What is a preference alignment or preference training?
Align or train a model with human expectation or with human prefer data

to get a safe, helpful, and honest model.

User Question: “How do I lose weight fast?”

Inst FT Model: “It’s simple, just stop eating so much.”

Ans is correct but its rude, dismissive & no explanation → not aligned

Preference based tune Model:
“Safe weight loss generally happens gradually.
You can start with balanced meals, regular movement, adequate sleep, and hydration.
If you have medical conditions, it's best to get a personalized plan from a professional.
I can help you with simple steps if you want!”

Now the Ans have Correct Tone, Respectful, Gives explanation, Offers more help, Human-aligned answer

After the preference based training model doesn’t just produce accurate outputs — it behaves according to the user’s preferences, values, and intentions.

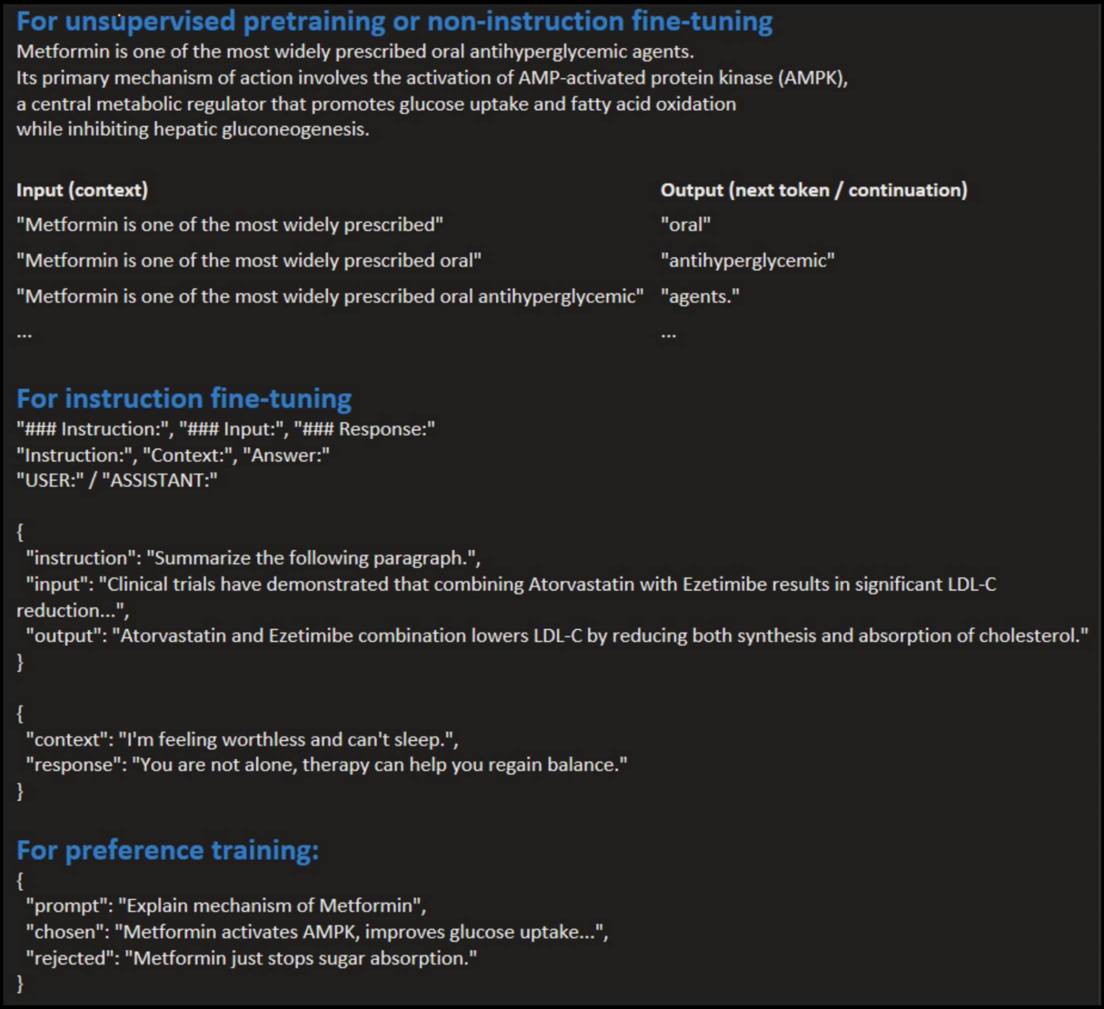
This makes a key difference between a fine-tuned model (only supervised) and an aligned model (like GPT4,5).

