# How Reinforcement Learning Is Used to Fine-Tune Large Language Models: A Comparison of PPO-based RLHF and DPO

**Ningyuan Xie**

Master of Computer Science, Siebel School of Computing and Data Science
University of Illinois at Urbana-Champaign
nxie3@illinois.edu

**Abstract**

Large language models (LLMs) are typically pretrained with next-token prediction, then adapted into useful assistants through a second stage of post-training. This survey asks why modern post-training expanded beyond supervised imitation to include preference-based optimization, and what tradeoffs the main approaches make. It reviews a widely used LLM pipeline for Reinforcement Learning from Human Feedback (RLHF): a combination of supervised fine-tuning (SFT) on demonstration responses, a preference-trained reward model, and a Proximal Policy Optimization (PPO) update to the generator. It then compares PPO-based RLHF with Direct Preference Optimization (DPO) and summarizes empirical claims about generalization and optimization behavior.

## Introduction

### Motivation and Problem Statement

This survey is about how reinforcement learning is used to fine-tune large language models into useful assistants. Specifically, it examines why modern post-training expanded beyond supervised imitation to include preference-based optimization, compares PPO-based RLHF and Direct Preference Optimization (DPO), and summarizes tradeoffs in supervision requirements, optimization complexity, and generalization.

For modern chat assistants, pretraining alone is not enough: simply scaling up does not guarantee better instruction following, and generations can still be wrong, harmful, or unhelpful (Ouyang et al. 2022). Bridging that gap is the job of *post-training*: further training on demonstrations, preferences, or other feedback from humans or AI (Ouyang et al. 2022; Bai et al. 2022).

Early post-training pipelines relied on supervised fine-tuning (SFT), which trains on written demonstration responses (Ouyang et al. 2022). In the settings studied by Chu et al. (2025), SFT helps stabilize response formatting before later RL, but it trains only on a fixed demonstration set (Ouyang et al. 2022). Classical imitation learning warns that cloning under a fixed offline distribution can suffer compounding errors when the learned policy visits uncovered states (Ross, Gordon, and Bagnell 2011).

Preference-based post-training emerged as one response. Some methods have humans or other models compare candidate responses, learn an explicit reward, and optimize it with RL while staying close to a reference model (Christiano et al. 2017; Bai et al. 2022; Ziegler et al. 2019; Ouyang et al. 2022). Methods such as DPO go a step further: they ask whether an explicit RL loop is necessary, or whether preference data can be turned directly into a closed-form training objective (Rafailov et al. 2023).

**Central question of this survey.** Why did LLM post-training expand beyond supervised fine-tuning to preference-based optimization, and what tradeoffs do the major approaches make, especially PPO-based RLHF and DPO, regarding supervision assumptions, optimization complexity, and generalization?

**Why did I personally choose this topic.** I have been curious about where the "learning" in modern chat assistants actually happens: pretraining alone does not make a model helpful, yet the post-training step is often glossed over. Choosing this topic was a way to work through that gap properly, rather than accepting "RLHF" as a label for something I did not fully understand.

**Why this matters to a broader audience.** Aligning language models matters because they are already used in many applications, and next-token pretraining alone does not make them follow user instructions helpfully and safely (Ouyang et al. 2022). InstructGPT closes that gap by fitting a reward model to ranked preferences, then improving the policy with PPO while a KL term anchors it to a reference model (Ouyang et al. 2022).

### Scope and Organization

This survey sticks to the InstructGPT-style pipeline and a small set of influential papers. The background covers the RL vocabulary needed to read the literature; later sections describe the RLHF pipeline and its evaluation, compare PPO-based RLHF with DPO, and summarize agreements, disagreements, and open gaps.

## Background: RL Concepts for Language Models

Reinforcement learning studies how an agent improves a policy through interaction or sampled experience to maximize expected cumulative reward (Sutton and Barto 2018).

The same vocabulary carries over to LLM post-training. The language model is treated as a stochastic policy $\pi_\theta(y \mid$

$x$) which maps a prompt $x$ to a response $y$ by sampling tokens one by one. The prompt sets the initial context; after each token, the state is the prompt plus tokens generated so far. A full response is a trajectory. Post-training changes $\pi_\theta$ so sampled responses score higher under some notion of quality.

