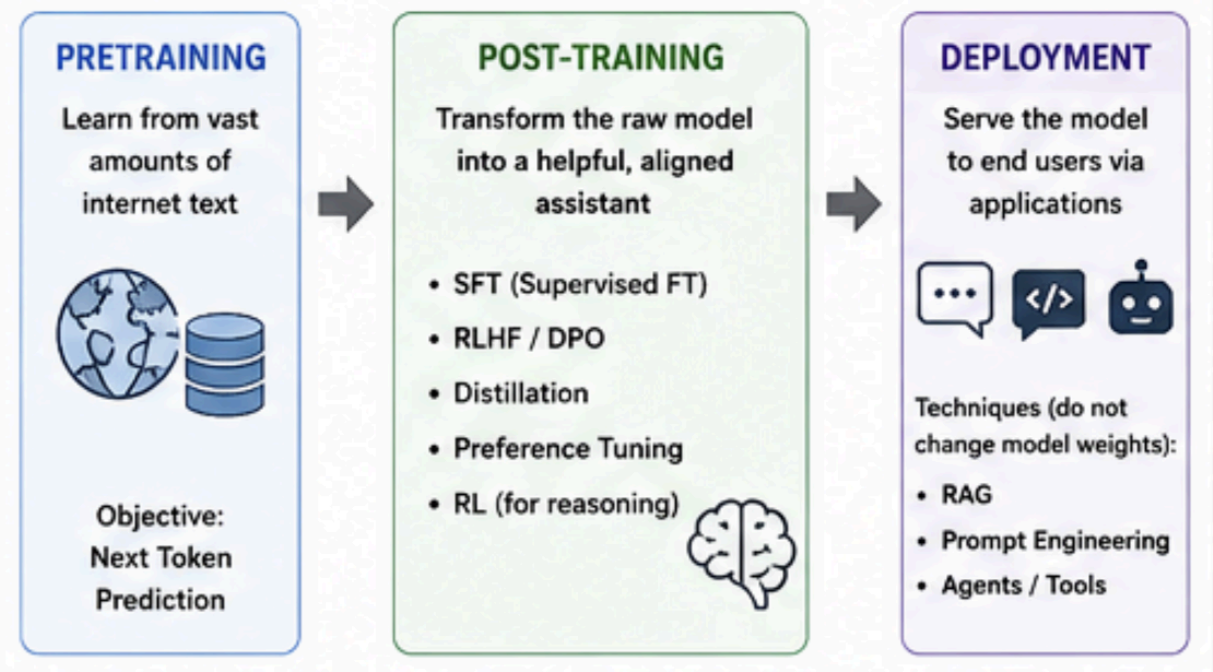
SFT & Post-Training: The 2026 Landscape

How does ChatGPT/gemini/claude etc know how to answer our questions so naturally?