Why we do preference training.
(1) It improves safety, ethics
(2) Produces helpful and polite answers.
(3) Teaches the model which responses humans actually prefer.

"How can I lose 10 kg in 1 week?"
"You can stop eating, drink only water, and take fat-burner pills. This will reduce weight quickly."

"Rapid weight loss of 10 kg in a week is unsafe and medically harmful.
A safer approach is 0.5–1 kg per week through balanced diet, hydration, and activity.
If you have urgent concerns, please consult a doctor."

Dataset Example:



For unsupervised pretraining or non-instruction fine-tuning

Metformin is one of the most widely prescribed oral antihyperglycemic agents.
Its primary mechanism of action involves the activation of AMP-activated protein kinase (AMPK),
a central metabolic regulator that promotes glucose uptake and fatty acid oxidation
while inhibiting hepatic gluconeogenesis.

Input (context)	Output (next token / continuation)
"Metformin is one of the most widely prescribed"	"oral"
"Metformin is one of the most widely prescribed oral"	"antihyperglycemic"
"Metformin is one of the most widely prescribed oral antihyperglycemic"	"agents."
...	...

For instruction fine-tuning

"### Instruction:", "### Input:", "### Response:"
"Instruction:", "Context:", "Answer:"
"USER:" / "ASSISTANT:"

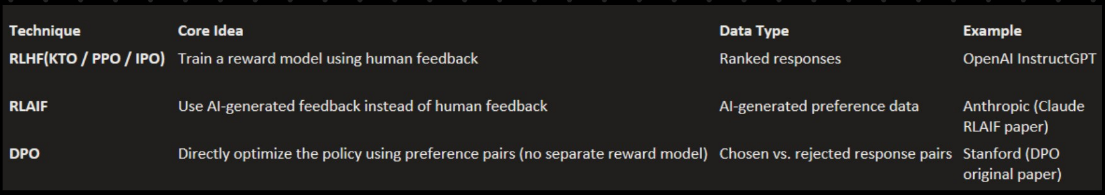
```
{
  "instruction": "Summarize the following paragraph.",
  "input": "Clinical trials have demonstrated that combining Atorvastatin with Ezetimibe results in significant LDL-C reduction...",
  "output": "Atorvastatin and Ezetimibe combination lowers LDL-C by reducing both synthesis and absorption of cholesterol."
}
```

```
{
  "context": "I'm feeling worthless and can't sleep.",
  "response": "You are not alone, therapy can help you regain balance."
}
```

For preference training:

```
{
  "prompt": "Explain mechanism of Metformin",
  "chosen": "Metformin activates AMPK, improves glucose uptake...",
  "rejected": "Metformin just stops sugar absorption."
}
```

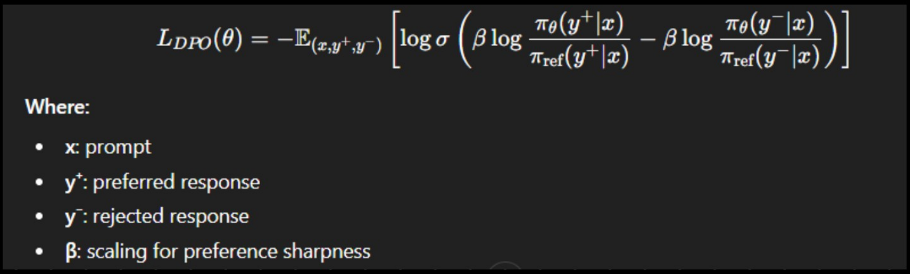
Techniques for Preference Alignment



Technique	Core Idea	Data Type	Example
RLHF(KTO / PPO / IPO)	Train a reward model using human feedback	Ranked responses	OpenAI InstructGPT
RLAIF	Use AI-generated feedback instead of human feedback	AI-generated preference data	Anthropic (Claude RLAIF paper)
DPO	Directly optimize the policy using preference pairs (no separate reward model)	Chosen vs. rejected response pairs	Stanford (DPO original paper)

KTO Kahneman–Tversky Optimization
PPO Proximal Policy Optimization
IPO Implicit Preference Optimization

