# **Large Language Model Fine-Tuning Techniques: A Comprehensive Explanation**

Large Language Models (LLMs) have emerged as powerful tools demonstrating remarkable advancements in the field of natural language processing.1 Their ability to understand and generate human-like text has revolutionized various applications. However, to effectively harness the capabilities of these models for specific real-world scenarios, a process known as fine-tuning is often essential.2 This adaptation process allows for the customization of LLMs to adhere to particular domain-specific requirements, enhance their reliability, and foster greater user trust.2 This process bridges the gap between the broad general knowledge acquired by pre-trained models and the distinct needs of specialized applications, ensuring a closer alignment between the language model's performance and human expectations.10 Pre-trained LLMs, while possessing a vast understanding of language, often lack the specific expertise or nuanced comprehension necessary for optimal performance in targeted domains. Fine-tuning addresses this by leveraging the foundational knowledge already embedded within the model and adapting it through further training on task-specific data, ultimately leading to improved accuracy and relevance in the desired context.

This report aims to provide a comprehensive explanation of several key techniques employed for fine-tuning LLMs. These include Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), Low-Rank Adaptation (LoRA), Quantized LoRA (QLoRA), and instruction fine-tuning. Furthermore, this analysis will clarify the categorization of these techniques and their interrelationships, addressing the question of whether they follow a hierarchical structure or a two-level selection process. Finally, other significant techniques utilized in the fine-tuning of LLMs will be discussed to provide a holistic understanding of this critical area in natural language processing.

## **Supervised Fine-Tuning (SFT)**

### **Definition and Core Principles**

Supervised Fine-Tuning (SFT) represents a fundamental methodology for adapting a pre-trained LLM to excel in a specific downstream task. This is achieved by subjecting the model to further training using a dataset tailored to the particular task.2 The effectiveness of SFT hinges on the utilization of high-quality labeled datasets, which contain input-output pairs meticulously curated by human experts. These datasets serve to optimize the LLM's internal parameters, ultimately leading to enhanced accuracy in the targeted application.2 The designation "supervised" in SFT underscores the nature of the training data, which explicitly includes both input examples and their corresponding desired outputs, providing a clear learning signal for the model.2 This method is a crucial step in transitioning general-purpose Foundation Models into specialized tools capable of delivering high accuracy and relevance when applied to domain-specific data.2 While pre-trained models gain a broad understanding of language from extensive datasets, SFT refines their capabilities to achieve superior performance in focused areas by learning the specific patterns and subtle nuances inherent in the labeled examples provided.

### **The SFT Process (Steps)**

The process of refining an LLM using SFT typically commences with a model that has already undergone pre-training. This initial phase equips the model with a general understanding of language, having been trained on a massive and diverse collection of text. Such a model, often referred to as a base model, has acquired a broad grasp of grammar, contextual understanding, semantic meaning, and even general world knowledge.2 SFT leverages this existing linguistic competence, building upon a solid foundation rather than initiating the learning process from scratch.2

Following the selection of a pre-trained model, the next vital stage involves the careful curation of a high-quality labeled dataset. This dataset comprises pairs of inputs and their corresponding desired outputs, specifically chosen to align with the requirements of the intended task.2 The success of the SFT process is strongly linked to the quality of this training data.2 For instance, if the objective is sentiment analysis, the dataset might include sentences paired with labels such as "positive" or "negative."

The subsequent training phase involves further training the pre-trained LLM specifically on this meticulously curated labeled dataset. This training adheres to the principles of supervised learning.2 During this phase, for each input in the dataset, the LLM generates its own predicted output. A loss function, commonly cross-entropy loss, is then employed to calculate the difference between the LLM's predicted output and the desired "ground truth" output provided in the labeled dataset. This calculated loss serves as a measure of how "wrong" the model was for that particular example. The model then utilizes this loss to adjust its internal parameters, specifically the weights and biases, through a process called back-propagation, guided by an optimization algorithm such as Adam. This cycle of generating a prediction, calculating the loss, and adjusting the model's parameters is repeated numerous times across the entire labeled dataset, spanning multiple epochs. This iterative refinement process gradually pushes the model to minimize the loss and produce outputs that more closely match the desired labeled outputs.2 This direct feedback mechanism is instrumental in guiding the adaptation of the pre-trained model to the specific nuances of the target task by effectively minimizing the discrepancy between its predictions and the provided labeled data.2 This enables the model to learn the intended input-output relationships and enhance its performance on the designated task.

### **Common Use Cases for SFT**

Supervised fine-tuning finds extensive application across a multitude of scenarios where the tasks are well-defined, often rule-based, and necessitate domain-specific expertise.2 One prominent use case lies in the creation of intelligent and brand-aligned chatbots designed for customer support automation.2 SFT enables these chatbots to offer more than just generic responses, providing consistent, accurate, and brand-appropriate communication. In the medical domain, SFT plays a crucial role in efficiently extracting key information from patient records, assisting healthcare professionals in diagnosis, treatment planning, and risk assessment.2 Furthermore, SFT-enhanced LLMs serve as valuable diagnostic support tools, capable of analyzing patient symptoms and medical history to provide potential diagnoses for clinicians to consider.2 The adaptability of SFT extends to improving model performance on tasks specific to particular domains, such as the analysis of legal documents or the comprehension of medical language.22 The versatility inherent in SFT allows it to be applied to a wide spectrum of NLP tasks, empowering pre-trained models to excel in various specialized fields.2 By undergoing fine-tuning on datasets tailored to specific requirements, general-purpose language models can be transformed into highly specialized tools, precisely engineered to meet the unique demands of diverse applications.

### **Key Techniques in SFT**

Within the realm of SFT, several advanced techniques are employed to further optimize the performance and efficiency of the models.22 **Full model fine-tuning** involves updating all the parameters of the pre-trained model using the labeled data.7 This comprehensive approach allows the model to fully adapt its learned representations to the nuances of the new task. **Layer-wise fine-tuning** offers a more granular approach, selectively adjusting parameters in specific layers of the model, often based on the intuition that different layers capture different levels of linguistic abstraction.22 **Feature-based fine-tuning** takes a different tack, utilizing the pre-trained model primarily as a feature extractor. In this method, the pre-trained model processes the input data, and its intermediate or final layer outputs are then fed into a separate, often simpler, model (like a classifier) which is trained on these extracted features.9

**Instruction fine-tuning** stands out as a specialized variant of SFT. It focuses on training the model using data specifically formatted as instructions paired with desired responses. This technique aims to significantly improve the model's ability to interpret and execute a wide variety of natural language instructions.3

Furthermore, to address the computational challenges associated with fine-tuning large models, **Parameter-Efficient Fine-Tuning (PEFT)** techniques have gained prominence. These methods, including **Low-Rank Adaptation (LoRA)** and adapter-based approaches, are designed to minimize the number of model parameters that need to be updated during the fine-tuning process, thereby leading to greater efficiency and reduced resource consumption.3 The foundational principles of **transfer learning** are inherently at play in SFT, as the process explicitly leverages the knowledge and representations that the model has already acquired during its initial pre-training phase.9

## **Reinforcement Learning from Human Feedback (RLHF)**

### **Definition and Motivation**

Reinforcement Learning from Human Feedback (RLHF) represents a sophisticated machine learning technique that leverages direct human input to refine and optimize AI models. This approach enables these models to learn with greater efficiency and to align their behavior more closely with the intricate preferences of humans.57 Unlike traditional supervised learning paradigms, which rely on pre-existing labeled datasets, RLHF allows models to learn through an interactive process of trial-and-error, with human feedback serving as a crucial guiding signal.59 This methodology proves particularly advantageous for tasks where the desired outcome is challenging to articulate through explicit rules or quantifiable metrics but is readily discernible and evaluable by human judgment. Examples of such tasks include generating text that is not only helpful and informative but also adheres to principles of harmlessness and honesty.41 RLHF signifies a fundamental evolution in the training of AI systems, aiming to ensure that their behavior is congruent with human values and preferences, effectively bridging the inherent gap between artificial intelligence and the nuanced expectations of human users.59 By directly incorporating human evaluations into the reward mechanisms that steer the AI's learning trajectory, RLHF can imbue AI systems with a more profound understanding of ethical considerations, prevailing social norms, and common-sense reasoning.

