SESSION 3

Fine-Tuning in LLMs

UniDive cost EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

RECAP

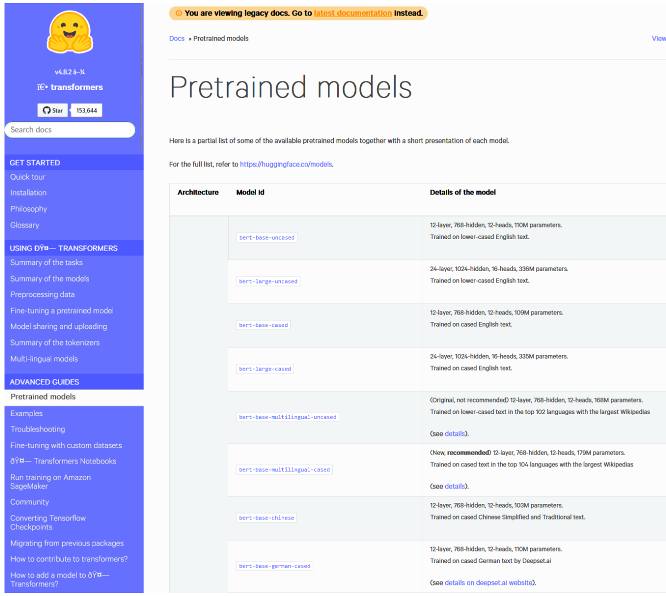
- Prompt engineering influences the quality, relevance and accuracy of generative AI outputs.

- CoT enables complex reasoning capabilities through intermediate reasoning steps.

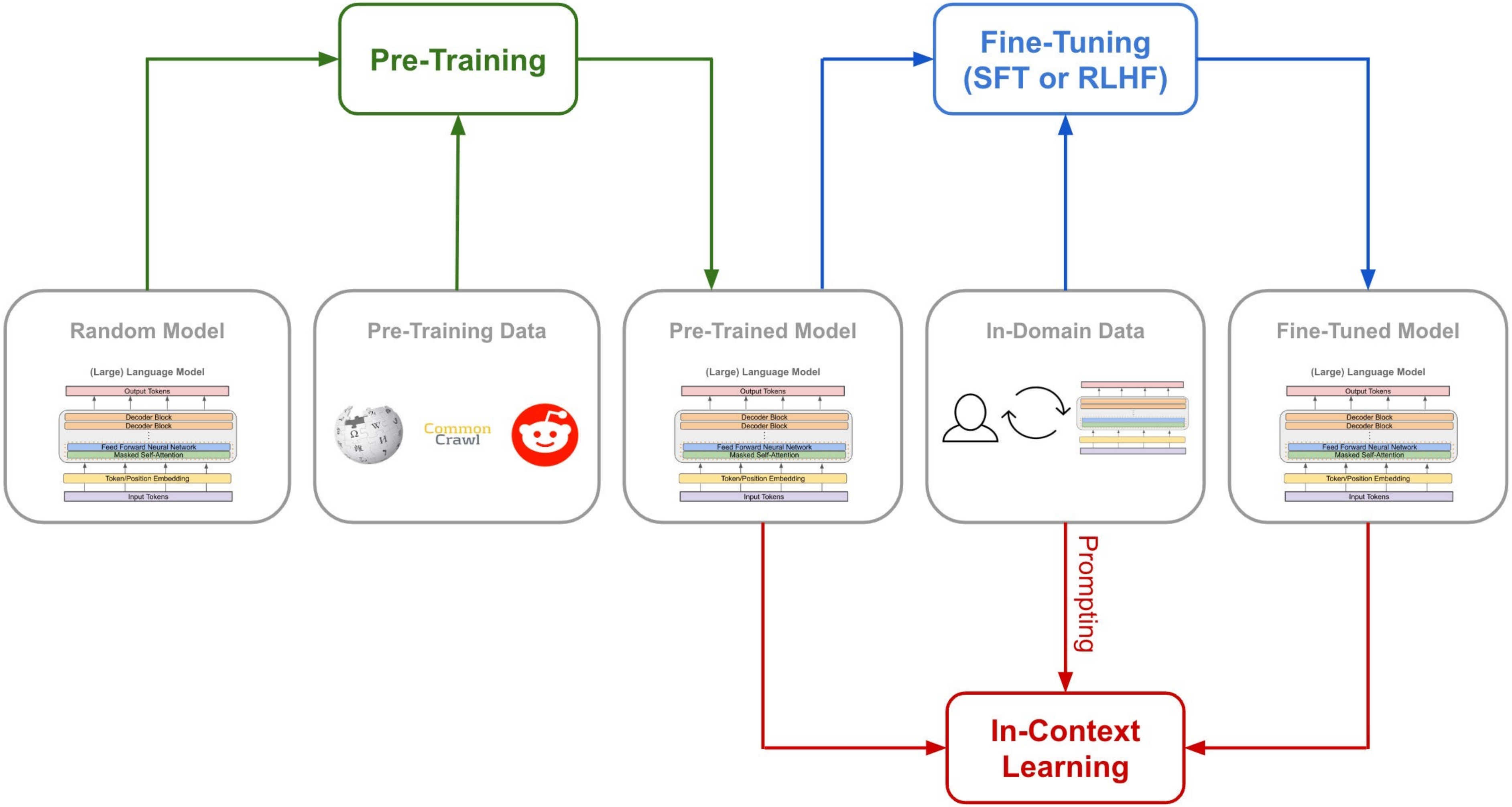
- Delimiters help structure your prompt.

