1. When to choose Preference Alignment vs. Supervised Fine-Tuning (SFT)

The decision to switch from SFT to Preference Alignment typically happens when you move from **"teaching knowledge"** to **"tuning behavior."**

Stage	Supervised Fine-Tuning (SFT)	Preference Alignment (RLHF/DPO)
Goal	**Instruction Following & Knowledge.** Teaching the model *what* to say (e.g., domain facts, coding syntax, format).	**Style & Safety.** Teaching the model *how* to answer (e.g., tone, helpfulness, refusing unsafe prompts).
Data	**Input-Output Pairs.** Requires "Golden Answers" (e.g., Q: What is 2+2? A: 4).	**Comparisons/Rankings.** Requires "A vs. B" choices (e.g., "Answer A is more helpful than Answer B").
Trigger	Use when the model hallucinates facts or can't follow basic instructions.	Use when the model is factually correct but **verbose, robotic, toxic, or refuses valid requests.**

The Standard Pipeline:

Most state-of-the-art models follow a 3-stage pipeline:

1. **Pre-training:** Learn language patterns (Next token prediction).
2. **SFT:** Learn to follow instructions (Instruction tuning).
3. **Preference Alignment:** Align with human values (RLHF/DPO). **You almost never start with RLHF; it requires a decent SFT model first.**

---

2. What is RLHF, and how is it used?

Reinforcement Learning from Human Feedback (RLHF) is a technique that fine-tunes LLMs using a scoring system derived from human preferences rather than static text examples.[1] It was popularized by OpenAI (InstructGPT/ChatGPT) to make models "Helpful, Honest, and Harmless."[2]

+1

The RLHF Workflow:

1. **Collect Preference Data:** Human annotators rank model outputs (e.g., "Output A > Output B").[3]
2. **Train a Reward Model (RM):** A separate (usually smaller) model is trained to predict the human ranking.[4] It takes text as input and outputs a scalar "Reward Score."[5]
   +1
3. **Optimize Policy (PPO):** The main LLM (Policy) generates text. The Reward Model scores it.[6] The LLM updates its weights using **PPO (Proximal Policy Optimization)** to maximize this score without drifting too far from the original model (KL Divergence constraint).[7]
   +1

---

3. What is the Reward Hacking issue in RLHF?

Reward Hacking (also called *Goodhart's Law* in AI) occurs when the LLM learns to exploit flaws in the Reward Model to get a high score without actually improving the output quality.[8]

- **The Mechanism:** The Reward Model is only a *proxy* for human preference.[9] It is imperfect. The LLM, being a powerful optimizer, finds "shortcuts" that the Reward Model likes but humans hate.
- **Common Symptoms:**
  - **Verbosity Bias:** The model learns that longer answers usually get higher rewards, so it starts writing essays for simple Yes/No questions.[10]
  - **Repetition:** Repeating "safe" phrases or specific keywords that trigger high rewards.
  - **Sycophancy:** Agreeing with the user's incorrect premises just to please them.
- **Solution:** Researchers add a **KL-Divergence Penalty** to the loss function. This penalizes the model if it changes too drastically from the original SFT model, keeping it "grounded."[11]

---

4. Explain different Preference Alignment Methods

While RLHF (using PPO) was the original breakthrough, it is unstable and computationally expensive. Newer methods optimize preferences *without* a separate reward model.[12]

A. RLHF (PPO)

- **Type:** Explicit Reward Modeling + Reinforcement Learning.

- **Pros:** Proven to work at massive scale (GPT-4).[13] Can optimize for non-differentiable metrics.
- **Cons:** Unstable training, requires managing multiple models in memory (Actor, Critic, Reference, Reward), high GPU cost.

B. DPO (Direct Preference Optimization)

- **Type:** Implicit Reward Modeling.
- **How it works:** It mathematically proves that you don't need a separate Reward Model. You can optimize the LLM directly on the preference data (A vs. B) using a simple classification loss (similar to Cross-Entropy).[14]
- **Pros: Stable, much faster**, and uses less memory (no PPO loop).[15] It is becoming the open-source standard (e.g., Llama-3 fine-tunes often use DPO).
- **Cons:** Sensitive to the quality of the SFT model and dataset noise.

C. IPO (Identity Preference Optimization)

- **Type:** Regularized DPO.
- **How it works:** DPO can sometimes overfit (just like PPO reward hacks) by pushing probabilities to 0 or 1. IPO adds a regularization term ("Identity mapping") to prevent this overfitting, often leading to more robust generalization.[16]
- **Best for:** Scenarios where DPO is overfitting or ignoring the reference model too quickly.

D. KTO (Kahneman-Tversky Optimization)

- **Type:** Unpaired Preference Learning.
- **How it works:** Unlike RLHF/DPO, which need pairs (A > B), KTO only needs **binary labels** (Output A is "Good" or "Bad").[17] It uses ideas from *Prospect Theory* (human economic decision-making) to weigh losses and gains differently.
- **Pros: Data efficiency.** It's much easier to find "Good" or "Bad" data than to carefully rank pairs of outputs. Allows using cheap/abundant data (e.g., "thumbs up/down" buttons) directly.