### **The RLHF Process (Stages)**

The process of training an LLM using RLHF typically unfolds across several distinct stages. Initially, the model often undergoes **pre-training** on a substantial corpus of text. This foundational step allows the model to develop a broad understanding of language structure and patterns.58 Following this, a crucial second stage frequently involves **supervised fine-tuning (SFT)** of the pre-trained model.57 This step is designed to prepare the model to generate responses in a format that aligns with user expectations. Often, this is achieved through instruction tuning, where the model is trained on labeled examples consisting of (prompt, response) pairs crafted by human experts.57 SFT plays a vital role within RLHF by establishing an initial policy that already exhibits a degree of alignment with human intent. This pre-alignment makes the subsequent reinforcement learning process more efficient, as the model is already capable of producing reasonable outputs, allowing RLHF to focus on refining these outputs based on human preferences rather than learning the task from the ground up.57

The core of the RLHF process lies in the **development of a separate reward model** that is trained using human feedback.57 This involves gathering data where human evaluators express their preferences between multiple responses generated by the language model in response to the same prompt.57 These preference judgments are then used to train the reward model. The goal of this model is to learn to automatically predict how a human would rate any given response to a prompt.57 The reward model serves as a learned approximation of human preferences, enabling the AI system to discern which types of responses are considered desirable by humans.57 This automated preference scoring is essential for scaling the learning process beyond the limitations of direct human evaluation.

Finally, the language model undergoes **optimization using the reward model**.57 Employing reinforcement learning algorithms, the language model refines its internal policy to generate responses that are likely to receive high reward scores from the reward model. This process effectively aligns the model's behavior with human preferences in a more optimized manner.57 Techniques such as Proximal Policy Optimization (PPO) are frequently utilized during this optimization stage.62

### **Applications of RLHF**

RLHF has proven to be a transformative technique in the development of sophisticated conversational AI models, most notably exemplified by ChatGPT. It has enabled these models to interpret human instructions with a high degree of naturalness and to exhibit judgment capabilities that approach human quality.59 Beyond its success in crafting advanced chatbots, RLHF possesses broad applicability across various domains. In AI image generation, it can be employed to evaluate and refine the realism or the specific artistic style of generated images.57 Similarly, in music generation, RLHF can assist in creating musical pieces that effectively evoke desired moods and complement specific activities.57 Furthermore, content platforms can significantly enhance their moderation tools by leveraging RLHF. This allows the models to dynamically adapt to evolving policy changes based on ongoing human feedback.2 The combined effect of structured instruction adherence and iterative refinement through human feedback can result in a model that follows instructions accurately and provides more contextually relevant and user-friendly responses.69

### **Challenges of RLHF**

Despite its considerable effectiveness, RLHF presents several notable challenges. The acquisition of sufficient and high-quality human feedback can be a time-consuming, labor-intensive, and ultimately costly endeavor.62 Moreover, the inherent subjectivity of human preferences can introduce biases into the reward model. If the feedback collection process is not meticulously designed and managed, this can potentially lead to unintended or undesirable outcomes.63 Ensuring that models trained with RLHF exhibit robust generalization capabilities when encountering new and unseen contexts, and that they avoid the generation of nonsensical or factually incorrect outputs (a phenomenon known as hallucinations), remains an active area of research and development.70 Additionally, the computational resources demanded by RLHF can be substantial, which may limit its accessibility to organizations or individuals with constrained computational infrastructure.59

## **Parameter-Efficient Fine-Tuning (PEFT) Techniques**

### **Introduction to PEFT**

Parameter-Efficient Fine-Tuning (PEFT) represents a collection of methodologies engineered to adapt expansive pre-trained models for specific tasks while requiring the training of a significantly smaller number of parameters compared to traditional full fine-tuning approaches.3 These techniques are specifically designed to curtail computational costs and memory demands during the fine-tuning process, all while maintaining or even enhancing the model's performance on downstream tasks.75 PEFT methods are particularly advantageous when dealing with very large language models, where the conventional approach of full fine-tuning can become prohibitively expensive and resource-intensive.1 By strategically focusing on updating only a small subset of the model's parameters, PEFT techniques drastically reduce the memory footprint and computational power needed for fine-tuning, thereby enabling experimentation and deployment on hardware with more limited resources.1 This increased efficiency makes the power of fine-tuned large language models accessible to a broader range of users and applications.

### **Low-Rank Adaptation (LoRA)**

#### **Definition and Mechanism**

Low-Rank Adaptation (LoRA) has emerged as a widely adopted PEFT technique designed to address the resource demands of fine-tuning large language models. LoRA operates by freezing the original weights and parameters of the pre-trained model as they are. Then, on top of this original model, it strategically introduces trainable rank decomposition matrices into specific layers of the model, with a particular focus on the attention weights within the Transformer architecture.1 Instead of directly updating the entire weight matrix during the fine-tuning process, LoRA cleverly approximates the often substantial changes in weights (denoted as ΔW) using the product of two significantly smaller matrices, known as low-rank matrices (typically represented as A and B). This approximation is mathematically expressed as ΔW = BA.1 The final, adapted weight matrix (W') that the model uses is then calculated by adding this low-rank update to the original, frozen pre-trained weight matrix (W), resulting in the equation W' = W + BA.16 The effectiveness of LoRA hinges on the underlying principle that the inherent dimensionality of large language models is considerably lower than the apparent size of their weight tensors. Furthermore, the changes that need to be introduced during the fine-tuning process to adapt the model for a specific task can often be effectively captured within a lower-dimensional representation, hence the use of low-rank matrices.16 This approach allows for a substantial reduction in the total number of parameters that require training, leading to a significantly more efficient fine-tuning process.

#### **Advantages of LoRA**

LoRA offers several compelling advantages that have contributed to its widespread adoption. One of the most significant benefits is its substantial reduction in GPU memory consumption. By modifying only a small fraction of the total parameters, LoRA drastically lowers the VRAM usage compared to traditional full fine-tuning methods.1 This also leads to a faster fine-tuning process, as the computational overhead of training fewer parameters is significantly lower.1 Because the original, pre-trained weights of the model remain unchanged throughout the LoRA fine-tuning, it allows for easy adaptation of the same base model across multiple different tasks. Moreover, it enables efficient multi-task learning scenarios where multiple low-rank adapters can be trained for different tasks without interfering with each other.75 Another key advantage of LoRA is that it does not introduce any additional latency during inference. Once the training of the low-rank matrices is complete, their weights can be merged back into the original weight matrix of the pre-trained model. This results in a model with the adapted behavior without any increase in the computational time required to generate predictions.75 Overall, LoRA strikes a favorable balance between computational efficiency and the resulting model performance. It often achieves results that are comparable to those obtained through full fine-tuning, all while demanding considerably fewer computational resources.1 This makes it an exceptionally attractive technique for researchers and practitioners who may have limited access to high-performance computing infrastructure.