Three distinctions recur throughout this survey:

1. **Supervision type** (what labels do we train on). SFT uses expert demonstrations $(x, y^\star)$; preference methods use triples $(x, y_w, y_l)$ in which annotators rank $y_w$ above $y_l$; methods that run an explicit RL loop further use a scalar reward $r(x, y)$, which may be learned (Christiano et al. 2017; Ouyang et al. 2022; Rafailov et al. 2023).
2. **Offline imitation versus learning from the policy's own samples** (fixed dataset versus the model's own responses). SFT trains only on a fixed demonstration dataset (Ouyang et al. 2022), whereas many RL-style methods update on responses from the current policy. As above, cloning under a fixed offline distribution can suffer compounding errors on uncovered states (Ross, Gordon, and Bagnell 2011). The papers below often label this contrast "offline" versus "on-policy."
3. **Reference-model / KL regularization** (how tightly we constrain drift from a reference model). Unconstrained reward maximization can push an LLM far from its pretrained distribution, so practical RLHF methods add a Kullback–Leibler (KL) penalty against a reference policy $\pi_{\text{ref}}$ (Ziegler et al. 2019; Ouyang et al. 2022).

## A Standard LLM RLHF Pipeline

RLHF combines supervised initialization, human preference data, and policy optimization into one post-training workflow. At LLM-assistant scale, the most cited version is InstructGPT's three-stage design (Ouyang et al. 2022), which this survey treats as the reference pipeline. Earlier work already used the same components at smaller scale for language (Ziegler et al. 2019) and summarization (Stiennon et al. 2020); Stiennon et al. (2020) describe an iterative loop of gathering preferences, fitting a reward model, and improving the generator, with SFT used as a warm start before that loop.

### Stage 1: SFT

Annotators write demonstration responses to prompts, and the pretrained model is updated on those examples with a supervised next-token loss (Ouyang et al. 2022). This stage establishes instruction-following behavior and response format, and often initializes later RL stages (Chu et al. 2025; Ouyang et al. 2022).

### Stage 2: Reward Modeling

This is where preference supervision comes in. For the same prompt, multiple responses are sampled and annotators compare or rank them (Ouyang et al. 2022). A reward model $r_\phi(x, y)$ is then trained to score responses consistently with these preferences, typically with a Bradley–Terry style ranking loss. Under that model, $P(y_w \succ y_l \mid x) = \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))$, and fitting reduces to the negative log-likelihood (Christiano et al. 2017; Stiennon et al. 2020).

$$\mathcal{L}_{\text{RM}}(r_\phi) = -\mathbb{E}_{(x,y_w,y_l)}\left[\log\big(\sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\big)\right]. \tag{1}$$

Table 1: How each RLHF stage addresses a specific limitation.

| Stage | Adds | Addresses |
|---|---|---|
| SFT | Demonstration imitation | Instruction behavior and response format |
| Reward model | Scalar preference score | Turn relative preferences into a trainable signal |
| PPO + KL | On-policy reward opt. + KL | Optimize learned preferences; control policy drift |

### Stage 3: Policy Optimization

Starting from the SFT model as the initial policy, training seeks higher expected reward subject to staying near a reference policy, giving the KL-regularized objective (Ziegler et al. 2019; Ouyang et al. 2022; Rafailov et al. 2023)

$$\max_\theta \ \mathbb{E}_{x\sim\mathcal{D}}\Big[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[r_\phi(x, y)\big] - \beta\,\mathrm{KL}\big(\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\big)\Big]. \tag{2}$$

PPO is commonly used for this update (Schulman et al. 2017; Ouyang et al. 2022). RLHF is the broader paradigm of learning from human feedback; PPO is a popular choice of optimizer within that pipeline. Without a constraint, the policy may exploit reward-model errors or collapse to degenerate high-reward outputs (Ziegler et al. 2019). Practical methods therefore add a KL penalty to limit movement away from the reference policy, which reduces room to exploit reward-model errors (Ouyang et al. 2022). The displayed objective is the sequence-level regularized target rather than the full PPO training surrogate.

### What Problem Does Each Stage Solve?

Table 1 summarizes how each stage addresses a specific limitation.

Empirically, human raters prefer InstructGPT outputs over much larger pretrained models that were not instruction-tuned with this procedure (Ouyang et al. 2022).