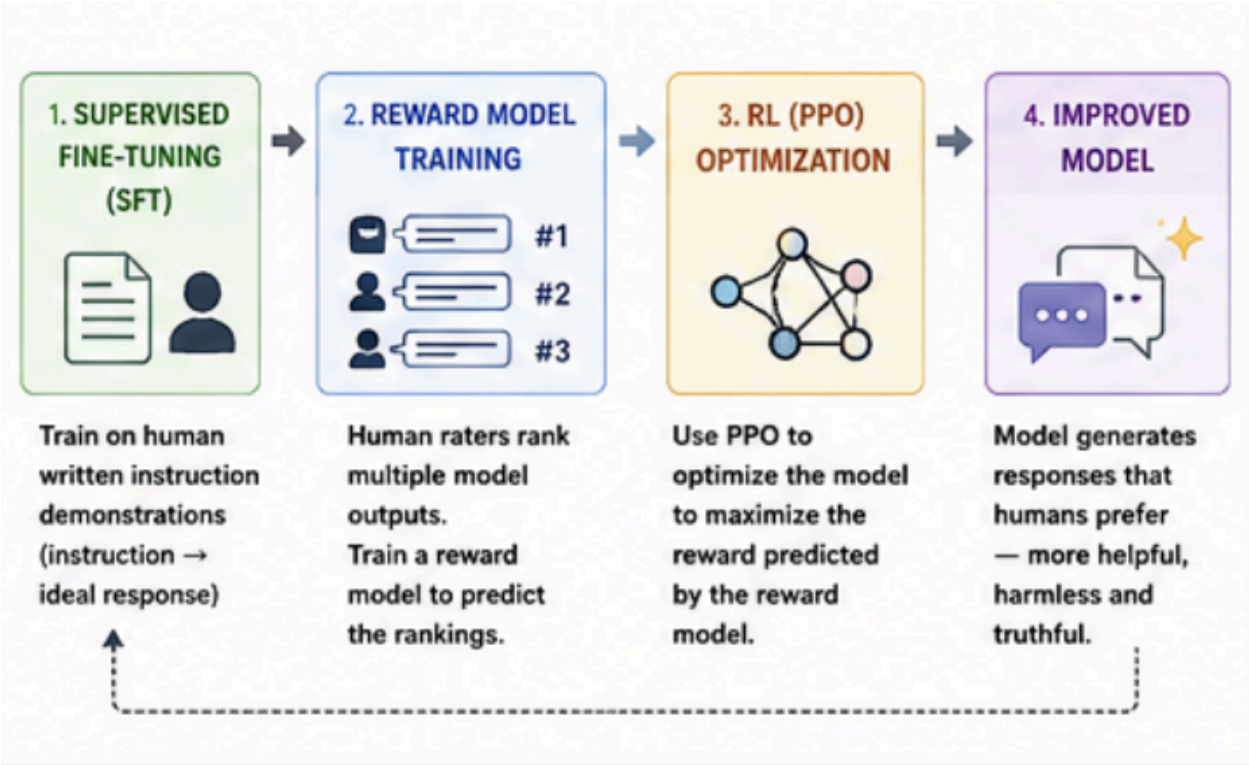
1. The LLM Lifecycle: Pretraining, Post-Training, and Deployment



To understand fine-tuning, one must first understand the three distinct stages of an LLM's life:

- **Pretraining:** The model processes vast amounts of raw internet text to learn how to predict the next word. At this stage, it acts as a powerful autocomplete engine but lacks the conversational framing of an "assistant".
- **Post-training:** This umbrella term encompasses all processes applied after pretraining to transform the raw model into a usable, aligned tool. It includes Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), Direct Preference Optimization (DPO), and distillation methodologies.
- **Deployment:** The final stage where the model is served to end-users. Adaptations here are lightweight and inference-based, such as Retrieval-Augmented Generation (RAG) and system prompting, which do not alter the underlying model weights.

2. Supervised Fine-Tuning (SFT) and the InstructGPT Paradigm



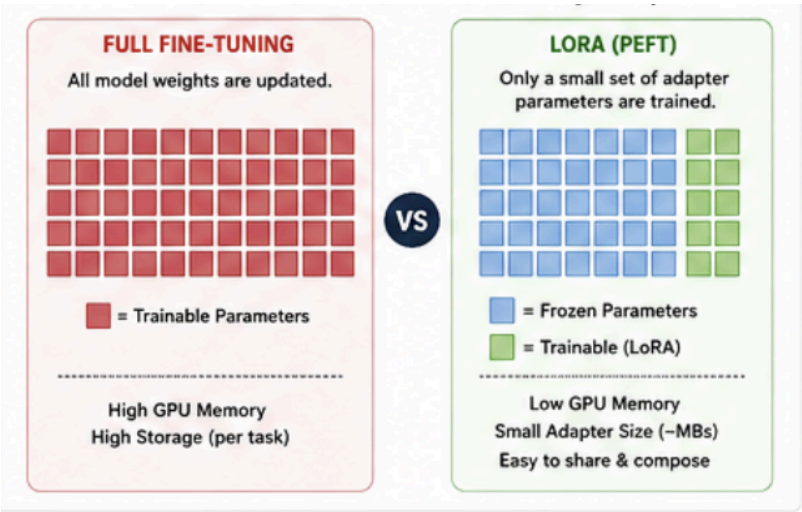
Supervised Fine-Tuning (SFT) generally serves as the first step of the post-training pipeline. During SFT, the pretrained model is trained on a highly curated dataset of (instruction→ideal-response) pairs. Because each example contains a definitive "correct" target output, the training is considered "supervised".