### **Quantized LoRA (QLoRA)**

#### **Definition and Mechanism**

Quantized Low-Rank Adaptation (QLoRA) builds upon the foundation of LoRA by incorporating quantization techniques to achieve an even greater level of efficiency. QLoRA combines the parameter-efficient benefits of low-rank adaptation with the memory-saving advantages of quantization.1 Specifically, QLoRA applies the process of gradient backpropagation through a pre-trained language model that has been frozen and quantized to 4-bit precision. This is done into the Low Rank Adapters (LoRA).31 To further enhance memory efficiency without compromising the model's performance on the target task, QLoRA employs several innovative approaches. These include the use of 4-bit NormalFloat (NF4) quantization, a data type that is theoretically optimal for representing normally distributed weights in a compact manner. Additionally, QLoRA utilizes double quantization to reduce the average memory footprint even further by quantizing the quantization constants themselves. Finally, it incorporates paged optimizers to effectively manage memory spikes that can occur during the training process.31 The combined effect of these techniques enables QLoRA to achieve remarkable memory savings. For instance, it makes it possible to fine-tune very large language models, such as those with 65 billion parameters, on a single consumer-grade GPU with limited memory capacity, all while preserving the task performance that would typically be associated with full 16-bit fine-tuning.1 By quantizing the base model to a lower precision and focusing the training on the much smaller low-rank adapters, QLoRA achieves a significant reduction in overall memory usage, thereby making large-scale fine-tuning much more accessible to a wider audience.

### **Other PEFT Techniques (Brief Overview)**

Beyond the widely used LoRA and its quantized variant QLoRA, the landscape of parameter-efficient fine-tuning offers a variety of other techniques, each with its own strengths and applications. **Adapter-based fine-tuning** involves inserting small, trainable neural network modules, known as adapters, into the layers of a pre-trained model. During fine-tuning, only the parameters of these newly added adapter modules are updated, while the original, often large, weights of the pre-trained model are kept frozen.7 This approach allows for efficient adaptation to specific tasks with minimal computational overhead. **Prefix tuning** takes a different approach by optimizing a sequence of continuous, task-specific vectors, referred to as the prefix. These prefixes are prepended to the input sequence that is fed into the language model. The key idea is that by training only these prefix vectors, the model can be effectively guided to perform new tasks without requiring any modifications to the original, pre-trained model's parameters.13 Similarly, **prompt tuning**, also often called soft prompting, focuses on optimizing continuous embeddings, known as soft prompts. These soft prompts are added to the input, and during training, only the parameters of these embeddings are updated, while the main model's weights remain frozen.3 Each of these PEFT techniques offers a unique set of trade-offs when considering the number of trainable parameters, the overall computational efficiency of the fine-tuning process, and the resulting performance of the adapted model on the specific target task.

## **Instruction Fine-Tuning**

### **Definition and Relationship with SFT**

Instruction fine-tuning is recognized as a specialized variant of the broader Supervised Fine-Tuning (SFT) methodology. Its primary objective is to significantly enhance a language model's inherent ability to accurately understand and effectively follow instructions provided by humans.3 While standard SFT involves training a model to execute a specific task using a dataset of labeled input-output pairs, instruction fine-tuning distinguishes itself by utilizing a particular type of labeled data known as "instruction-based" data. This data typically consists of pairs where each pair contains a natural language instruction that specifies a task, alongside the corresponding desired response or output from the model.3 The overarching goal of instruction fine-tuning extends beyond simply mastering a single, narrowly defined task. Instead, it aims to equip the model with the capability to interpret and execute a diverse range of instructions in an effective and generalizable manner.3 This refined approach to SFT explicitly focuses on aligning the model's behavior with the subtle nuances of human language instructions, ultimately leading to the development of more versatile and user-friendly AI assistants.3 By training the model on a wide spectrum of instructions and their anticipated outputs, it learns to generalize its understanding and execution abilities to new and previously unseen instructions, a crucial attribute for interactive and adaptable applications.

### **The Instruction Fine-Tuning Process**

The process of instruction fine-tuning typically involves training a pre-existing language model on a specially prepared labeled dataset. Each example within this dataset is structured to include an instruction that clearly specifies a task for the model to perform. Additionally, these examples may contain optional supplementary information or context that is relevant to the task at hand. Crucially, each example also includes the desired output or response that the model is expected to generate given the instruction and any provided context.3 During the training, the model analyzes its own generated predictions and compares them to the ground truth outputs provided in the dataset. Any discrepancies between the predicted and desired outputs are then used to adjust the model's internal parameters. This adjustment process allows the model to learn the underlying mapping between various types of instructions and their corresponding appropriate responses.3 In practice, to further enhance the model's performance and its ability to handle a broader array of inputs and tasks, the training process may involve the use of multiple different instructions within the dataset.45 The effectiveness of instruction fine-tuning is heavily reliant on the creation of a high-quality and diverse dataset that accurately reflects the desired behavior of the model in response to a wide range of instructions.3 The model's ultimate ability to follow instructions is directly influenced by the variety, clarity, and representativeness of the instructions it is exposed to during training. A well-constructed dataset enables the model to generalize its understanding and execution capabilities to new and previously unseen instructions with greater proficiency.

### **Benefits of Instruction Fine-Tuning**

Instruction fine-tuning offers several noteworthy advantages that make it a valuable technique for adapting large language models. One key benefit is the enhanced performance on specific tasks. Through training on instruction-based data, the model learns task-specific patterns and nuances that lead to improved accuracy and more relevant outputs.3 Perhaps more significantly, instruction fine-tuning significantly improves the model's general ability to follow a wide range of instructions. This enhanced instruction-following capability reduces the reliance on extensive prompt engineering or the need to provide numerous few-shot examples to guide the model toward the desired behavior.3 By being trained to approach problems in a step-by-step, logical manner, rather than simply producing an answer that appears linguistically coherent, models that undergo instruction tuning often develop improved reasoning skills.41 This results in model behavior that is more useful, predictable, and aligned with user expectations in practical applications.41 Furthermore, instruction fine-tuning has been shown to significantly boost a model's zero-shot learning capabilities. This means that the model can often perform well on tasks it has never explicitly encountered during training, relying solely on its ability to understand and execute the instructions provided at inference time.40

## **Categorization and Relationships of Fine-Tuning Techniques**

### **Hierarchical Organization**

The various techniques employed for fine-tuning LLMs can be effectively organized into a hierarchical structure that reflects their relationships and levels of specialization. At the highest level of this hierarchy is the broad category of **Fine-Tuning**. This encompasses any method that involves taking a pre-trained language model and further adapting it for a more specific purpose or task.2

A significant branch stemming from Fine-Tuning is **Supervised Fine-Tuning (SFT)**. This category specifically refers to fine-tuning methods that utilize labeled data, where each input in the training set is paired with a corresponding desired output, to guide the model's adaptation.2 Within SFT, several sub-techniques exist. **Full Fine-Tuning** represents one such approach, where the entire set of parameters within the pre-trained model is updated during the training process using the labeled data.7 Another important sub-category of SFT is **Instruction Fine-Tuning**. This specialized method employs instruction-based datasets, where the labeled examples consist of natural language instructions paired with the desired model responses. The goal here is to train the model to effectively understand and execute a wide variety of instructions.3

Furthermore, within the realm of SFT, **Parameter-Efficient Fine-Tuning (PEFT)** has emerged as a critical area of focus. PEFT techniques are designed to fine-tune large language models while significantly reducing the number of trainable parameters, leading to substantial gains in efficiency and reduced computational costs.3 Several important techniques fall under the PEFT umbrella. **Low-Rank Adaptation (LoRA)** is one such prominent method, which freezes the pre-trained model's weights and introduces low-rank matrices to adapt the model.1 **Quantized LoRA (QLoRA)** builds upon LoRA by incorporating quantization techniques to further reduce memory usage during fine-tuning.1 Other PEFT methods include adapter-based fine-tuning, prefix tuning, and prompt tuning, each offering unique ways to efficiently adapt LLMs.

Separately, **Reinforcement Learning from Human Feedback (RLHF)** represents a distinct approach to fine-tuning.2 It is noteworthy that RLHF often leverages a model that has already undergone SFT as its initial starting point.57

The relationship between these techniques is primarily hierarchical, with broader categories encompassing more specialized methods. SFT serves as a foundational technique upon which instruction tuning is built, and it is frequently used as a preliminary step before applying RLHF. PEFT techniques, such as LoRA and QLoRA, are often employed within the context of SFT to enhance the efficiency of the fine-tuning process. This hierarchical organization reflects the evolution and increasing specialization of fine-tuning methods, developed to address the diverse needs and challenges associated with adapting large language models for a wide range of applications.