- Low-resource language challenges: dialects, code-switching, data scarcity, bias

FINE-TUNING



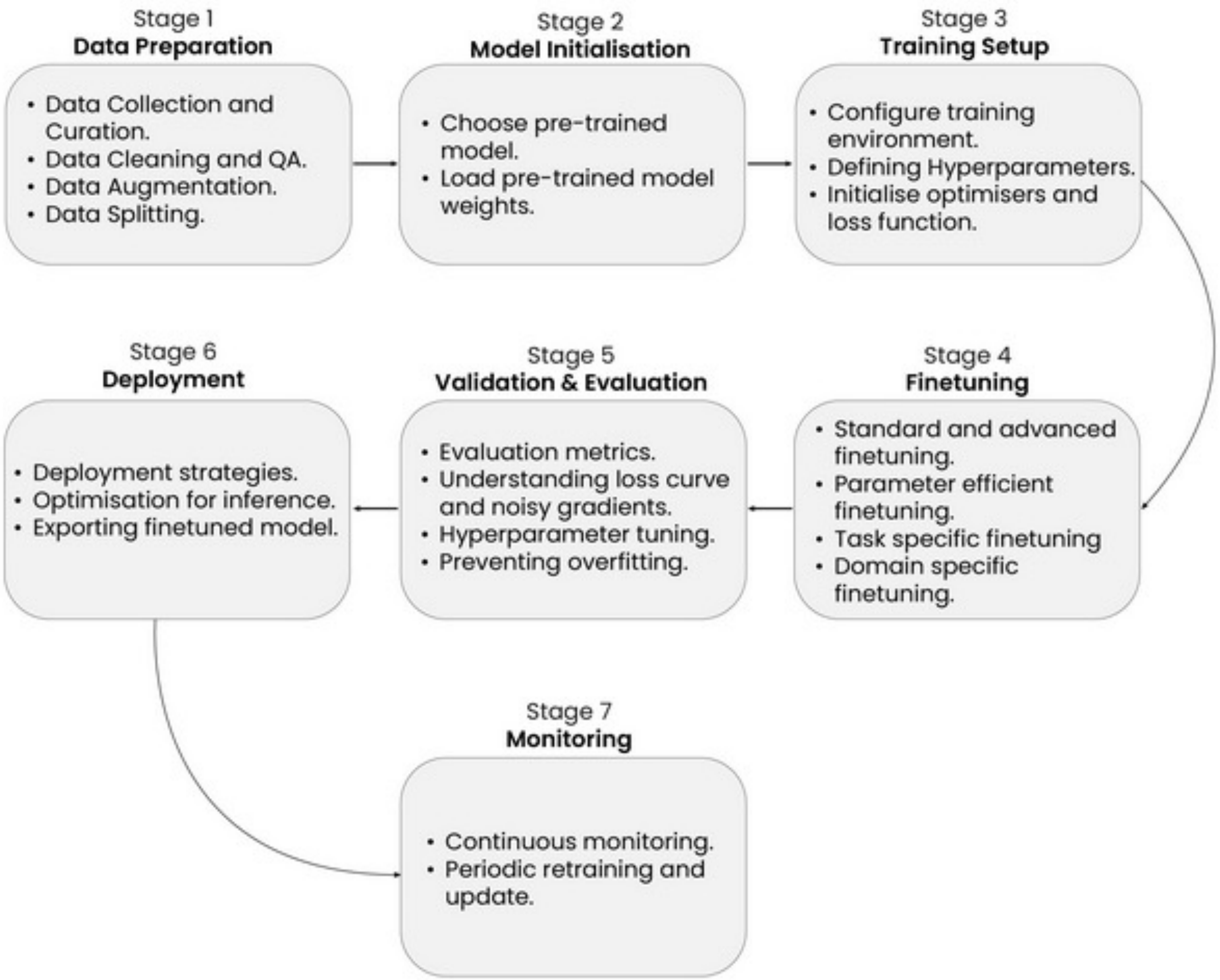
- Fine-tuning is the process of taking a pre-trained model and further training it on a **domain-specific dataset**.
- Transformers grant access to an extensive collection of pre-trained models suited for various tasks.
- Fine-tuning these models is a crucial step for improving the model's ability to perform specific tasks, such as sentiment analysis, question answering, or document summarization.