### How the Pipeline Is Evaluated

InstructGPT evaluates with human preference ratings rather than a single gold label: labelers rank multiple responses per prompt, and win rates against a baseline are measured. In pairwise human evaluations, the 175B InstructGPT output was preferred to 175B pretrained GPT-3 $85 \pm 3\%$ of the time; separately, even the 1.3B InstructGPT model beat the 175B GPT-3 baseline while using fewer than one percent as many parameters (Ouyang et al. 2022). Parallel evidence comes from summarization: Stiennon et al. (2020)

find that a learned reward model predicts human preferences better than ROUGE, and that optimizing the reward model yields summaries humans prefer over ROUGE-optimized ones. They also show that pushing reward-model optimization too far can lower agreement with humans even as the proxy score keeps rising (Stiennon et al. 2020).

## A Focused Comparison: PPO-based RLHF versus DPO

The three-stage pipeline is one way to turn preference data into a trained policy. For KL-regularized preference optimization with a fixed comparison dataset, DPO shows how to update the policy directly from those pairs, skipping a standalone reward model and the PPO loop (Rafailov et al. 2023).

### How DPO Works

DPO starts from the same KL-regularized reward-maximization objective as RLHF, but observes that the best policy under that objective can be written directly in terms of the reward. Substituting that relation into the Bradley–Terry preference model yields a loss that depends only on the policy $\pi_\theta$ and a frozen reference $\pi_{\text{ref}}$. Training then looks like pairwise classification on preference data: raise the relative likelihood of preferred responses over dispreferred ones. Under this reparameterization the policy itself encodes a reward-like ranking over responses, so separate reward fitting and PPO sampling are skipped. The resulting objective is (Rafailov et al. 2023)

$$\mathcal{L}_{\text{DPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)}\Big[\log \sigma\Big( \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\Big)\Big], \tag{3}$$

where $\sigma$ is the sigmoid and $\beta$ controls regularization toward the reference. In the underlying regularized objective, larger $\beta$ imposes a stronger penalty for departing from the reference. The loss is a logistic pairwise classification loss on log-likelihood ratios. The reference policy enters through those ratios, which come from DPO's KL-regularized derivation; no separate value or reward network is required.

### Tradeoffs

Table 2 compares the two families on the main practical axes.

Both methods treat preferences as coming from a latent reward and constrain drift from a reference model (Rafailov et al. 2023; Ouyang et al. 2022). They differ in whether that reward is a separate network and whether optimization uses an online RL loop over the model's own samples. Empirically, on Reddit TL;DR summarization with a GPT-J SFT initialization, DPO reached about a 61% win rate against reference summaries under GPT-4 evaluation, compared with about 57% for a PPO baseline at its best reported sampling temperature (Rafailov et al. 2023). Standard DPO learns only from the fixed preference set and does not sample new responses during training (Rafailov et al. 2023). PPO-based RLHF remains useful when training should adapt to fresh samples from the current policy, when a separately trained reward must score responses beyond the original comparison pairs, or when reward modeling and policy optimization need separate control.

Table 2: Conceptual tradeoffs between PPO-based RLHF and DPO.

| Axis | PPO-based RLHF | DPO |
|---|---|---|
| Supervision | Preferences → reward model → on-policy rollouts | Preferences used directly (standard DPO is offline) |
| Optimization | Explicit RL, often PPO | Offline preference loss |
| On-policy signal | Updates from current policy samples | Fixed preference pairs |
| Main appeal | Separate reward fitting and policy updates | Removes reward-model and PPO machinery |

## Agreements, Disagreements, and Gaps

### Where the Literature Largely Agrees

- **SFT alone often underperforms preference optimization on alignment evaluations.** In the instruction-following and summarization settings reviewed here, preference-trained models receive higher human preference ratings than the corresponding SFT baselines (Stiennon et al. 2020; Ouyang et al. 2022).
- **KL and reference constraints matter.** Unconstrained reward maximization is risky; staying near a reference policy is a shared design principle (Ziegler et al. 2019; Ouyang et al. 2022; Rafailov et al. 2023).
- **Preference data encode relative, not absolute, quality.** Preference labels provide relative judgments between candidate responses rather than absolute quality scores (Christiano et al. 2017).

