming

Complementing RLHF with adversarial evaluation and self-critique

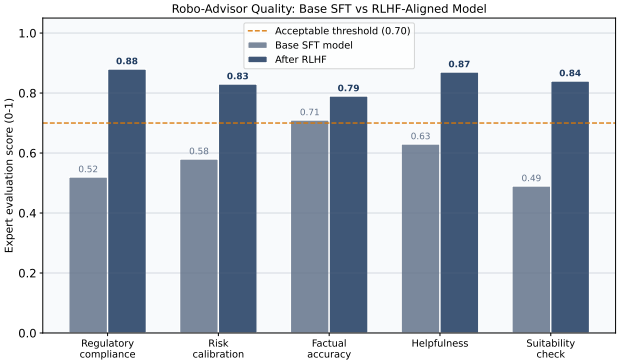
- RLHF optimises for helpfulness; safety requires adversarial testing on top of it
- **Red-teaming**: human or automated adversaries craft prompts designed to elicit harmful outputs; findings become new training data
- **Safety RLHF**: add a separate safety reward model trained on harmful vs harmless outputs; combine with the helpfulness RM as a weighted sum
- **Constitutional AI (CAI)** (Anthropic 2022): an LLM critiques and revises its own outputs against a written constitution; synthetic preference data replaces human annotation for harmful content (RLAIF)
- Finance: red-team for regulatory violations (market manipulation advice, suitability breaches); the safety RM penalises MiFID II or SEC suitability rule violations

Red-teaming + Constitutional AI + safety RM: the three-layer safety stack in production LLM deployments

Finance Application: Aligning a Robo-Advisor

Training a financial advisory LLM to give compliant, risk-appropriate advice

- Stage 1 SFT: fine-tune on curated Q&A from certified financial advisors
- Stage 2 RM: CFP annotators rank responses on: accuracy, risk calibration, regulatory compliance
- Stage 3 PPO/DPO: maximise the compliance-weighted reward score
- Key constraint: the RM must penalise advice that violates MiFID II or SEC suitability rules
- Result: a model that gives advice a regulator would approve rather than advice that sounds confident



RLHF for finance: replace "human preference" with "certified expert + regulatory constraint"

Code Pointer

RLHF and DPO fine-tuning in Python

- **TRL (Hugging Face)**: `pip install trl`; SFT, DPO, PPO, GRPO trainers built on top of Transformers; 5 lines of code for DPO fine-tuning
- **PEFT + LoRA**: fine-tune only low-rank adapters; 100x fewer trainable parameters; makes RLHF feasible on a single GPU
- **OpenRLHF**: multi-GPU PPO training at scale; used by open-source RLHF pipelines
- **Annotation tools**: Argilla (open-source) or Scale AI for collecting preference data
- Finance: add a domain-specific reward signal (compliance scorer) alongside the human RM

TRL + LoRA is the standard entry point: full RLHF pipeline on a consumer GPU in hours

Recent Advances: GRPO and Reasoning Models

Beyond RLHF: process reward models and chain-of-thought RL

- **GRPO** (DeepSeek-R1): Group Relative Policy Optimisation; no critic network needed; averages rewards within a group of sampled responses; computationally efficient
- **Process Reward Models (PRM)**: reward each reasoning step, not just the final answer; dramatically improves multi-step reasoning (maths, coding, financial calculation)
- **OpenAI o1 / DeepSeek-R1**: chain-of-thought tokens are the RL action space; models learn to reason more carefully by being rewarded for correct final answers
- Finance: PRMs can reward correct intermediate steps in a DCF valuation or option pricing chain

Reasoning models (o1, R1, Gemini 2.0) are trained with RL on chain-of-thought traces

Finance Application: Domain-Specific Reward Signals

Replacing generic human preference with expert and regulatory feedback

- Generic RLHF trains on crowd-sourced preferences; financial advice requires certified domain expert annotators
- **Composite reward**: combine factual accuracy (verified against Bloomberg data), regulatory compliance (automated MiFID II rule checker), risk calibration (Sharpe/drawdown realism), and communication clarity
- Automated compliance signal: a deterministic rule-based engine as a second RM; no annotation cost for black-letter regulatory rules
- Finance DPO pipeline: pair SFT responses against expert-rewritten alternatives; curate 50k pairs from CFPs; DPO fine-tune in under 12 hours on 4x A100s
- Key concern: the model may learn to sound compliant without being compliant; out-of-sample compliance audits by independent reviewers are essential

Domain RLHF with composite reward: automated rule-checking reduces annotation cost by 10x

Summary and Open Questions

Three things to remember

- RLHF: SFT then train a reward model on human pairwise preferences then PPO fine-tune with a KL penalty; DPO replaces the RL step with a simpler supervised objective
- KL penalty prevents reward hacking; the reward model is the alignment bottleneck – its quality determines the ceiling of the aligned model's behaviour
- Finance: RLHF aligns advisory LLMs to expert and regulatory standards; compliance-weighted reward signals replace pure human preference

Open questions:

1. Can process reward models fully replace outcome reward models for multi-step tasks?
2. How do we handle conflicting human preferences in a diverse user population?
3. Does RLHF improve calibration, or does it teach models to sound confident?

Further reading: Ouyang et al. (2022) InstructGPT; Rafailov et al. (2023) DPO; Shao et al. (2024) GRPO