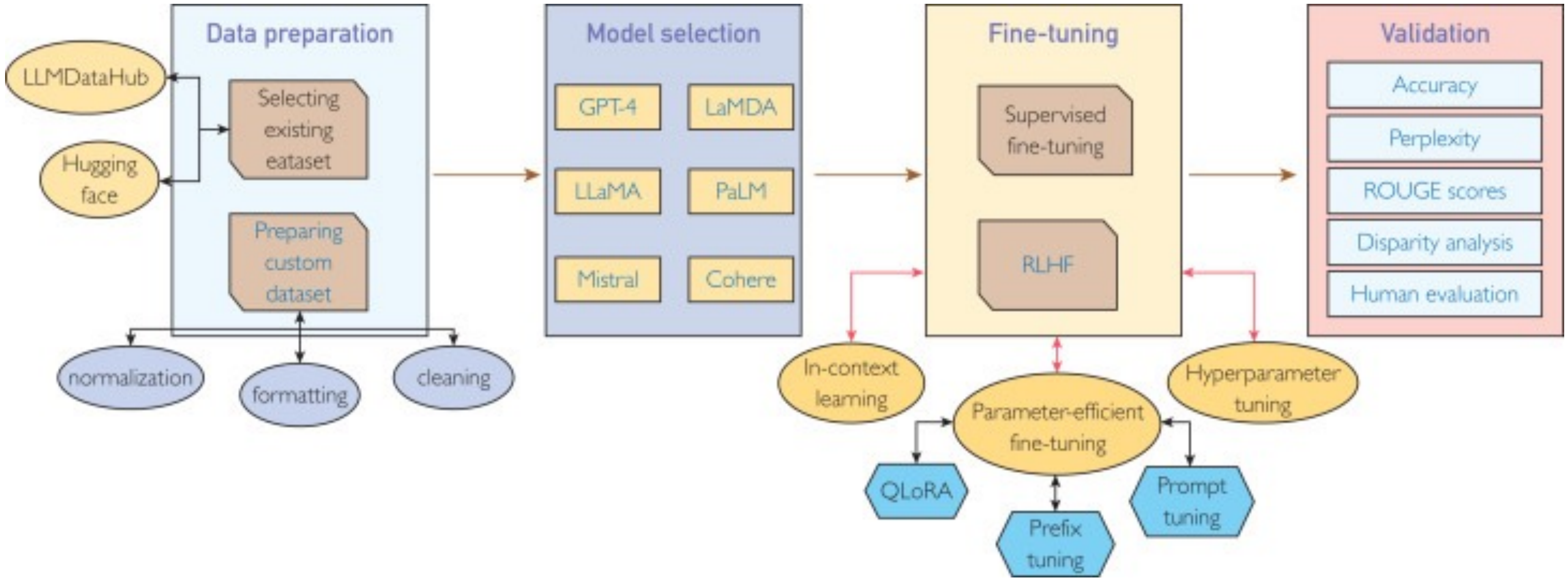

Pre-Training
Fine-Tuning
(SFT or RLHF)
Random Model
(Large) Language Model
Pre-Training Data
Common
Crawl
Pre-Trained Model
(Large) Language Model
In-Domain Data
Fine-Tuned Model
(Large) Language Model
Prompting
In-Context
Learning