### Where Claims Still Conflict or Remain Soft

- **Does RL mainly reshape existing capabilities, or create new ones?** Across two tasks, an arithmetic card game with textual rule variants (GeneralPoints) and a visual navigation environment (V-IRL), SFT reaches strong in-distribution accuracy but transfers poorly to novel rule variants and visual shifts, while RL maintains higher out-of-distribution accuracy. The same work finds that a prior SFT stage is still useful for locking in response structure, without which end-to-end RL fails to improve (Chu et al. 2025). A recent preprint disputes that interpretation: it argues that apparent SFT failures often come from narrow demonstration diversity and rigid prompting setups, and that more varied prompts plus chain-of-thought supervision lets SFT transfer competitively on related benchmarks (Lin et al. 2025). These experiments use outcome

rewards on specific reasoning and navigation settings, not a direct PPO-versus-DPO comparison for open-ended assistants.

- **Is an explicit RL loop necessary?** On the one hand, DPO shows that for many preference-learning settings a derived offline objective can replace PPO: preferences are turned into a classification loss on a fixed comparison set, with no sampling from the current policy during training (Rafailov et al. 2023). On the other hand, methods that keep updating from the model's own generations remain motivated when only demonstrations exist, or when continual learning is the main concern, as in Self-Distillation Fine-Tuning (SDFT) (Shenfeld et al. 2026). Group Relative Policy Optimization (GRPO) sits on that second side of the debate: it still runs an explicit sample–score–update RL loop, but drops the critic used in many actor–critic PPO setups and instead estimates relative advantage across several samples for the same prompt. In DeepSeekMath, that loop improves over the instruction-tuned baseline without a separately trained critic (Shao et al. 2024). The open question is therefore not whether PPO itself is mandatory, but when an online RL loop over fresh samples is worth keeping versus a pure offline preference objective.
- **Human versus AI feedback.** Classic RLHF asks people to compare answers. Constitutional AI instead uses a model to critique answers against a written list of principles, then trains on that AI feedback, so fewer preference labels have to come from direct human comparisons (Bai et al. 2022).

### On-Policy Learning Without Preference Data

The offline-versus-on-policy distinction also appears outside preference learning. SDFT asks whether a model can still benefit from its own samples when only demonstrations exist: the same network is prompted with an expert example to form a teacher distribution, then that signal is distilled into a student that sees only the query (Shenfeld et al. 2026). That result suggests the on-policy versus offline distinction runs deeper than the PPO-versus-DPO choice alone.

## Takeaways

Post-training expanded beyond supervised imitation because assistant quality is hard to capture with a single demonstration target. Preferences define "better," an explicit reward model or DPO's implicit reward representation turns that into an optimization signal, and KL-regularized updates keep the policy usable (Ouyang et al. 2022; Rafailov et al. 2023; Ziegler et al. 2019). PPO-based RLHF buys separate reward fitting, refreshed rollouts, and independently tuned policy updates at higher operational cost; DPO collapses that stack into an offline preference loss (Rafailov et al. 2023; Ouyang et al. 2022). In the instruction-following and summarization studies reviewed here, preference optimization outperforms the corresponding SFT baselines on human-preference evaluations (Stiennon et al. 2020; Ouyang et al. 2022), while disagreement remains about how much new capability RL adds and when an explicit RL loop is required (Chu et al. 2025; Lin et al. 2025; Rafailov et al. 2023).

## References


Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30.

Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q. V.; Levine, S.; and Ma, Y. 2025. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-Training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, 10818–10838. PMLR.

Lin, X.; Sang, H.; Wang, Z.; and Zhang, X. 2025. Debunk the Myth of SFT Generalization. arXiv preprint arXiv:2510.00237. Preprint; withdrawn ICLR 2026 submission.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*, volume 35, 27730–27744.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model Is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, 53728–53741.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 627–635. JMLR Workshop and Conference Proceedings.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, Y.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; et al. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Shenfeld, I.; Damani, M.; Hübotter, J.; and Agrawal, P. 2026. Self-Distillation Enables Continual Learning. *arXiv preprint arXiv:2601.19897*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to Summarize with Human Feedback. In *Advances in Neural Information Processing Systems*, volume 33, 3008–3021.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2 edition.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv:1909.08593*.