The modern paradigm for this pipeline was established by the 2022 InstructGPT research. The researchers introduced a three-stage methodology:

1. Initial SFT using demonstrations written by humans.
2. Training a distinct reward model based on how human raters ranked multiple model outputs.
3. Applying reinforcement learning (specifically Proximal Policy Optimization, or PPO) to optimize the model against the learned reward model.

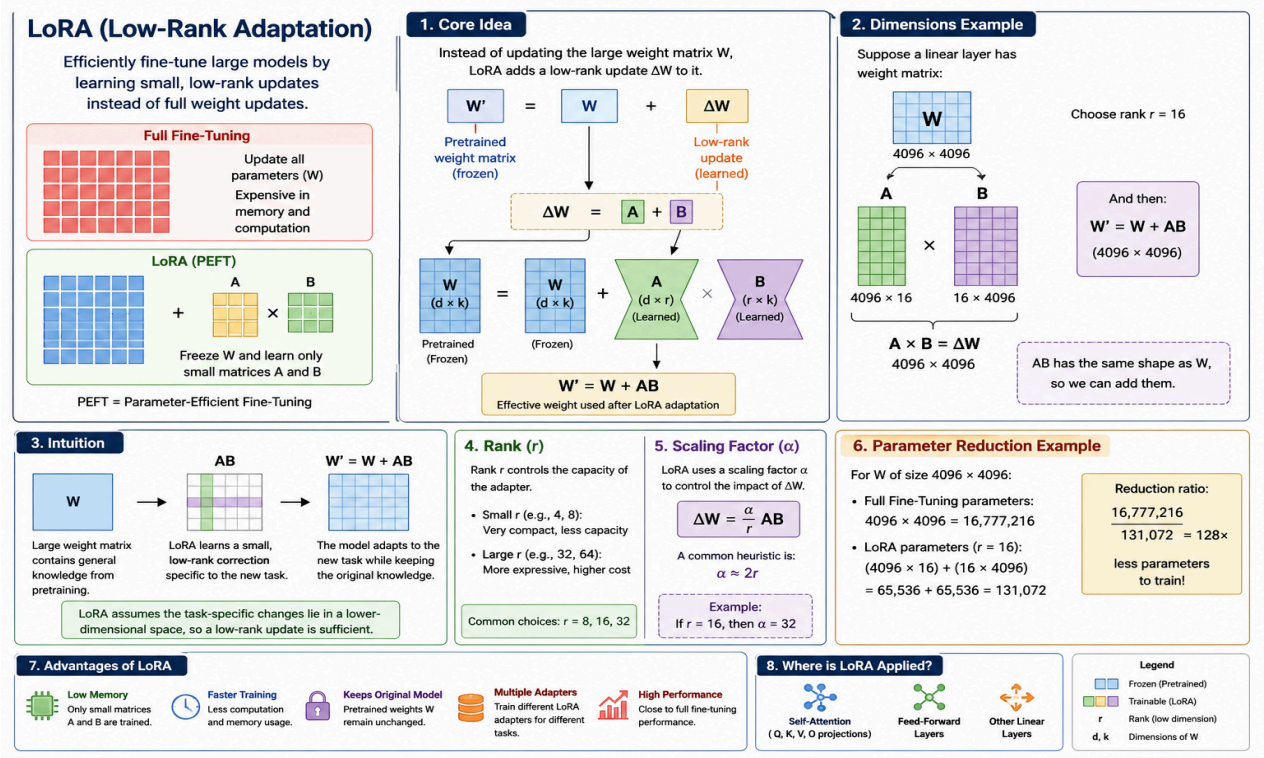
A profound finding from this research was that human raters consistently preferred the outputs of a 1.3-billion-parameter InstructGPT model over those of a raw 175-billion-parameter GPT-3 model. This heavily underscored that post-training methodologies can be more critical to user perception than raw parameter scale.