Aspect	**Pre-training**	**Fine-tuning**
Definition	Training on a vast amount of unlabelled text data	Adapting a pre-trained model to specific tasks
Data Requirement	Extensive and diverse unlabelled text data	Smaller, task-specific labelled data
Objective	Build general linguistic knowledge	Specialise model for specific tasks
Process	Data collection, training on large dataset, predict next word/sequence	Task-specific data collection, modify last layer for task, train on new dataset, generate output based on tasks
Model Modification	Entire model trained	Last layers adapted for new task
Computational Cost	High (large dataset, complex model)	Lower (smaller dataset, fine-tuning layers)
Training Duration	Weeks to months	Days to weeks
Purpose	General language understanding	Task-specific performance improvement
Examples	GPT, LLaMA 3	Fine-tuning LLaMA 3 for summarisation

7-STAGE PIPELINE

- Dataset preparation
- Model initialisation
- Training Environment Setup
- Partial or full-time fine tuning
- Evaluation and Validation
- Deployment
- Monitoring and Maintenance


Stage 1
Data Preparation
• Data Collection and Curation.
• Data Cleaning and QA.
• Data Augmentation.
• Data Splitting.
Stage 2
Model Initialisation
• Choose pre-trained model.
• Load pre-trained model weights.
Stage 3
Training Setup
• Configure training environment.
• Defining Hyperparameters.
• Initialise optimisers and loss function.
Stage 4
Finetuning
• Standard and advanced finetuning.
• Parameter efficient finetuning.
• Task specific finetuning
• Domain specific finetuning.
Stage 5
Validation & Evaluation
• Evaluation metrics.
• Understanding loss curve and noisy gradients.
• Hyperparameter tuning.
• Preventing overfitting.
Stage 6
Deployment
• Deployment strategies.
• Optimisation for inference.
• Exporting finetuned model.
Stage 7
Monitoring
• Continuous monitoring.
• Periodic retraining and update.



(Anisuzzaman et al., 2025)

TYPES OF FINE-TUNING

- Supervised or semi-supervised fine-tuning
- RLHF (Reinforcement Learning from Human Feedback)
- Few-shot learning
- Transfer learning
- Domain-specific fine-tuning

SUPERVISED FINE-TUNING (SFT)

- The model is further trained on a labeled dataset specific to the **target task** to perform, such as text classification or named entity recognition.

SFT CHALLENGES

- **Overfitting:** The model can overfit on the fine-tuning dataset and perform poorly on unseen examples.
- **Hyperparameter tuning:** SFT might be more efficient than pre-training, but the hyperparameter tuning operations can be cumbersome. Hyperparameter configurations: learning rate, batch size, and sequence length.
- **Data quality issues:** Model performance after fine-tuning largely depends on the quality of the training data.
- **Catastrophic forgetting:** Since SFT updates model weights directly, the update procedure may overwrite the previous knowledge learned during pre-training.

SELF-SUPERVISION

- In self-supervision, instead of training a model to perform a task that requires explicit annotations, the model learns from the vast amounts of unlabeled data available, extracting patterns and understanding context without human intervention.

Aspect	Supervised Fine-Tuning	Self-Supervised Fine-Tuning
Key difference	Uses human-provided labels	Uses labels generated from the data itself
Learning signal	External ground truth	Intrinsic structure of the data
Annotation required	Yes	No (or minimal)