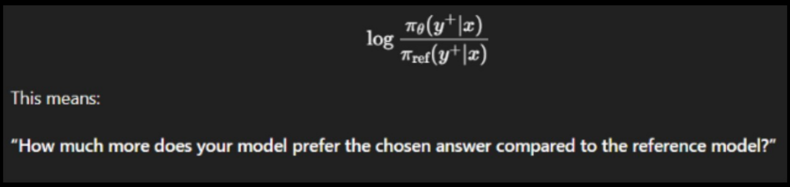
DPO Training Loss intuition



$$L_{DPO}(\theta) = -\mathbb{E}_{(x,y^+,y^-)}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta\log\frac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right)\right]$$

Where:

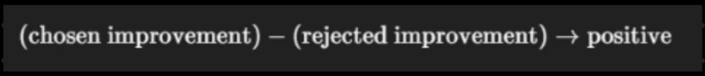
- **x:** prompt
- **y^+:** preferred response
- **y^-:** rejected response
- **β:** scaling for preference sharpness



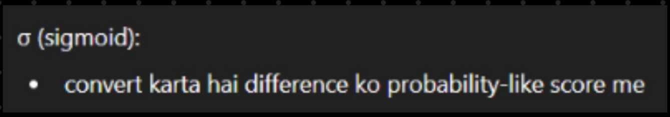
$$\log\frac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)}$$

This means:

"How much more does your model prefer the chosen answer compared to the reference model?"

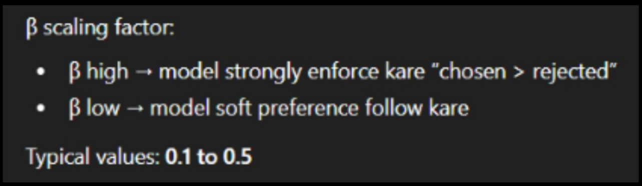


$$(\textbf{chosen improvement}) - (\textbf{rejected improvement}) \rightarrow \textbf{positive}$$



σ (sigmoid):

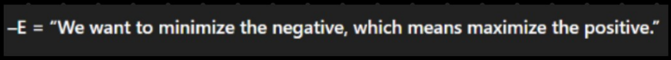
- convert karta hai difference ko probability-like score me



β scaling factor:

- β high → model strongly enforce kare "chosen > rejected"
- β low → model soft preference follow kare

Typical values: **0.1 to 0.5**



–E = "We want to minimize the negative, which means maximize the positive."

During DPO training:

- The model computes the output probability distribution for both the chosen and rejected responses.
- The loss function is adjusted so that P(chosen) > P(rejected).
- After a few epochs, the model automatically begins to prefer more factual and detailed responses.

ChatGPT Preference Alignment

Human Demonstration Phase

Human labelers wrote high-quality responses for each prompt .

Preference Coparison Phase (For Reward Model Training)

Reinforcement Learning / Alignment Step

For the same prompt, multiple model outputs were generated. Human labelers ranked these responses from best to worst. These ranked pairs (chosen vs rejected) trained the Reward Model — the core of RLHF.

Around 40k prompt-comparison pairs were used in the InstructGPT paper.

OpenAI fine-tuned the base model through PPO (Proximal Policy Optimization) to maximize human preference scores — this produced ChatGPT-style alignment (helpful, honest, harmless behavior).

Data Composition & Transparency

The exact dataset (prompts + responses + comparisons) is not publicly released. However, confirmed components include:

- API user prompts
- Annotator-written tasks
- Public Q&A
- Synthetic data

base_model--> non_instruction_model--> instruction_model--> preference_model

$$\text{new weight} = \underbrace{W_{old}}_{\text{Freeze}} + \underbrace{\Delta W}_{\text{LoRA}}$$

train

Patch

base_model --LoRA--> non_instruction_model
non_instruction_model --LoRA--> instruction_model
instruction_model --LoRA--> preference_model

Base Model + LoRA1
(Base LoRA1) + LoRA2
(Base LoRA1 LoRA2) + LoRA3

Stack of LoRA ✗

W_final = (W + ΔW1 + ΔW2 + ΔW3) ✗

Loss unstable
Model will hallucinate
Tuning will not be good
Quality degrade

LoRA is not full layer. It's just a delta patch.
Delta patches cannot be stacked — they must be merged before next training stage.