### **Two-Level Selection**

The organization of LLM fine-tuning techniques is not best described as a simple two-level selection process. Instead, it is more accurately characterized as a multi-layered hierarchy. At each level of this hierarchy, the choice of which technique to employ is contingent upon a variety of factors. These include the specific objectives of the fine-tuning process, the computational resources that are available, and the particular characteristics of the task at hand. For example, if the primary goal is to align the model's responses with human preferences in a nuanced way, RLHF might be considered, possibly after an initial phase of instruction fine-tuning (a type of SFT). If computational resources are limited, PEFT techniques like QLoRA would be a more suitable choice, and these can be applied within an SFT framework. The selection is therefore a more complex decision-making process that involves navigating a multi-level structure based on the specific constraints and goals of the application.

### **Table for Categorization and Relationships**

| **Category** | **Technique** | **Description** | **Relationship to Other Techniques** |
| --- | --- | --- | --- |
| **Fine-Tuning** | Supervised Fine-Tuning (SFT) | Uses labeled input-output pairs to adapt a pre-trained model for a specific task. | Foundation for instruction tuning; often a precursor to RLHF; PEFT techniques can be applied within SFT. |
|  | Reinforcement Learning from Human Feedback (RLHF) | Optimizes a language model using human preferences as a reward signal, often following an initial SFT phase. | Often uses an SFT model as a starting point; can incorporate PEFT techniques. |
| **Parameter-Efficient Fine-Tuning (PEFT)** | Low-Rank Adaptation (LoRA) | Freezes pre-trained weights and introduces low-rank matrices for training, reducing the number of trainable parameters. | Can be used within SFT and potentially RLHF; basis for QLoRA. |
|  | Quantized LoRA (QLoRA) | Extends LoRA by incorporating quantization techniques (e.g., 4-bit NormalFloat) to further reduce memory usage. | A memory-efficient variant of LoRA; used within SFT. |
|  | Adapter-Based Fine-Tuning | Inserts small, trainable adapter modules into the pre-trained model's layers while keeping original weights frozen. | A PEFT technique that can be used in various fine-tuning scenarios. |
|  | Prefix Tuning | Optimizes a continuous, task-specific prefix prepended to the input, keeping the main model frozen. | A PEFT technique focused on natural language generation tasks. |
|  | Prompt Tuning (Soft Prompting) | Optimizes continuous embeddings (soft prompts) added to the input, while the model's weights remain frozen. | A PEFT technique often used for adapting models to specific tasks. |
| **SFT Sub-category** | Instruction Fine-Tuning | A form of SFT that uses instruction-based datasets (instruction-response pairs) to improve the model's ability to follow instructions. | A specialized type of SFT; can benefit from PEFT techniques. |
| **Other Techniques** | Prompt Engineering | Crafting specific prompts to elicit desired responses from LLMs without modifying model weights. | Used in conjunction with various fine-tuning techniques or as an alternative. |
|  | Distillation | Training a smaller, more efficient "student" model to mimic the behavior of a larger "teacher" model. | Can be used to create smaller versions of fine-tuned models. |
|  | Retrieval-Augmented Generation (RAG) | Enhances LLMs by retrieving relevant information from external sources and incorporating it into the generation process. | Can be used to improve the factual accuracy of fine-tuned models or as an alternative to fine-tuning for knowledge integration. |
|  | Transfer Learning | The general concept of leveraging knowledge from a pre-trained model for a new task, encompassing fine-tuning. | Underlies all fine-tuning techniques. |
|  | Few-Shot Learning | Enabling models to adapt to new tasks with only a limited number of examples provided in the prompt. | Can be used in conjunction with fine-tuning to guide the model's learning with specific examples. |
|  | Zero-Shot Learning | Aiming to perform tasks without any specific training examples, relying on the model's pre-existing knowledge and instructions in the prompt. | Often a benchmark against which fine-tuned models are compared. |

## **Other Important Fine-Tuning Techniques**

Beyond the core techniques of SFT, RLHF, LoRA, QLoRA, and instruction fine-tuning, several other important methodologies contribute to the effective adaptation of LLMs. **Prompt engineering** is a critical skill that involves carefully designing input queries or instructions (prompts) to elicit more accurate and relevant outputs from LLMs without altering the underlying model parameters.3 This technique is often used in conjunction with fine-tuning to further refine the model's behavior. **Distillation** is another valuable approach where a smaller, more computationally efficient "student" model is trained to replicate the behavior and knowledge of a larger, often more accurate, "teacher" model.8 This is particularly useful for deploying LLMs in resource-constrained environments. **Retrieval-Augmented Generation (RAG)** is a framework that enhances the capabilities of LLMs by allowing them to retrieve relevant information from external knowledge bases and incorporate this information into their response generation process.13 This is crucial for tasks requiring up-to-date or domain-specific knowledge not present in the model's original training data.

The overarching concept of **transfer learning** is fundamental to all fine-tuning techniques, as it involves leveraging the knowledge and representations learned by a model during its initial pre-training phase and applying them to a new, often related, task.7 Within this context, **few-shot learning** and **zero-shot learning** represent paradigms where the model is expected to perform new tasks with very limited or no task-specific training examples, respectively, relying on the model's general knowledge and the instructions provided in the prompt.5 Additionally, techniques like multi-task learning (training a model on multiple related tasks simultaneously), sequential fine-tuning (adapting a model through a series of stages for different tasks), and domain-specific fine-tuning (tailoring a model to understand and generate text specific to a particular domain) address more specialized requirements in LLM adaptation.

## **Conclusion**

In summary, the field of LLM fine-tuning offers a diverse and evolving set of techniques to adapt pre-trained models for specific applications and to align their behavior with desired outcomes. Supervised Fine-Tuning (SFT) provides a foundational approach by leveraging labeled data to guide the model's learning. Reinforcement Learning from Human Feedback (RLHF) builds upon this by incorporating human preferences to further refine the model's outputs and ensure alignment with human values. Parameter-Efficient Fine-Tuning (PEFT) techniques, including Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA), offer efficient strategies for fine-tuning large models with significantly reduced computational resources. Instruction fine-tuning, a specialized form of SFT, focuses on enhancing the model's ability to understand and execute natural language instructions effectively. These techniques are not isolated but rather organized in a hierarchical structure, with SFT often serving as a basis for many others, and RLHF frequently following an initial SFT phase. The selection of the most appropriate fine-tuning technique is a nuanced decision that depends on the specific requirements of the application, the availability of relevant data and computational resources, and the desired level of alignment with human preferences. The continuous advancements in this field promise even more sophisticated and efficient methods for harnessing the full potential of large language models in the future.