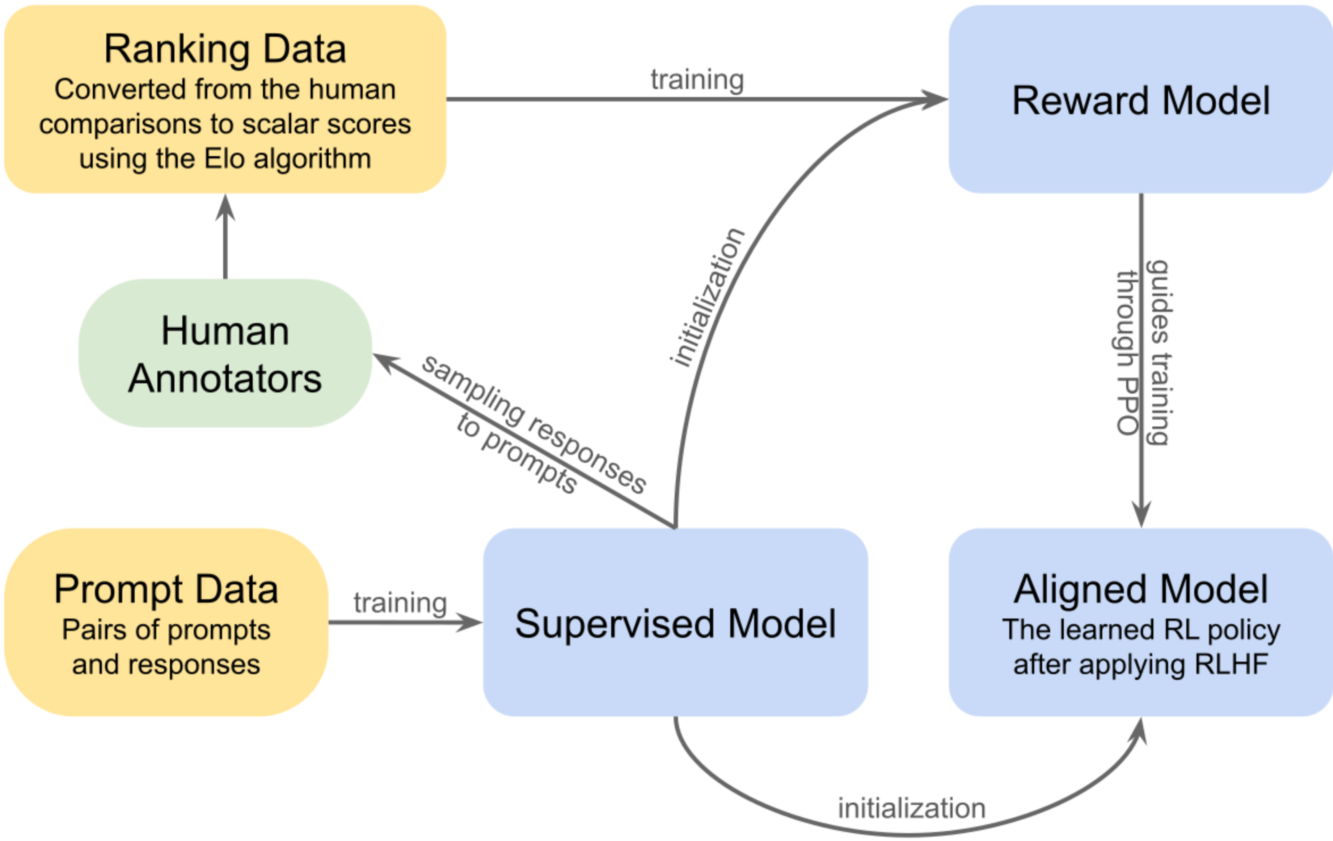
REINFORCEMENT LEARNING FROM HUMAN FEEDBACK (RLHF)

This method uses the knowledge of human evaluators; in addition, it also allows the model to adjust and develop in response to real-world input, Some standard RLHF techniques are:

- *Reward modelling*: the model generates multiple potential outputs or actions, which are subsequently assessed by human evaluators who assign a **ranking or rating** on the basis of their quality.
- *Proximal policy optimization:* A policy refers to the **strategy** or set of rules that a reinforcement learning agent uses to make decisions in an environment.

REINFORCEMENT LEARNING FROM HUMAN FEEDBACK(RLHF)

- *Comparative ranking*: the model produces several outputs or actions, which human investigators then rank according to **compatibility** or **quality**. The model then modifies its behavior to generate higher-ranked outputs.
- *Preference feedback*: This technique involves the model generating several outputs and human experts **selecting among them**. This method is useful when assigning a numeric value (reward) to an output is difficult.


Ranking Data
Converted from the human comparisons to scalar scores using the Elo algorithm
training
Reward Model
initialization
guides training through PPO
Human Annotators
sampling responses to prompts
Prompt Data
Pairs of prompts and responses
training
Supervised Model
Aligned Model
The learned RL policy after applying RLHF
initialization

FEW-SHOT LEARNING

- Few-shot learning tries to address this by providing a few examples (or shots) of the required task at the beginning of the input prompts.
- This helps the model have a better context of the task without an extensive fine-tuning process.