3. Full Fine-Tuning vs. Parameter-Efficient Fine-Tuning (PEFT)



While full fine-tuning updates every weight across the entire model, it requires massive GPU memory to simultaneously store the model, gradients, and optimizer states.Parameter-Efficient Fine-Tuning (PEFT) resolves this by freezing almost all base weights and training only a small subset of additional parameters .This approach allows for the creation of small, task-specific "adapters" rather than maintaining multiple massive copies of the full model.

Low-Rank Adaptation (LoRA)



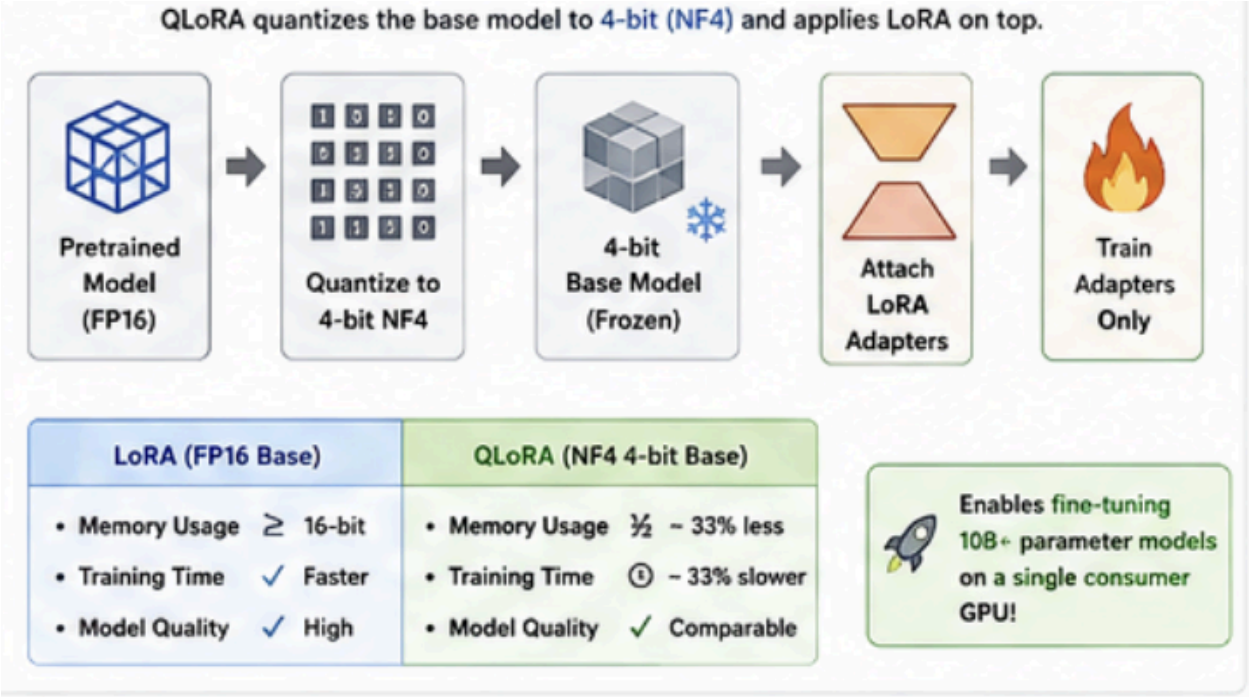
LoRA is the industry-standard PEFT method. Instead of calculating and applying a full weight matrix update, LoRA represents the update as the product of two much smaller matrices via low-rank decomposition. Only these small matrices are trained, while the base weights remain entirely frozen.This works because the mathematical adjustments required to adapt a model to a new task typically exist in a much lower-dimensional space than the model's full parameter count .

