

Review

Training Methods for Large Language Models: Current Approaches and Challenges

Dimitris Karydas , Dimosthenis Margaritis  and Helen C. Leligou 

Department of Industrial Design and Production Engineering, Ancient Olive Grove Campus, University of West Attica, 250 Thivon & P. Ralli Str., 122 41 Egaleo, Greece; mscres-84@uniwa.gr (D.M.); e.leligkou@uniwa.gr (H.C.L.)

* Correspondence: karidasd23@uniwa.gr

Abstract

Large Language Models (LLMs) have emerged as a dominant paradigm in natural language processing, demonstrating strong performance across a wide range of generation and reasoning tasks. These systems depend on multi-stage training pipelines that integrate large-scale self-supervised pre-training, supervised fine-tuning, and alignment techniques. This paper presents a systematic mapping study of contemporary LLM training methodologies, emphasizing transformer-based architectures, optimization objectives, and data curation strategies as well as emerging sparse architectures such as Mixture-of-Experts (MoE) models. We analyze parameter-efficient fine-tuning approaches, retrieval-augmented generation frameworks, and multimodal training techniques, which we organize into a unified comparative taxonomy. We discuss key technical challenges such as scalability constraints, hallucination, bias amplification, and alignment–capability tradeoffs, then identify emerging research directions such as reasoning-centric training. This work provides a concise technical reference for researchers and practitioners working on scalable and reliable language model training.

Keywords: large language models; training strategies; transformer architectures; sparse models; mixture-of-experts; model alignment

1. Introduction

Large Language Models (LLMs) have rapidly become foundational technologies in modern artificial intelligence, particularly in Natural Language Processing (NLP). Models such as GPT, BERT, and LLaMA demonstrate an unprecedented ability to comprehend, generate, and increasingly reason over human language at scale. Their success is primarily driven by transformer-based architectures, large-scale datasets, and the availability of massive compute resources. As a result, LLMs now support a wide range of high-impact applications, including conversational agents, search engines, coding assistants, scientific discovery tools, and decision-support systems.

Understanding how LLMs are trained has become essential for both researchers and practitioners. First, pre-training state-of-the-art models requires substantial computational and financial investment, making training efficiency and scalability central engineering concerns. Second, training pipelines strongly influence model behavior, including the emergence of biases, hallucinations, and safety risks that originate from data and optimization objectives. Third, the modern LLM lifecycle is no longer limited to pre-training alone but increasingly depends on post-training stages such as instruction tuning, preference alignment (e.g., RLHF and DPO), and reasoning-oriented optimization.



Academic Editor: Chuangyin Dang

Received: 22 January 2026

Revised: 9 February 2026

Accepted: 13 February 2026

Published: 19 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

While many surveys have reviewed components of this landscape, the literature often treats training stages in isolation, focusing separately on pre-training objectives, fine-tuning strategies, or alignment methods. However, recent developments such as parameter-efficient adaptation, retrieval-augmented pipelines, and sparse Mixture-of-Experts (MoE) architectures highlight that LLM training is best understood as a multi-stage and evolving methodology rather than a single monolithic process. This motivates the need for a unified and systematic comparative perspective that connects foundational training pipelines with frontier reasoning-centric paradigms.

In this paper, we provide a structured systematic mapping study of contemporary LLM training approaches spanning the full pipeline, including pre-training foundations, supervised and instruction fine-tuning, alignment via reinforcement learning and preference optimization, Parameter-Efficient Fine-Tuning (PEFT), retrieval-augmented and multi-modal extensions, and sparse MoE architectures exemplified by recent DeepSeek models.

This work makes the following key contributions:

- **Systematic methodological synthesis:** We adopt a transparent survey methodology and organize an initial set of 58 core studies, expanded to 68 during revision, across major stages of the LLM training pipeline, supported by PRISMA-style reporting and thematic distribution.
- **Unified analytical taxonomy:** Beyond thematic review, we propose an original comparative framework positioning training paradigms along the axes of training efficiency and alignment/reasoning quality, offering new insight into dense scaling versus sparse and post-training optimization strategies.
- **Quantitative frontier comparison:** We strengthen the engineering perspective of the survey by including quantitative evidence from dense and sparse frontier model families (e.g., LLaMA-3 vs. DeepSeek MoE systems), highlighting the efficiency–capability tradeoffs shaping current research trends.