ONE-SHOT LEARNING

- LLMs are given **a single example** of what to do.
- **Example Prompt:** “Translate English to French: ‘Hello’ → ‘Bonjour’, ‘Goodbye’ →”
- **When to use:** When task format is uncommon or nuanced and benefits from one clear reference.


Zero-shot, One-shot, Few-shot Learning
Zero-shot
Translate "Good morning" to French
One-shot
"Hello" → "Bonjour"
Now translate: "Goodbye" →
Few-shot
"Hello" → "Bonjour"
"Yes" → "Oui"
"No" → "Non"
"Thank you" →

TRANSFER LEARNING

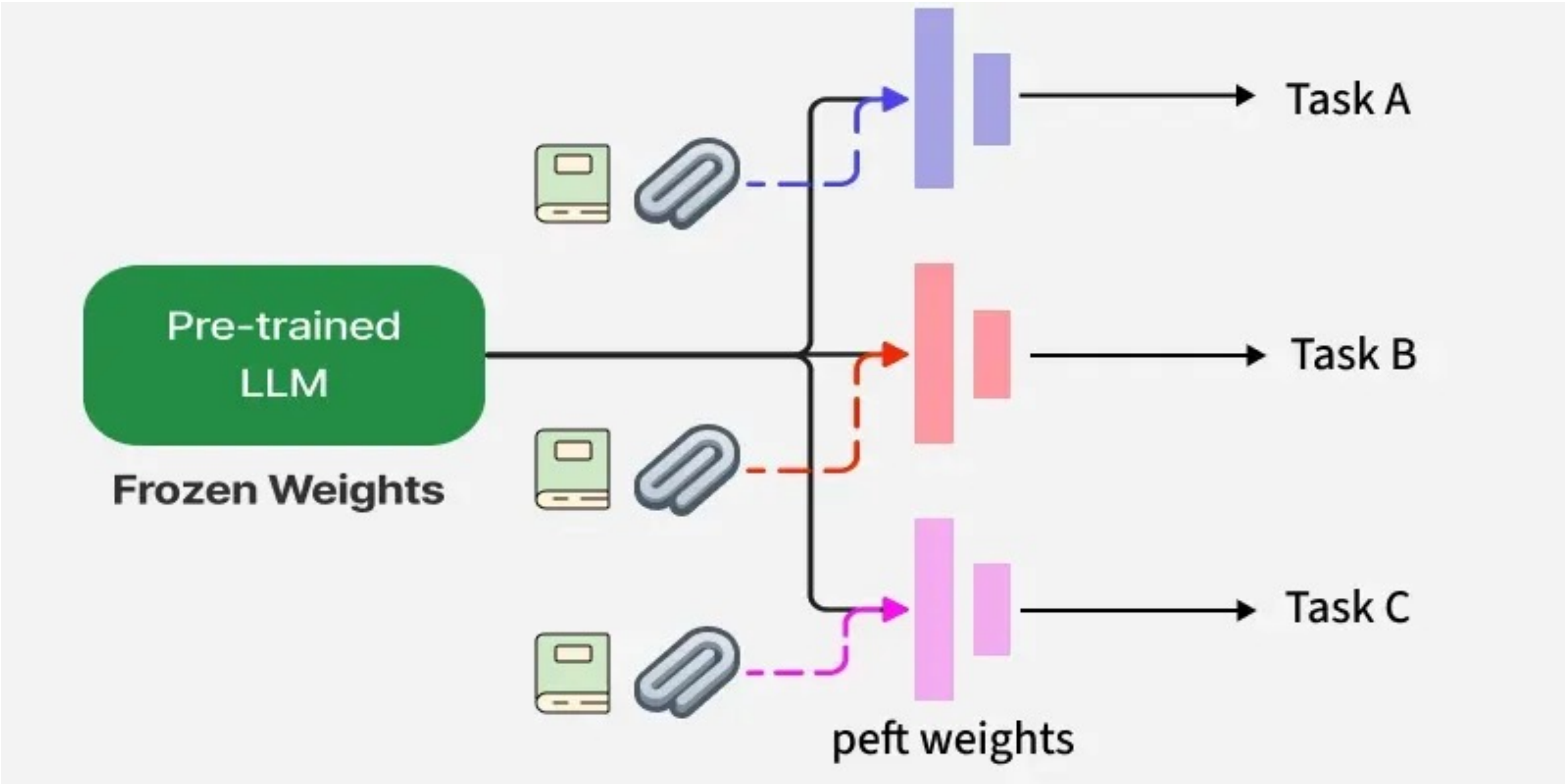
- Even though all fine-tuning techniques are a form of transfer learning, this category is specifically aimed to allow a model to perform a task different from the task it was initially trained on.
- The main idea is to leverage the knowledge the model has gained from a large, general dataset and apply it to a more specific or related task.

