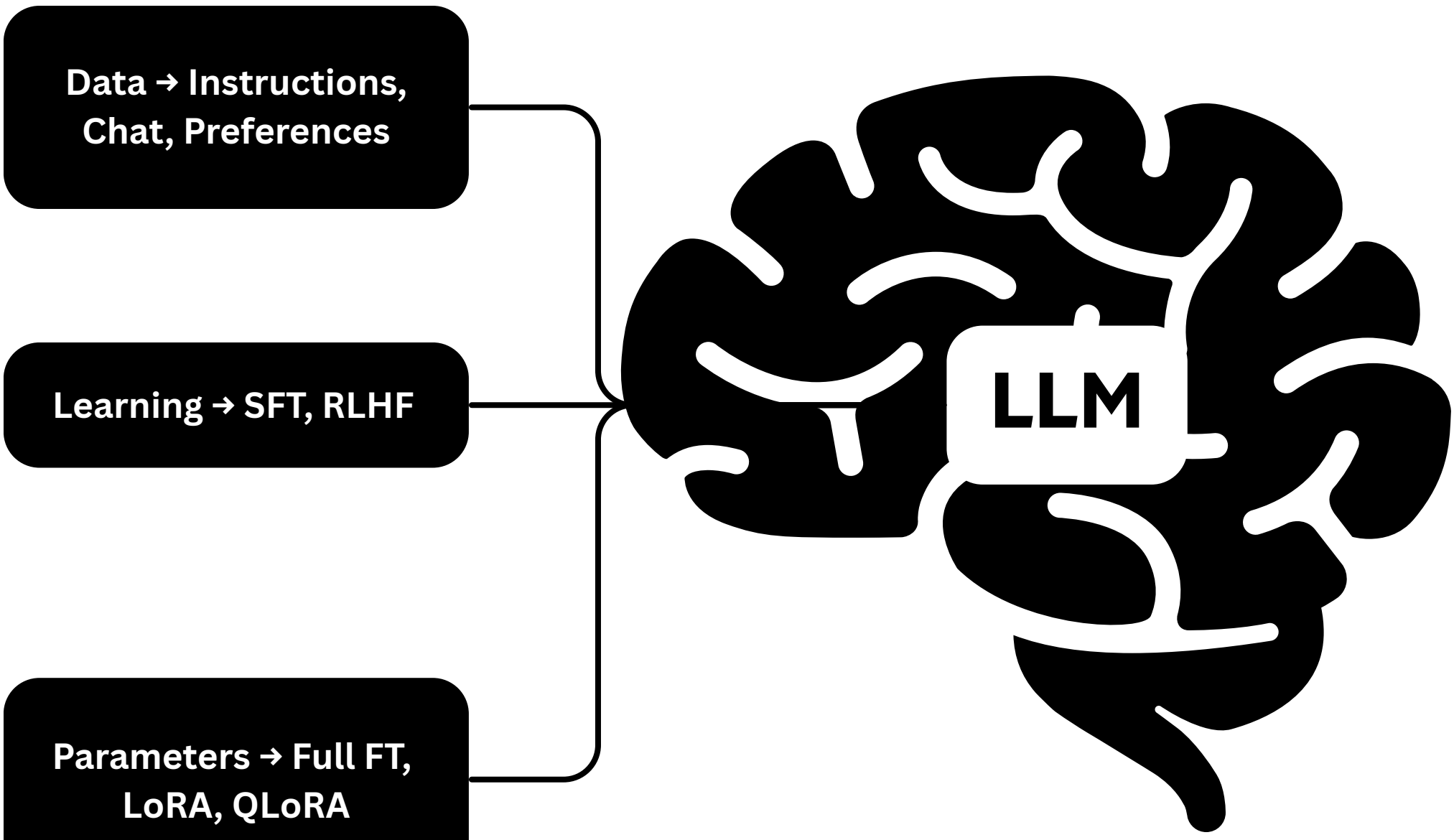
Fine-Tuning LLMs Without the Confusion

How SFT, RLHF, LoRA, QLoRA, and instruction tuning actually fit together

Naresh Edagotti
Follow For More



PracticAI

What happens after pre-training?