Extensive experimental data has revealed several practical guidelines for LoRA:

- Training results are surprisingly stable and consistent across varying runs .
- Applying LoRA matrices to all layers of the model, rather than just the attention key/value layers, yields meaningfully superior performance .
- A common, effective heuristic is setting the scaling factor ("alpha") to approximately twice the LoRA rank .
- Remarkably, a 7-billion-parameter model can be fully fine-tuned on a single consumer GPU with roughly 14GB of VRAM in a matter of hours .
- The specific choice of optimizer (e.g., Adam) has less impact on LoRA fine-tuning

outcomes than traditionally assumed.

Quantized LoRA (QLoRA)



QLoRA pushes hardware efficiency further by merging LoRA with weight quantization. The base model is compressed into lower precision, most commonly 4-bit NormalFloat (NF4), a format tailored specifically for normally-distributed network weights before the LoRA adapters are applied on top.
This creates a distinct operational trade-off: QLoRA reduces GPU memory requirements by roughly 33%, but increases overall training time by about 33% compared to standard LoRA. Crucially, this technique democratizes AI development by allowing models over 10 billion parameters to be fine-tuned on single consumer-grade GPUs.

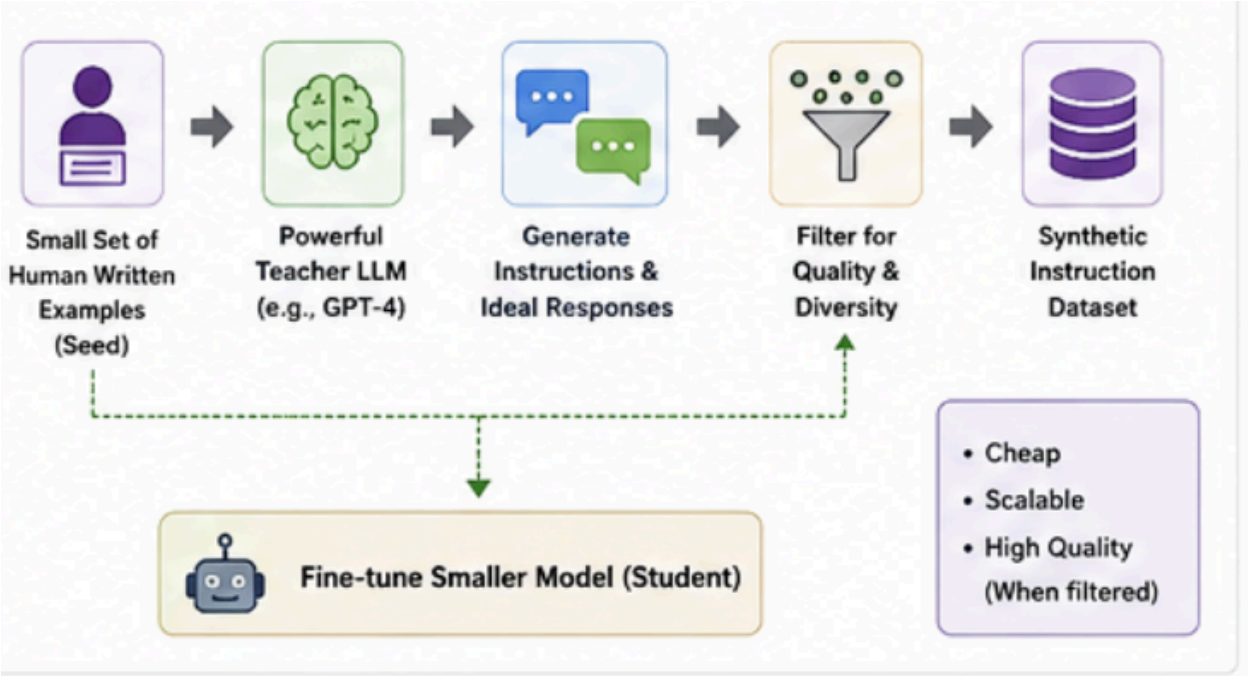
Alternative PEFT Architectures

Though LoRA is dominant, other PEFT methods exist under the same philosophy of freezing the base model .:

- **Adapters:** Trainable neural network layers injected between frozen layers, typically concentrated in the upper model layers handling high-level semantic representations ..