DOMAIN-SPECIFIC FINE-TUNING

- The model is fine-tuned on a dataset composed of text from the target domain to improve its context and knowledge of domain-specific tasks.
- For instance, to generate a chatbot for a medical app, the model would be trained with medical records, to adapt its language understanding capabilities to the health field.

PARAMETER-EFFICIENT FINE-TUNING (PEFT)

- **PEFT** freezes most of the pretrained model and adds or modifies **a small number of trainable parameters**
- PEFT balances efficiency and performance to help organizations maximize computational resources while minimizing storage costs.
- If a base model is too large to completely retrain or if the new task is different from the original, PEFT can be an ideal solution.


Pre-trained
LLM
Frozen Weights
Task A
Task B
Task C
peft weights

PEFT TECHNIQUES

- AI teams have various PEFT techniques and algorithms at their disposal, each with its relative advantages and specializations. Many of the most popular PEFT tools can be found on Hugging Face and numerous other GitHub communities.
  - Adapters
  - LoRA
  - QLoRA
  - Prefix-tuning
  - Prompt-tuning
  - P-tuning

LORA (LOW-RANK ADAPTION)

- LoRA is a technique used to adapt ML models **to new contexts.** It can adapt large models to specific uses by adding lightweight pieces to the original model rather than changing the entire model.
- We can quickly expand the ways that a model can be used rather than requiring them to build an entirely new model.
- As an example, a full fine-tuning of the GPT-3 model requires training **175 billion parameters** because of the size of its training dataset.
- Using LoRA, the trainable parameters for GPT-3 can be reduced to roughly **18 million parameters**.

INSTRUCTION TUNING

- **An instruction:** A natural language text *input* that specifies a given task. For example, "*translate this sentence from English to Spanish.*"
- **Additional information:** Optional, supplementary information that provides context relevant to the task at hand. For example, an input for a reading comprehension task might include a brief passage (and then instruct the model to answer a given question about it).
- **Desired output:** The target *output*—response—for the given prompt, per the instructions and context provided. This will serve as a ground truth against which the model's predictions are evaluated and optimized.

PREFERENCE OPTIMIZATION

- **Preference optimization** is a training approach where a model is optimized to **choose outputs that humans prefer**, rather than to match a single "correct" label.
- Instead of "This answer is correct.", the output is "Answer A is better than Answer B."

BENEFITS OF FINE-TUNING

- Domain-specific knowledge
- Specific task optimization
- Data efficiency
- Better performance
- Resource efficiency

(Anisuzzaman et al., 2025)

0. 🏗️ Fine-Tuning Fundamentals: Theory and Practice

0.1 Fine-Tuning Taxonomy: A Systematic Overview

Fine-tuning is the process of adapting a pretrained language model to specialized tasks or domains using additional labeled data. Understanding the landscape of approaches is crucial for making informed decisions.

Approach	Parameters Updated	Memory Requirement	Training Speed	Best For	Cost
🔥 **Full Fine-tuning**	All parameters (100%)	Very High (4x model size)	Slow	High-resource tasks	$
⚡ **Parameter-Efficient (PEFT)**	Small subset (0.1-10%)	Low (1.2x model size)	Fast	Low-resource languages	$$
🎯 **LoRA**	Low-rank adapters (~1%)	Very Low	Very Fast	Most practical cases	$
📚 **Instruction Tuning**	Task-specific layers	Medium	Medium	Following instructions	$$$
🎪 **Preference Optimization**	Value/reward layers	Medium	Medium	Human alignment	$$$