An LLM after pre-training only knows how to **predict text**.

It does not know how to:

→ Follow instructions

→ Be helpful

→ Be safe

→ Talk like a chatbot

So we apply fine-tuning.

Fine-tuning = updating a pre-trained model using new data to change its behavior.

Fine-tuning has 2 independent layers

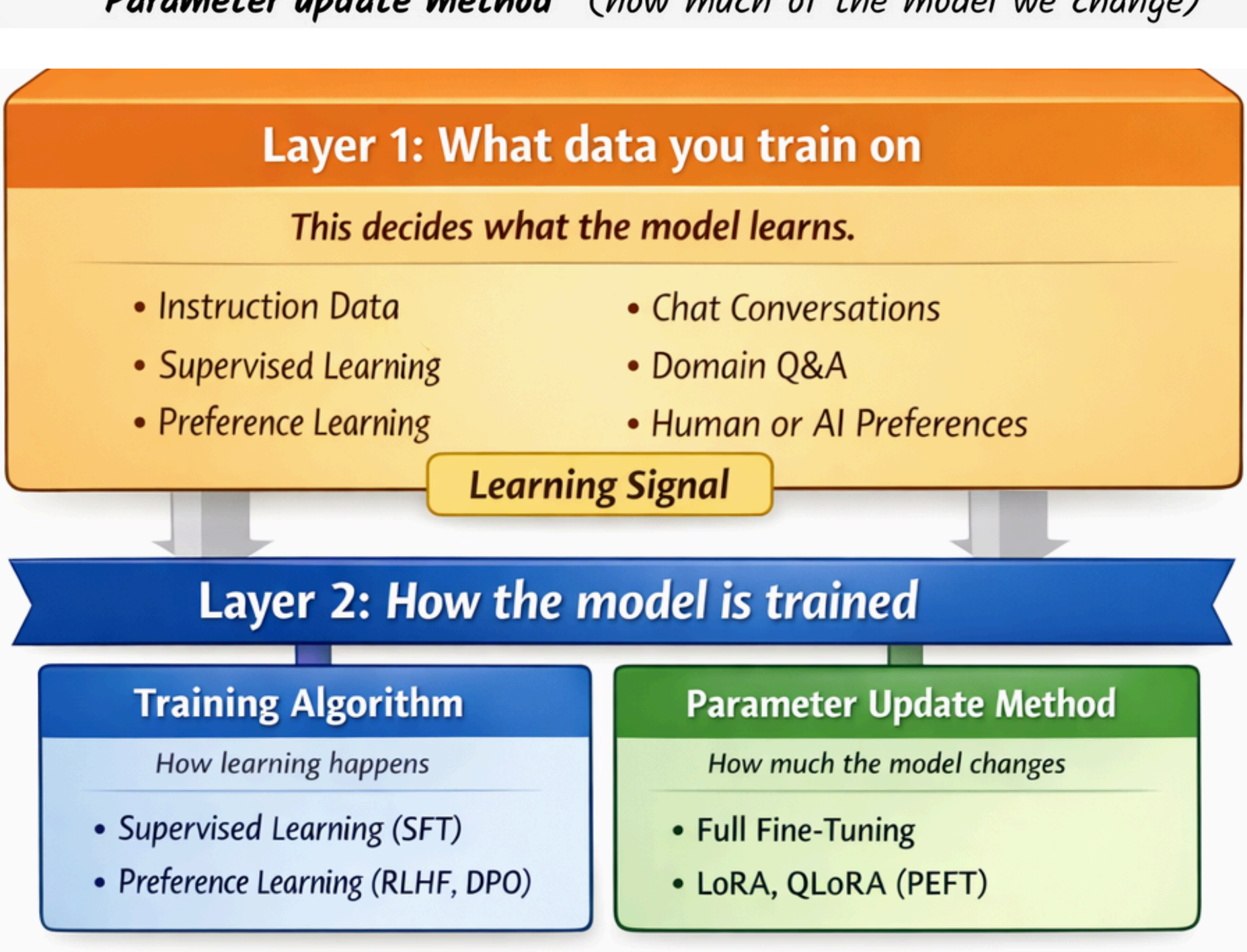
Most confusion comes from **mixing these three**.