The remainder of this survey is organized as follows: Sections 3 and 4 discuss foundational pre-training and supervised adaptation; Section 5 focuses on alignment and PEFT techniques; Section 6 examines frontier extensions, including RAG, multimodality, and MoE reasoning-centric architectures; while the later sections summarize evaluation challenges and emerging future directions.

2. Survey Methodology

This systematic mapping study followed a structured protocol to ensure transparency, reproducibility, and reduced selection bias. We searched Scopus, Google Scholar, Web of Science, and IEEE Xplore using keywords including “large language models,” “LLM training,” “pre-training objectives,” “fine-tuning,” “RLHF,” and “parameter-efficient fine-tuning” for publications between 2017 and 2025.

The initial search returned approximately 1500 records. After duplicate removal and title/abstract screening, 240 studies were retained for full-text review. Inclusion criteria prioritized peer-reviewed publications with an explicit focus on LLM training pipelines, empirical or theoretical methodological contributions, and full-text availability. Selected high-impact frontier preprints were also incorporated given the rapid pace of development in the field.

For classification as “high-impact frontier preprints”, we applied explicit operational criteria. Included preprints were required to demonstrate substantial scholarly uptake relative to publication recency, operationalized as at least 100 citations, or at least 50 citations for works published within the preceding 12 months (as indexed via Google Scholar at the time of writing). Citation thresholds were interpreted flexibly in light of the rapid evolution of the field. In addition, preprints were considered eligible where there was clear

evidence of field impact, such as integration into subsequent peer-reviewed publications, inclusion in major technical reports, benchmark reporting, or widespread adoption in open-source ecosystems.

Exclusion criteria removed non-English works, non-LLM-specific studies, and papers lacking methodological contributions, resulting in 58 core studies included in the initial analysis (Figure 1).