#### Works cited

1. What is LoRA (Low-Rank Adaption)? - IBM, accessed May 3, 2025, <https://www.ibm.com/think/topics/lora>
2. Supervised Fine-Tuning vs. RLHF: How to Choose the Right Approach to Train Your LLM, accessed May 3, 2025, <https://www.invisible.co/blog/supervised-fine-tuning-vs-rlhf-how-to-choose-the-right-approach-to-train-your-llm>
3. Fine-Tuning LLMs: Top 6 Methods, Challenges & Best Practices - Acorn Labs, accessed May 3, 2025, <https://www.acorn.io/resources/learning-center/fine-tuning-llm/>
4. How to Train an LLM Using Fine-Tuning? Best Practices for Businesses - Botscrew, accessed May 3, 2025, <https://botscrew.com/blog/how-to-train-an-llm-using-fine-tuning/>
5. How to fine-tune a large language model (LLM) | Generative-AI – Weights & Biases - Wandb, accessed May 3, 2025, <https://wandb.ai/byyoung3/Generative-AI/reports/How-to-fine-tune-a-large-language-model-LLM---VmlldzoxMDU2NTg4Mw>
6. Top Tools and Techniques for LLM Fine-Tuning: A Comprehensive Guide - ADaSci, accessed May 3, 2025, <https://adasci.org/top-tools-and-techniques-for-llm-fine-tuning-a-comprehensive-guide/>
7. Finetuning in large language models - Oracle Blogs, accessed May 3, 2025, <https://blogs.oracle.com/ai-and-datascience/post/finetuning-in-large-language-models>
8. LLMs: Fine-tuning, distillation, and prompt engineering | Machine Learning, accessed May 3, 2025, <https://developers.google.com/machine-learning/crash-course/llm/tuning>
9. The Ultimate Guide to LLM Fine Tuning: Best Practices & Tools - Lakera AI, accessed May 3, 2025, <https://www.lakera.ai/blog/llm-fine-tuning-guide>
10. Fine-tuning large language models (LLMs) in 2025 - SuperAnnotate, accessed May 3, 2025, <https://www.superannotate.com/blog/llm-fine-tuning>
11. Fine-Tuning Large Language Models - Analytics Vidhya, accessed May 3, 2025, <https://www.analyticsvidhya.com/blog/2023/08/fine-tuning-large-language-models/>
12. LLM Fine-Tuning—Overview with Code Example - Nexla, accessed May 3, 2025, <https://nexla.com/enterprise-ai/llm-fine-tuning/>
13. Comprehensive Guide to LLM Fine-Tuning - hiberus blog - Exploring Technology, AI, and Digital Experiences, accessed May 3, 2025, <https://www.hiberus.com/en/blog/guide-to-llm-fine-tuning/>
14. LLM Fine-Tuning: What It Is, Common Techniques, And More - Multimodal.dev, accessed May 3, 2025, <https://www.multimodal.dev/post/llm-fine-tuning>
15. LLM Fine-Tuning: Use Cases, Best Practices, and Top 8 PEFT Methods - Kolena, accessed May 3, 2025, <https://www.kolena.com/guides/llm-fine-tuning-use-cases-best-practices-and-top-8-peft-methods/>
16. What LoRA Adapters for LLM Fine Tuning - Datawizz.ai, accessed May 3, 2025, <https://datawizz.ai/blog/what-are-low-rank-(lora)-adapters>
17. 5 Reasons Why LoRA Adapters are the Future of Fine-tuning - Predibase, accessed May 3, 2025, <https://predibase.com/blog/5-reasons-why-lora-adapters-are-the-future-of-fine-tuning>
18. Fine-Tuning LLMs: A Guide With Examples - DataCamp, accessed May 3, 2025, <https://www.datacamp.com/tutorial/fine-tuning-large-language-models>
19. What is Fine-Tuning LLM? Methods & Step-by-Step Guide in 2025 - Turing, accessed May 3, 2025, <https://www.turing.com/resources/finetuning-large-language-models>
20. Fine-Tuning LLMs: LoRA or Full-Parameter? An in-depth Analysis with Llama 2 - Anyscale, accessed May 3, 2025, <https://www.anyscale.com/blog/fine-tuning-llms-lora-or-full-parameter-an-in-depth-analysis-with-llama-2>
21. Unlocking LLM Training: Transfer Learning vs Fine-tuning Explained - DX Talks, accessed May 3, 2025, <https://www.dxtalks.com/blog/news-2/unlocking-llm-training-transfer-learning-vs-fine-tuning-explained-544>
22. What is Supervised Fine-Tuning? Overview and Techniques - Sapien, accessed May 3, 2025, <https://www.sapien.io/blog/what-is-supervised-fine-tuning-overview-and-techniques>
23. Supervised Fine-Tuning (SFT) | LLM Knowledge Base - Promptmetheus, accessed May 3, 2025, <https://promptmetheus.com/resources/llm-knowledge-base/supervised-fine-tuning-sft>
24. Difference between Fine-Tuning, Supervised fine-tuning (SFT) and Instruction Fine-Tuning, accessed May 3, 2025, <https://www.geeksforgeeks.org/difference-between-fine-tuning-supervised-fine-tuning-sft-and-instruction-fine-tuning/>
25. Supervised Fine-Tuning: How to Customize Your LLM? - Toloka, accessed May 3, 2025, <https://toloka.ai/blog/supervised-fine-tuning/>
26. Supervised Fine-tuning: customizing LLMs - Mantis NLP, accessed May 3, 2025, <https://mantisnlp.com/blog/supervised-fine-tuning-customizing-llms/>
27. Supervised fine-tuning (SFT) - Klu.ai, accessed May 3, 2025, <https://klu.ai/glossary/supervised-fine-tuning>
28. Supervised Fine-tuning Trainer - Hugging Face, accessed May 3, 2025, <https://huggingface.co/docs/trl/sft_trainer>
29. Fine-Tuning LLMs: Supervised Fine-Tuning and Reward Modelling, accessed May 3, 2025, <https://huggingface.co/blog/rishiraj/finetune-llms>
30. Supervised Fine Tuning for Gemini LLM | Google Cloud Blog, accessed May 3, 2025, <https://cloud.google.com/blog/products/ai-machine-learning/supervised-fine-tuning-for-gemini-llm>
31. A Comparison of LLM Fine-tuning Methods and Evaluation Metrics with Travel Chatbot Use Case - arXiv, accessed May 3, 2025, <https://arxiv.org/html/2408.03562v1>
32. What is supervised fine-tuning in LLMs? Unveiling the process - Nebius, accessed May 3, 2025, <https://nebius.com/blog/posts/fine-tuning/supervised-fine-tuning>
33. Supervised Fine-Tuning - Hugging Face NLP Course, accessed May 3, 2025, <https://huggingface.co/learn/nlp-course/chapter11/3>
34. 4. Fine-Tuning LLMs — GenAI 0.1 documentation - Jie Ding, accessed May 3, 2025, <https://genai-course.jding.org/finetuning/index.html>
35. Customizing LLMs: When to Choose LoRA or Full Fine-Tuning - Gradient Flow, accessed May 3, 2025, <https://gradientflow.com/lora-or-full-fine-tuning/>
36. Hierarchical Fine-Tuning: Staged Approach to Domain-Specific LLMs - Algos, accessed May 3, 2025, <https://algos-ai.com/hierarchical-fine-tuning/>
37. LLM Fine Tuning Methods: Standard & Enhanced - Label Your Data, accessed May 3, 2025, <https://labelyourdata.com/articles/llm-fine-tuning/llm-fine-tuning-methods>
38. Understanding Parameter-Efficient Finetuning of Large Language Models: From Prefix Tuning to LLaMA-Adapters - Lightning AI, accessed May 3, 2025, <https://lightning.ai/pages/community/article/understanding-llama-adapters/>
39. ubiai.tools, accessed May 3, 2025, <https://ubiai.tools/what-is-instruction-fine-tuning-and-why-is-it-important-2/#:~:text=Instruction%20tuning%20is%20one%20of,for%20leveraging%20its%20full%20potential.>
40. Review: Fine-tuned language models are zero-shot learners - Paperspace Blog, accessed May 3, 2025, <https://blog.paperspace.com/instruction-tuning/>
41. What Is Instruction Tuning? | IBM, accessed May 3, 2025, <https://www.ibm.com/think/topics/instruction-tuning>
42. What is the difference between pre-training, fine-tuning, and instruct-tuning exactly? - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/learnmachinelearning/comments/19f04y3/what_is_the_difference_between_pretraining/>
43. Instruction Tuning: What is fine-tuning? - DataScientest, accessed May 3, 2025, <https://datascientest.com/en/instruction-tuning-what-is-fine-tuning>