FINE-TUNING HAS:

→ **Learning signal** (what data we give)

→ **Training algorithm** (how loss is computed)

Parameter update method (how much of the model we change)



Layer 1: Learning signal

This answers: **what kind of feedback are we training on?**

THERE ARE ONLY TWO REAL ONES AFTER PRE-TRAINING:

A) Supervised Fine-Tuning (SFT)

You give:

```
Input → Correct output
```

B) Preference Learning (RLHF family)

You give:

```
Prompt
Answer A
Answer B
Which is better?
```

Everything else is built on these two.

What is Supervised Fine-Tuning (SFT)?

SFT means:

Train the model to copy correct answers.

DATA LOOKS LIKE:

```
"Summarize this article" → "Correct summary"
```

```
"What is RAG?" → "Correct explanation"
```

The model learns using **normal cross-entropy loss**.

Instruction tuning is not new

PEOPLE SAY:

→ Instruction tuning

→ Chat tuning

→ Domain tuning

These are not different techniques.

They are just **different SFT datasets.**

TYPE	WHAT THE DATA LOOKS LIKE
Instruction tuning	"Do X" → "Response"
Chat tuning	User + Assistant messages
Domain tuning	Legal, medical, finance Q&A

All of them are still SFT.

What is RLHF?

RLHF is not supervised learning.

Instead of correct answers, we collect **preferences**.

Example:

```
Prompt: "Explain AI"

Answer A
Answer B

Human: A is better
```

*The model learns what humans **prefer**, not what is "correct".*

How RLHF works

PROCESS

→ Train a **reward model** from human preferences

→ Use **reinforcement learning** to push the LLM to get higher reward

This changes:

→ Helpfulness

→ Tone

→ Safety

→ Politeness

→ Style

More than **factual knowledge**.

DPO, PPO, RLAIF

These are all **RLHF-style methods**.

NAME	MEANING
PPO	Classic RLHF algorithm
DPO	Simpler preference optimization
RLAIF	AI gives the preferences instead of humans

They all use preference data, not labels.

Layer 2: How much of the model changes

This is where **LoRA and QLoRA** live.

This answers:

How many parameters do we train?

TWO MAIN APPROACHES:

→ **Full Fine-Tuning:** Update all parameters

→ **PEFT:** Update only a small subset

Full Fine-Tuning

All model weights are updated.

CHARACTERISTICS

→ Highest flexibility

→ Very expensive

→ Needs lots of data and GPUs

→ Used by big labs

PEFT (Parameter-Efficient Fine-Tuning)

Instead of changing all weights, we add **small trainable layers**.

THIS INCLUDES:

→ LoRA

→ QLoRA

→ Adapters

→ Prefix tuning

Base model stays frozen.

LoRA vs QLoRA

METHOD	WHAT IT DOES
LoRA	Train small low-rank matrices
QLoRA	Same, but base model is 4-bit quantized

QLoRA is much cheaper, almost same quality.

How everything fits

A real LLM pipeline looks like:

→ **1. Pretrain** (text prediction)

→ **2. SFT** (instruction + chat data)

→ **3. RLHF / DPO** (human or AI preferences)

Each step can use:

→ Full fine-tuning

→ or

→ PEFT (LoRA / QLoRA)

One clean formula

Fine-tuning =

Learning signal (SFT or RLHF)

×

Training method (cross-entropy or preference optimization)

×

Weight update style (Full FT or PEFT)

Final takeaway

→ Instruction tuning is **not** a new algorithm.

→ LoRA is **not** a learning method.

→ RLHF is **not** supervised fine-tuning.

Once you separate:

→ **Data**

→ **Learning**

→ **Parameters**

The whole fine-tuning world becomes simple.