Comparison of LLM Customization Methods

Method	Definiton	Advantages	Disadvantages
Prompt Engineering	Strategically designing input prompts to guide model behavior through instructions, examples, and context without modifying the model's parameters.	○ Immediate implementation with no training required ○ Zero computational overhead for model modification ○ Highly flexible and easily adjustable ○ Cost-effective for most applications ○ No technical expertise in machine learning (ML) required	○ Performance is constrained by the amount of information that can fit into the prompt. ○ Outputs can be highly sensitive to small changes in the prompt, and the model may produce inconsistent results for the same prompt over time. ○ Does not permanently alter the model's underlying knowledge base or reasoning patterns; it only guides the output for a specific interaction. ○ Struggles to consistently achieve high performance on tasks that require deep, multi-step reasoning or specialized domain expertise
Few-Shot Learning	Teaching the model new tasks by providing a small number of input-output examples within the prompt, leveraging the model's in-context learning abilities.	○ No model training or modification required ○ Quick adaptation to new tasks ○ Works with any LLM via application programming interface (API) ○ Easy to iterate and improve ○ Combines well with other prompt techniques ○ No additional infrastructure needed	○ Performance is constrained by the amount of information that can fit into the prompt. ○ Outputs can be highly sensitive to small changes in the prompt, and the model may produce inconsistent results for the same prompt over time. ○ Does not permanently alter the model's underlying knowledge base or reasoning patterns; it only guides the output for a specific interaction. ○ Struggles to consistently achieve high performance on tasks that require deep, multi-step reasoning or specialized domain expertise
RAG (Retrieval-Augmented Generation)	Combining LLMs with external knowledge retrieval systems to access real-time, domain-specific information during generation.	○ Access to up-to-date information ○ Domain-specific knowledge without retraining ○ Reduced hallucination through factual grounding ○ Scalable knowledge base updates ○ Maintains general model capabilities	○ Requires setting up and maintaining an external knowledge base (e.g., a vector database) and a retriever, adding cost and technical overhead. ○ The final output is critically dependent on the accuracy of the retrieval system. If the retriever fetches irrelevant or incorrect documents, the output will be flawed. ○ The two-step process (retrieve, then generate)

		○ Cost-effective for knowledge-intensive tasks ○ Traceable information sources	makes it slower than querying the model directly. ○ The model does not "learn" the external information permanently; it only uses what is provided at the time of the query, limiting its ability to synthesize that knowledge in other contexts.
LoRA (Low-Rank Adaptation)	Parameter-efficient fine-tuning technique that freezes original model weights and trains small adapter modules inserted into the model architecture.	○ Much lower computational requirements than full fine-tuning ○ Preserves original model capabilities ○ Multiple task-specific adapters can coexist ○ Faster training compared to full fine-tuning ○ Reduced storage requirements ○ Lower risk of catastrophic forgetting	○ As a parameter-efficient compromise, it may not achieve the absolute peak performance possible with full fine-tuning on complex tasks. ○ While excellent for adapting style and behavior, it is less effective for teaching the model entirely new, broad domains of knowledge from scratch. ○ Although computationally cheaper, it still necessitates a high-quality, clean dataset to perform effectively. ○ Managing and deploying multiple task-specific adapters can introduce complexity into the application workflow.
Fine-tuning	Continuing the training process of a pre-trained model on a specific dataset to adapt it for particular tasks or domains.	○ Highest level of customization possible ○ Can achieve domain-specific expertise ○ Improved performance on target tasks ○ Better consistency and reliability ○ Can modify model's knowledge base ○ Reduced need for lengthy prompts	○ Requires significant investment in GPUs and can take a long time to train, making it the most expensive method. ○ The model may lose some of its original, general-purpose abilities as it specializes in the new task. ○ The model might memorize the training data instead of learning to generalize, leading to poor performance on new, unseen examples. ○ Demands very large and meticulously cleaned datasets, the creation of which is often a major bottleneck. ○ The entire process, from data preparation to hyperparameter tuning, requires a high level of machine learning expertise to manage successfully.