44. What is the difference between pre-training, fine-tuning, and instruct-tuning exactly? - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LanguageTechnology/comments/19f05hr/what_is_the_difference_between_pretraining/>
45. Instruction Fine-tuning in LLM Explained - YouTube, accessed May 3, 2025, <https://www.youtube.com/watch?v=GX2XxvQSpeU>
46. Supervised Fine-Tuning - Hugging Face LLM Course, accessed May 3, 2025, <https://huggingface.co/learn/llm-course/chapter11/1>
47. Transfer Learning vs. Fine Tuning LLMs: Key Differences - 101 Blockchains, accessed May 3, 2025, <https://101blockchains.com/transfer-learning-vs-fine-tuning/>
48. Transfer learning and fine-tuning | TensorFlow Core, accessed May 3, 2025, <https://www.tensorflow.org/tutorials/images/transfer_learning>
49. Transfer Learning for Finetuning Large Language Models - arXiv, accessed May 3, 2025, <https://arxiv.org/html/2411.01195v1>
50. Transfer learning or fine tuning : r/learnmachinelearning - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/learnmachinelearning/comments/1axtl44/transfer_learning_or_fine_tuning/>
51. Is LLM fine tuning the same as transfer learning? - Community Proposals - Codidact, accessed May 3, 2025, <https://proposals.codidact.com/posts/290005>
52. How to Train LLMs for Few-Shot and Zero-Shot Learning? - DEV Community, accessed May 3, 2025, <https://dev.to/hakeem/how-to-train-llms-for-few-shot-and-zero-shot-learning-2c52>
53. Leveraging model distillation to fine-tune a model | OpenAI Cookbook, accessed May 3, 2025, <https://cookbook.openai.com/examples/leveraging_model_distillation_to_fine-tune_a_model>
54. Distilling step-by-step: Outperforming larger language models with less training, accessed May 3, 2025, <https://research.google/blog/distilling-step-by-step-outperforming-larger-language-models-with-less-training-data-and-smaller-model-sizes/>
55. [2305.02301] Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2305.02301>
56. [R] Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/MachineLearning/comments/1381gd3/r_distilling_stepbystep_outperforming_larger/>
57. What is RLHF? - Reinforcement Learning from Human Feedback ..., accessed May 3, 2025, <https://aws.amazon.com/what-is/reinforcement-learning-from-human-feedback/>
58. What Is Reinforcement Learning From Human Feedback (RLHF)? - IBM, accessed May 3, 2025, <https://www.ibm.com/think/topics/rlhf>
59. What is Reinforcement Learning from Human Feedback (RLHF)? - Alation, accessed May 3, 2025, <https://www.alation.com/blog/what-is-rlhf-reinforcement-learning-human-feedback/>
60. Reinforcement learning from human feedback - Wikipedia, accessed May 3, 2025, <https://en.wikipedia.org/wiki/Reinforcement_learning_from_human_feedback>
61. Exploring Reinforcement Learning from Human Feedback (RLHF) - Kili Technology, accessed May 3, 2025, <https://kili-technology.com/large-language-models-llms/exploring-reinforcement-learning-from-human-feedback-rlhf-a-comprehensive-guide>
62. RLHF: Reinforcement Learning from Human Feedback - Chip Huyen, accessed May 3, 2025, <https://huyenchip.com/2023/05/02/rlhf.html>
63. Reinforcement Learning from Human Feedback (RLHF) - Niklas Heidloff, accessed May 3, 2025, <https://heidloff.net/article/rlhf/>
64. Illustrating Reinforcement Learning from Human Feedback (RLHF) - Hugging Face, accessed May 3, 2025, <https://huggingface.co/blog/rlhf>
65. Fine-Tuning LLMs: RLHF, LoRA, and Instruction Tuning - Synthesis AI, accessed May 3, 2025, <https://synthesis.ai/2024/08/13/fine-tuning-llms-rlhf-lora-and-instruction-tuning/>
66. Reinforcement Learning from Human Feedback (RLHF) - OpenRLHF's documentation!, accessed May 3, 2025, <https://openrlhf.readthedocs.io/en/latest/rl.html>
67. The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities (Version 1.0) - arXiv, accessed May 3, 2025, <https://arxiv.org/html/2408.13296v1>
68. [2305.14314] QLoRA: Efficient Finetuning of Quantized LLMs - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2305.14314>
69. Webinar Recap: Techniques in Fine-Tuning LLMs - Kili Technology, accessed May 3, 2025, <https://kili-technology.com/large-language-models-llms/webinar-recap-techniques-in-fine-tuning-llms>
70. What is Reinforcement Learning from Human Feedback? - DataCamp, accessed May 3, 2025, <https://www.datacamp.com/blog/what-is-reinforcement-learning-from-human-feedback>
71. A Deep Dive into the Trade-Offs of Parameter-Efficient Preference Alignment Techniques, accessed May 3, 2025, <https://arxiv.org/html/2406.04879v1>
72. [2502.03884] Rank Also Matters: Hierarchical Configuration for Mixture of Adapter Experts in LLM Fine-Tuning - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2502.03884>
73. [2304.01933] LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2304.01933>
74. AGI-Edgerunners/LLM-Adapters: Code for our EMNLP 2023 Paper: "LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models" - GitHub, accessed May 3, 2025, <https://github.com/AGI-Edgerunners/LLM-Adapters>
75. Fine-Tuning of Large Language Models with LoRA and QLoRA - Analytics Vidhya, accessed May 3, 2025, <https://www.analyticsvidhya.com/blog/2023/08/lora-and-qlora/>
76. Low-Rank Adaptation (LoRA): Revolutionizing AI Fine-Tuning - Coralogix, accessed May 3, 2025, <https://coralogix.com/ai-blog/low-rank-adaptation-a-closer-look-at-lora/>
77. What is QLoRA? | QLoRA – Weights & Biases - Wandb, accessed May 3, 2025, <https://wandb.ai/sauravmaheshkar/QLoRA/reports/What-is-QLoRA---Vmlldzo2MTI2OTc5>
78. Supervised Fine-Tuning - Hugging Face NLP Course, accessed May 3, 2025, <https://huggingface.co/learn/nlp-course/chapter11/1>
79. LoRA (Low-Rank Adaptation) - Hugging Face NLP Course, accessed May 3, 2025, <https://huggingface.co/learn/nlp-course/chapter11/4>
80. What is Low Rank Adaptation (LoRA)? - GeeksforGeeks, accessed May 3, 2025, <https://www.geeksforgeeks.org/what-is-low-rank-adaptation-lora/>
81. Mastering Low-Rank Adaptation (LoRA): Enhancing Large Language Models for Efficient Adaptation | DataCamp, accessed May 3, 2025, <https://www.datacamp.com/tutorial/mastering-low-rank-adaptation-lora-enhancing-large-language-models-for-efficient-adaptation>
82. Understanding LoRA - Low Rank Adaptation For Finetuning Large Models, accessed May 3, 2025, <https://towardsdatascience.com/understanding-lora-low-rank-adaptation-for-finetuning-large-models-936bce1a07c6/>
83. Low Rank Adaptation: A Technical deep dive - ML6, accessed May 3, 2025, <https://www.ml6.eu/blogpost/low-rank-adaptation-a-technical-deep-dive>
84. Low-Rank Adaptation (LoRA) And Quantization Unlocking AI Efficiency, accessed May 3, 2025, <https://sapientai.io/low-rank-adaptation-lora-and-quantization/>
85. Fine-Tuning Llama 3 with LoRA: Step-by-Step Guide - Neptune.ai, accessed May 3, 2025, <https://neptune.ai/blog/fine-tuning-llama-3-with-lora>
86. Using LoRA for efficient fine-tuning: Fundamental principles - ROCm Blogs - AMD, accessed May 3, 2025, <https://rocm.blogs.amd.com/artificial-intelligence/lora-fundamentals/README.html>
87. Finetuning LLMs with LoRA and QLoRA: Insights from Hundreds of Experiments, accessed May 3, 2025, <https://lightning.ai/pages/community/lora-insights/>