- **Prefix-tuning:** Involves prepending trainable "virtual token" vectors to the inputs of every layer, optimized via a small independent feed-forward network.
- **Prompt-tuning:** A lighter variant of prefix-tuning that only adds trainable tokens at the

initial input embedding layer, originally built for text classification tasks.

4. Training Data: Generation and Curation

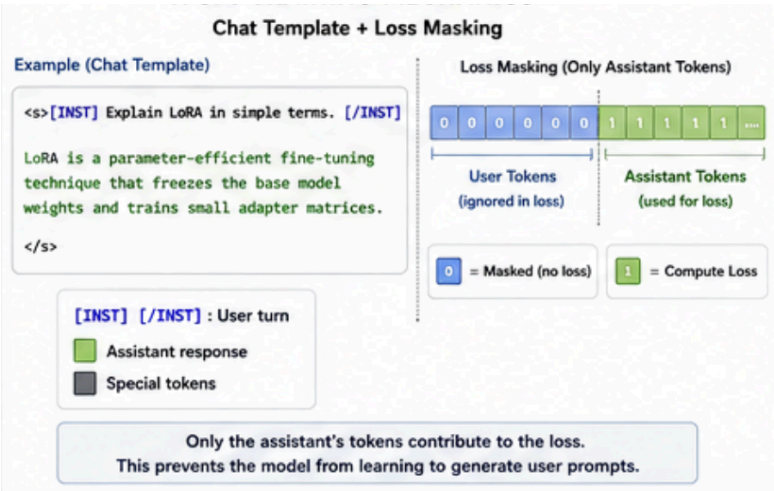


Procuring high-quality SFT data is often a steeper bottleneck than selecting the tuning methodology. While human-written demonstrations are gold-standard for quality, they are prohibitively expensive and slow to scale..

To bypass this, developers utilize **Self-Instruct**, an automated paradigm where a powerful existing LLM generates new instructions and ideal responses from a small human-written seed set, which are subsequently filtered for diversity and quality. A landmark example of this is the **Stanford Alpaca** project. Researchers successfully fine-tuned the LLaMA 7B model on 52,000 instruction pairs generated by OpenAI's text-davinci-003. The entire data generation process cost under $500, yet the resulting Alpaca model performed roughly on par with text-davinci-003 in blind human evaluations.. This process fundamentally mimics **Distillation**, where a smaller model is trained directly on the high-quality outputs of a larger, more capable parent model.

There is an active debate regarding synthetic-data quality. Since a large portion of SFT and preference data is now AI-generated (using Self-Instruct techniques), there is a known risk of "model collapse" if models are trained too heavily on other models' outputs instead of human-grounded data