88. Finetuning with LoRA and variants - Prem AI Blog, accessed May 3, 2025, <https://blog.premai.io/lora/>
89. Efficient Fine-Tuning with LoRA for LLMs | Databricks Blog, accessed May 3, 2025, <https://www.databricks.com/blog/efficient-fine-tuning-lora-guide-llms>
90. LoRA Fine-tuning & Hyperparameters Explained (in Plain English) | Entry Point AI, accessed May 3, 2025, <https://www.entrypointai.com/blog/lora-fine-tuning/>
91. LoRA selected as the fine-tuning technique added to MLPerf Training v4.0 - MLCommons, accessed May 3, 2025, <https://mlcommons.org/2024/06/lora-fine-tuning-mlperf-training-v4-0/>
92. Is LoRA Fine-Tuning Sometimes Less Effective Than Full Fine-Tuning of Smaller Models?, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1eg0cap/is_lora_finetuning_sometimes_less_effective_than/>
93. LongLoRA: Efficient long-context fine-tuning, supervised fine-tuning : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/16p1k2e/longlora_efficient_longcontext_finetuning/>
94. Seeking Clarification on LoRA, Adapters, and Prefix Tuning in LLMs : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/17mrd3y/seeking_clarification_on_lora_adapters_and_prefix/>
95. Adapters: the game changer for fine-tuning - Geoffrey Angus - The Data Scientist Show #084, accessed May 3, 2025, <https://www.youtube.com/watch?v=BQYD9HivFLY>
96. How to Fine-tune LLMs with Unsloth: Complete Guide - YouTube, accessed May 3, 2025, <https://www.youtube.com/watch?v=Lt7KrFMcCis>
97. Quantized low-rank adaptation (QLoRA) fine tuning - IBM, accessed May 3, 2025, <https://www.ibm.com/docs/en/watsonx/w-and-w/2.1.x?topic=tuning-qlora-fine>
98. What is QLoRA (Quantized Low-Rank Adapter)? - GeeksforGeeks, accessed May 3, 2025, <https://www.geeksforgeeks.org/what-is-qlora-quantized-low-rank-adapter/>
99. Quantized LoRA: Fine-Tuning Large Language Models Made Easy - Athina AI Hub, accessed May 3, 2025, <https://hub.athina.ai/blogs/quantized-lora-fine-tuning-large-language-models-with-ease/>
100. QDyLoRA: Quantized Dynamic Low-Rank Adaptation for Efficient Large Language Model Tuning - arXiv, accessed May 3, 2025, <https://arxiv.org/html/2402.10462v1>
101. Paper Review: QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models - Andrey Lukyanenko, accessed May 3, 2025, <https://andlukyane.com/blog/paper-review-qalora>
102. How to fine-tune open LLMs in 2025 with Hugging Face - Philschmid, accessed May 3, 2025, <https://www.philschmid.de/fine-tune-llms-in-2025>
103. @philschmid on Hugging Face: "What's the best way to fine-tune open LLMs in 2024? Look no further! I am…", accessed May 3, 2025, <https://huggingface.co/posts/philschmid/542513204804942>
104. I will do the fine-tuning for you, or here's my DIY guide : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/18n2bwu/i_will_do_the_finetuning_for_you_or_heres_my_diy/>
105. Memory-efficient Fine-tuning with with QLoRA - Niklas Heidloff, accessed May 3, 2025, <https://heidloff.net/article/qlora/>
106. QLORA: Efficient Finetuning of Quantized LLMs - arXiv, accessed May 3, 2025, <https://arxiv.org/pdf/2305.14314>
107. Do we need a standardized format for LLM adapters? (like GGUF) : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1dju7dp/do_we_need_a_standardized_format_for_llm_adapters/>
108. Prefix-Tuning: Optimizing Continuous Prompts for Generation, accessed May 3, 2025, <https://learnprompting.org/docs/trainable/prefix-tuning>
109. Prefix tuning - Hugging Face, accessed May 3, 2025, <https://huggingface.co/docs/peft/package_reference/prefix_tuning>
110. Prefix Tuning vs. Fine-Tuning and other PEFT methods - Toloka, accessed May 3, 2025, <https://toloka.ai/blog/prefix-tuning-vs-fine-tuning/>
111. Prompt Tuning and Prefix Tuning, accessed May 3, 2025, <https://ericwiener.github.io/ai-notes/AI-Notes/Large-Language-Models/Prompt-Tuning-and-Prefix-Tuning>
112. Prefix Tuning for Large Language Model (LLM) Explained - YouTube, accessed May 3, 2025, <https://www.youtube.com/watch?v=kg0ODV8AogI>
113. L-TUNING: Synchronized Label Tuning for Prompt and Prefix in LLMs - arXiv, accessed May 3, 2025, <https://arxiv.org/html/2402.01643v1>
114. [2101.00190] Prefix-Tuning: Optimizing Continuous Prompts for Generation - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2101.00190>
115. When Do Prompting and Prefix-Tuning Work? A Theory of Capabilities and Limitations, accessed May 3, 2025, <https://openreview.net/forum?id=JewzobRhay>
116. How prompt tuning works - IBM Developer, accessed May 3, 2025, <https://developer.ibm.com/articles/awb-how-prompt-tuning-works/>
117. Understanding Prompt Tuning: Enhance Your Language Models with Precision - DataCamp, accessed May 3, 2025, <https://www.datacamp.com/tutorial/understanding-prompt-tuning>
118. Prompt Tuning vs. Fine-Tuning—Differences, Best Practices, and Use Cases | Nexla, accessed May 3, 2025, <https://nexla.com/ai-infrastructure/prompt-tuning-vs-fine-tuning/>
119. Prompt Tuning with Soft Prompts, accessed May 3, 2025, <https://learnprompting.org/docs/trainable/soft_prompting>
120. Want Better AI Outputs? Use Prompt Tuning. | Built In, accessed May 3, 2025, <https://builtin.com/articles/ai-outputs-prompt-tuning>
121. LLMs Optimization Techniques: Prompt Tuning and Prompt Engineering - Toloka, accessed May 3, 2025, <https://toloka.ai/blog/llms-optimization-techniques/>
122. What is Prompt Engineering vs Prompt Tuning vs Fine tuning in LLM? : r/LLMDevs - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LLMDevs/comments/16ngqem/what_is_prompt_engineering_vs_prompt_tuning_vs/>
123. An Introduction to Large Language Models: Prompt Engineering and P-Tuning, accessed May 3, 2025, <https://developer.nvidia.com/blog/an-introduction-to-large-language-models-prompt-engineering-and-p-tuning/>
124. Using LLMs for Prompt-tuning? : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1c0ir44/using_llms_for_prompttuning/>
125. My experience on starting with fine tuning LLMs with custom data : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/14vnfh2/my_experience_on_starting_with_fine_tuning_llms/>
126. Build an LLM from Scratch 6: Finetuning for Classification - YouTube, accessed May 3, 2025, <https://www.youtube.com/watch?v=5PFXJYme4ik>
127. A comprehensive overview of everything I know about fine-tuning. : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1ilkamr/a_comprehensive_overview_of_everything_i_know/>
128. Comparing LLM fine-tuning methods - SignalFire, accessed May 3, 2025, <https://www.signalfire.com/blog/comparing-llm-fine-tuning-methods>
129. How do you actually fine-tune a LLM on your own data? : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1fm59kg/how_do_you_actually_finetune_a_llm_on_your_own/>
130. labelbox.com, accessed May 3, 2025, <https://labelbox.com/guides/zero-shot-learning-few-shot-learning-fine-tuning/#:~:text=Few%2Dshot%20learning%20%E2%80%94%20a%20technique,as%20a%20new%20model%20checkpoint>
131. Zero-Shot Learning vs. Few-Shot Learning vs. Fine-Tuning - Labelbox, accessed May 3, 2025, <https://labelbox.com/guides/zero-shot-learning-few-shot-learning-fine-tuning/>
132. Fine-tuning vs. Few-shot Learning: How to Customize a Large Language Model for Beginners - The Augmented Advantage, accessed May 3, 2025, <https://blog.tobiaszwingmann.com/p/fine-tuning-vs-few-shot-learning-how-to-customize-a-large-language-model-for-beginners>
133. [2502.02715] An Analysis of LLM Fine-Tuning and Few-Shot Learning for Flaky Test Detection and Classification - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2502.02715>
134. Zero-Shot and Few-Shot Learning with LLMs - Neptune.ai, accessed May 3, 2025, <https://neptune.ai/blog/zero-shot-and-few-shot-learning-with-llms>
135. Few-Shot VS Finetuning? : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/14u5peb/fewshot_vs_finetuning/>
136. What are the differences between fine tuning and few shot learning? - Stack Overflow, accessed May 3, 2025, <https://stackoverflow.com/questions/72611335/what-are-the-differences-between-fine-tuning-and-few-shot-learning>
137. Few-shot prompt engineering and fine-tuning for LLMs in Amazon Bedrock - AWS, accessed May 3, 2025, <https://aws.amazon.com/blogs/machine-learning/few-shot-prompt-engineering-and-fine-tuning-for-llms-in-amazon-bedrock/>
138. Fine tuning vs few shot training & caching a trained model : r/LLMDevs - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LLMDevs/comments/1dnmhtw/fine_tuning_vs_few_shot_training_caching_a/>
139. LLM distillation demystified: a complete guide | Snorkel AI, accessed May 3, 2025, <https://snorkel.ai/blog/llm-distillation-demystified-a-complete-guide/>
140. Distilling vs Fine Tunning : r/LLMDevs - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LLMDevs/comments/1irpz19/distilling_vs_fine_tunning/>
141. Distilling a faster and smaller custom LLM using Google Gemini - Labelbox, accessed May 3, 2025, <https://labelbox.com/guides/end-to-end-workflow-for-knowledge-distillation-with-nlp/>
142. Google Open-Sources AI Fine-Tuning Method Distilling Step-by-Step - InfoQ, accessed May 3, 2025, <https://www.infoq.com/news/2023/10/google-distillation/>
143. Model Distillation for Large Language Models | Niklas Heidloff, accessed May 3, 2025, <https://heidloff.net/article/model-distillation-large-language-models/>
144. Fine-Tuning with Reinforcement Learning from Human Feedback (RLHF) Training Course, accessed May 3, 2025, <https://www.nobleprog.com/cc/ftrlhf>
145. Has there been any research into using parameter-efficient training like LoRA and QLoRA during RL for pretrained models? I have some big models I want to run RL on and would prefer to avoid buying a zillion GPUs. : r/reinforcementlearning - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/reinforcementlearning/comments/18oe4t2/has_there_been_any_research_into_using/>
146. Fine-tuning LLMs on Human Feedback (RLHF + DPO) - YouTube, accessed May 3, 2025, <https://www.youtube.com/watch?v=bbVoDXoPrPM>
147. [D] What the best way to resolve QLORA tuned model forgetting ? : r/MachineLearning, accessed May 3, 2025, <https://www.reddit.com/r/MachineLearning/comments/1cgdndx/d_what_the_best_way_to_resolve_qlora_tuned_model/>
148. RAG vs. Fine-tuning and more | Google Cloud Blog, accessed May 3, 2025, <https://cloud.google.com/blog/products/ai-machine-learning/to-tune-or-not-to-tune-a-guide-to-leveraging-your-data-with-llms>
149. Make with MakerSuite Part 2: Tuning LLMs - Google for Developers Blog, accessed May 3, 2025, <https://developers.googleblog.com/en/make-with-makersuite-part-2-tuning-llms/>
150. Optimizing LLM Accuracy - OpenAI API, accessed May 3, 2025, <https://platform.openai.com/docs/guides/optimizing-llm-accuracy>
151. Prompt Engineering Guide, accessed May 3, 2025, <https://www.promptingguide.ai/>
152. Prompt engineering: A guide to improving LLM performance - CircleCI, accessed May 3, 2025, <https://circleci.com/blog/prompt-engineering/>
153. Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications - Amazon.com, accessed May 3, 2025, <https://www.amazon.com/Prompt-Engineering-LLMs-Model-Based-Applications/dp/1098156153>
154. Prompt Engineering of LLM Prompt Engineering : r/PromptEngineering - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/PromptEngineering/comments/1hv1ni9/prompt_engineering_of_llm_prompt_engineering/>
155. Prompt Engineering for LLMs[Book] - O'Reilly Media, accessed May 3, 2025, <https://www.oreilly.com/library/view/prompt-engineering-for/9781098156145/>
156. Prompt Engineering for AI Guide | Google Cloud, accessed May 3, 2025, <https://cloud.google.com/discover/what-is-prompt-engineering>
157. Prompt Engineering for LLMs, accessed May 3, 2025, <https://zncd.ir/wp-content/uploads/2025/01/John-Berryman-Albert-Ziegler-Prompt-Engineering-for-LLMs_-The-Art-and-Science-of-Building-Large-Language-Model-Based-Applications-2024-OReilly-Media-libgen.li_.pdf>
158. Prompt Engineering for Generative AI | Machine Learning - Google for Developers, accessed May 3, 2025, <https://developers.google.com/machine-learning/resources/prompt-eng>
159. Prompt engineering essentials: Getting better results from LLMs | Tutorial - YouTube, accessed May 3, 2025, <https://www.youtube.com/watch?v=LAF-lACf2QY>
160. A developer's guide to prompt engineering and LLMs - The GitHub Blog, accessed May 3, 2025, <https://github.blog/ai-and-ml/generative-ai/prompt-engineering-guide-generative-ai-llms/>
161. Full fine tuning llama 7B without LORA : r/LocalLLaMA - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1af93xl/full_fine_tuning_llama_7b_without_lora/>
162. Fine-tuning LLM with limited documents and hierarchy - Data Science Stack Exchange, accessed May 3, 2025, <https://datascience.stackexchange.com/questions/123456/fine-tuning-llm-with-limited-documents-and-hierarchy>
163. What is Retrieval-Augmented Generation (RAG)? | Google Cloud, accessed May 3, 2025, <https://cloud.google.com/use-cases/retrieval-augmented-generation>
164. What is RAG? - Retrieval-Augmented Generation AI Explained - AWS, accessed May 3, 2025, <https://aws.amazon.com/what-is/retrieval-augmented-generation/>
165. What Is Retrieval-Augmented Generation, aka RAG? - NVIDIA, accessed May 3, 2025, <https://resources.nvidia.com/en-us-ai-large-language-models/retrieval-augmented-generation-explainer?lx=176kQp>
166. Retrieval-augmented generation - Wikipedia, accessed May 3, 2025, <https://en.wikipedia.org/wiki/Retrieval-augmented_generation>
167. Retrieval Augmented Generation (RAG) for LLMs - Prompt Engineering Guide, accessed May 3, 2025, <https://www.promptingguide.ai/research/rag>
168. What is Retrieval Augmented Generation (RAG)? - Databricks, accessed May 3, 2025, <https://www.databricks.com/glossary/retrieval-augmented-generation-rag>
169. What is retrieval-augmented generation (RAG)? - IBM Research, accessed May 3, 2025, <https://research.ibm.com/blog/retrieval-augmented-generation-RAG>
170. Retrieval-Augmented Generation for Large Language Models: A Survey - arXiv, accessed May 3, 2025, <https://arxiv.org/abs/2312.10997>
171. How does Retrieval Augmented Generation (RAG) actually work? : r/MLQuestions - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/MLQuestions/comments/16mkd84/how_does_retrieval_augmented_generation_rag/>
172. [D] retrieval-augmented generation vs Long-context LLM, are we sure the latter will substitute the first? : r/MachineLearning - Reddit, accessed May 3, 2025, <https://www.reddit.com/r/MachineLearning/comments/1fabu65/d_retrievalaugmented_generation_vs_longcontext/>