5. Practical Mechanics of Fine-Tuning



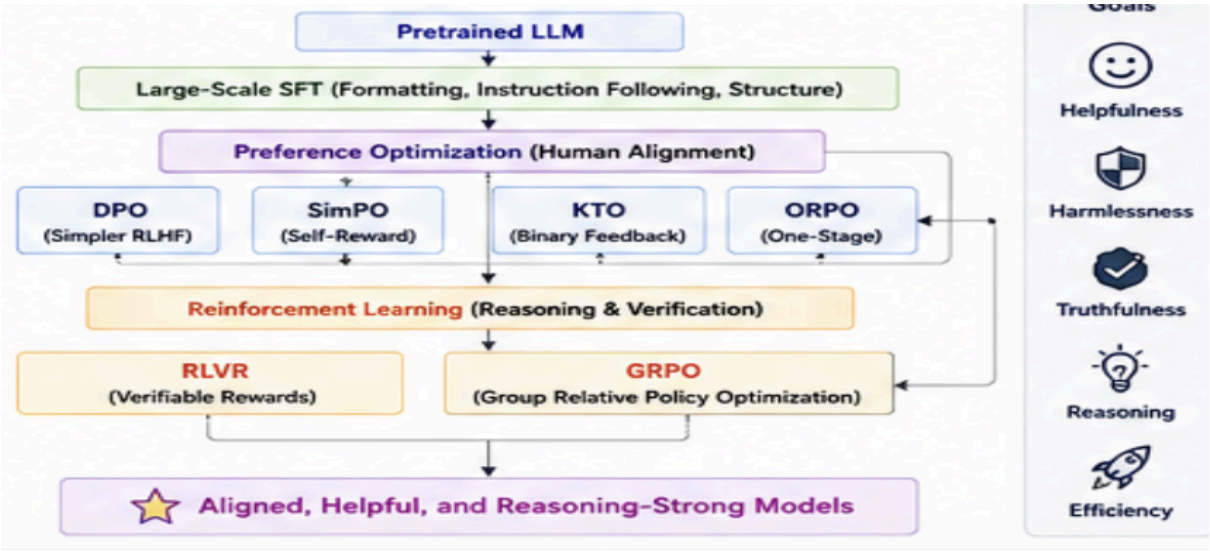
Executing a fine-tuning run requires specific formatting and data-handling mechanics:

- **Chat Templates:** Raw conversational data must be wrapped in specific text formats featuring special tokens that distinctly mark "user" versus "assistant" turns.
- **Loss Masking:** During SFT, the mathematical loss function is applied exclusively to the tokens generated by the assistant. User prompt tokens are masked out to prevent the model from learning how to generate user questions.
- **Packing:** To maximize GPU efficiency, multiple short training sequences are concatenated (packed) into a single long sequence, eliminating wasted compute cycles spent processing empty padding tokens.

Evaluating Post-Training Success

As benchmarks like AlpacaEval have become saturated, with frontier models clustering above 95% win-rates, the field has shifted to more robust evaluation methods. Current SOTA tracking emphasizes harder benchmarks to better discriminate between models: Arena-Hard-Auto, WildBench, and LiveBench for instruction-following, alongside MMLU-Pro and GPQA Diamond for reasoning discrimination.

6. The 2026 State of the Art (SOTA) in Post-Training



The classical pipeline of SFT followed by PPO-based RLHF has rapidly evolved. The 2026 post-training landscape is characterized by a modular toolbox of techniques optimized for specific outcomes.

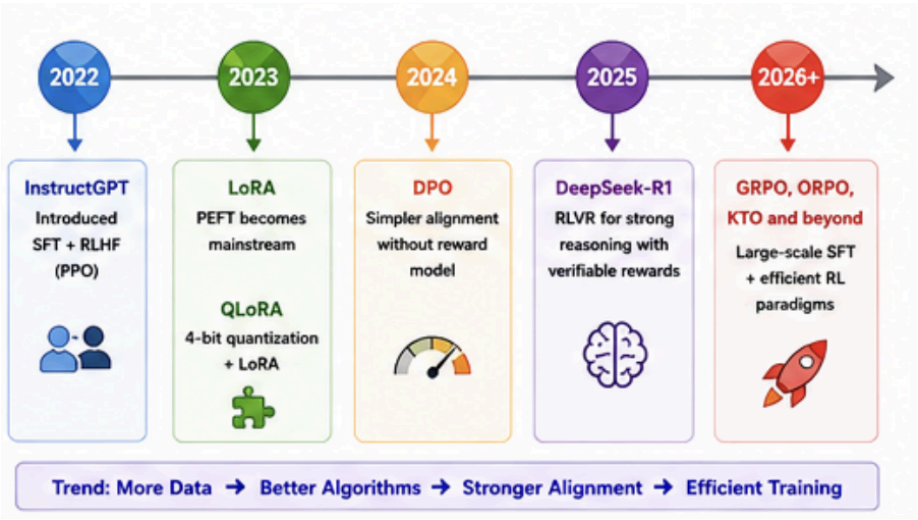
Modern SFT and Preference Optimization

Modern SFT operates at a massive scale; production models like Nemotron 3 Super leverage up to 7 million curated SFT samples drawn from a 40-million-example pool. SFT is now relied upon for teaching rigid formatting, deep instruction following, and structured output generation. For human alignment, **Direct Preference Optimization (DPO)** has heavily displaced traditional RLHF. By directly optimizing the model on chosen-versus-rejected response pairs, DPO entirely bypasses the expensive need to train a separate reward model. Furthermore, variations on preference tuning have emerged:

- **SimPO:** Outperforms standard DPO benchmarks by utilizing the model's own average response probability as an implicit reward, negating the need to store a reference model in memory.
- **KTO (Kahneman-Tversky Optimization):** Utilizes simple binary (thumbs-up/thumbs-down) signals rather than complex paired comparisons, drastically reducing data collection costs .
- **ORPO:** Seamlessly merges the SFT and preference-tuning stages into a single execution step, avoiding behavioral drift between training phases.

The Resurgence of Reinforcement Learning (RL)

While DPO simplified conversational alignment, Reinforcement Learning is experiencing a major resurgence for complex reasoning tasks. Techniques like **GRPO (Group Relative Policy Optimization)** and **RLVR (Reinforcement Learning from Verifiable Rewards)** generate multiple answers per prompt . Instead of querying a human-preference reward model, these frameworks score outputs using deterministic, mathematically verifiable signals, such as automated code execution or regex matching. This paradigm is the driving force behind the immense mathematical and coding capabilities seen in recent frontier models, most notably DeepSeek-R1.



7. Comparison of Post-Training Techniques

Term	Definition / Role in Pipeline
SFT	Supervised Fine-Tuning on labeled pairs to establish instruction following and formatting.
RLHF	Reinforcement Learning from Human Feedback using a trained

Term	Definition / Role in Pipeline
	human-preference reward model.
DPO	Direct Preference Optimization; a lightweight RLHF alternative skipping the separate reward model.
PEFT	Parameter-Efficient Fine-Tuning; tuning only a small fraction of added parameters to save memory.
LoRA	The dominant PEFT method representing weight updates as low-rank matrix decompositions.
QLoRA	Combines LoRA with 4-bit base model quantization (NF4) for immense memory savings.
Self-Instruct	Using an advanced LLM to autonomously generate training data from a small human seed set.

References

1. Ouyang et al., "Training language models to follow instructions with human feedback" (InstructGPT paper), 2022; IntuitionLabs; Queirozf.
2. Hugging Face PEFT documentation — conceptual guides on Adapters and Soft Prompts.
3. Sebastian Raschka, "Practical Tips for Finetuning LLMs Using LoRA" and related experiments.
4. arXiv survey (2410.00207) on QLoRA's 4-bit NormalFloat quantization.

5. Stanford Alpaca project documentation and GitHub repo.
6. Wang et al., "Self-Instruct: Aligning Language Models with Self-Generated Instructions", 2022.
7. llm-stats.com, "Post-Training in 2026: GRPO, DAPO, RLVR & Beyond".
8. Hugging Face, "A Guide to Reinforcement Learning Post-Training for LLMs" (2026) and TRL v1.0 documentation.
9. DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", 2025.