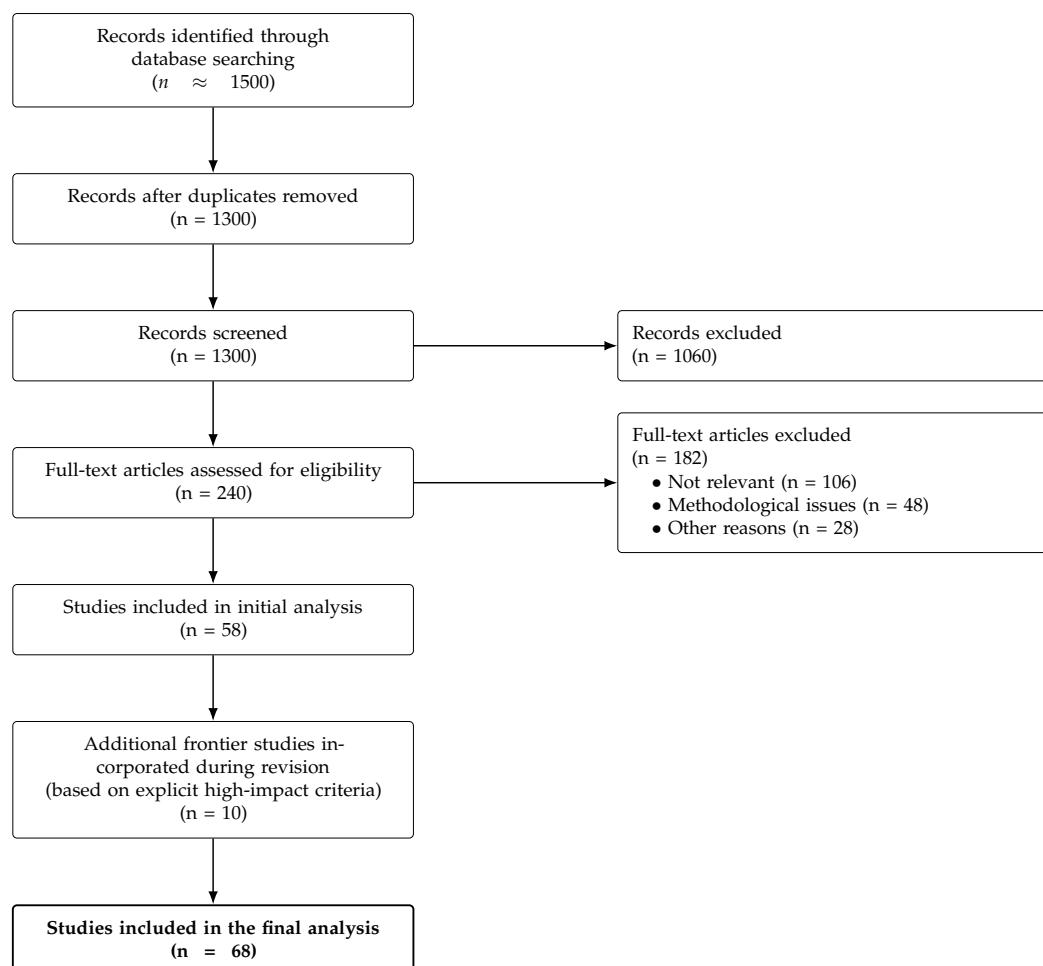


Figure 1. PRISMA flow diagram of the study selection process. The final corpus includes an additional set of frontier studies ($n = 10$) incorporated during revision based on explicit high-impact criteria described in the Survey Methodology section.

Following reviewer feedback, an additional set of recent frontier works was incorporated, covering emerging advances in reasoning-centric alignment (e.g., verifiable reward optimization), preference-based post-training, and safety-focused evaluation frameworks. This expansion yielded a final corpus of 68 included studies.

To support transparency, we also report the thematic distribution of the included studies across the survey categories (Table 1). Overall, the goal is to provide a structured methodological synthesis of representative paradigms shaping modern LLM training, rather than claiming exhaustive completeness of the entire literature.

Research Questions

To guide the mapping and comparative synthesis, the study is structured around the following research questions:

- RQ1: What are the dominant strategies in modern LLM pre-training and data curation?

- RQ2: How do Parameter-Efficient Fine-Tuning (PEFT) and alignment methods balance computational efficiency with response quality and safety?
- RQ3: To what extent do sparse Mixture-of-Experts (MoE) architectures improve the capability–efficiency tradeoff compared to dense transformer models?

Table 1. Distribution of the 68 included studies across survey categories. The final set was expanded during revision to incorporate recent frontier advances in preference optimization (DPO), verifiable-reward alignment (RLVR), tool-augmented LLMs, and representation-based evaluation.

Category	Included Studies
Foundation Pre-training Objectives	19
Supervised/Instruction Fine-Tuning (SFT/IFT)	9
Alignment and Post-training + PEFT	16
RAG/Multimodal/Reasoning-centric (Sparse MoE case studies, e.g., DeepSeek)	16
Evaluation Benchmarks and Recent Advances	9
Total	68

3. Foundations of Pre-Training for Large Language Models

In Large Language Models (LLMs), the pre-training stage aims to build a fundamental understanding of language and gather enormous amounts of world knowledge that are encoded within the model’s parameters. This is accomplished by self-supervised learning, in which the model learns by anticipating patterns and information gaps in large, varied text datasets, effectively converting the data into the supervision signal [1]. This research abstract [2] suggests that while Masked Language Modeling (MLM) generally produces better overall results for text representation tasks, Causal Language Modeling (CLM) offers advantages in data efficiency and fine-tuning stability. Recognizing these tradeoffs, the study proposes and validates a biphasic training strategy that sequentially applies CLM followed by MLM. This method achieves optimal performance within a fixed computational budget. Furthermore, this biphasic approach is particularly beneficial when starting with an already pre-trained CLM model, significantly lowering the computational cost required to train state-of-the-art encoder models for subsequent NLP tasks.

Large Language Models (LLMs) can learn fundamental linguistic knowledge and generalizable representations from a variety of textual sources through the pre-training phase, which combines precisely defined training objectives with carefully selected large-scale datasets [3].

A crucial component of the pre-training phase in Large Language Models (LLMs) is the construction of high-quality large-scale datasets, as the model’s generalization capability depends heavily on the diversity and cleanliness of the data it is exposed to. Modern foundation models are trained on heterogeneous corpora that combine web-scale sources with curated high-quality text. The largest portion typically originates from Common Crawl, a massive but noisy web dataset that requires extensive preprocessing and filtering to be usable at scale [4]. Additional high-quality sources such as Wikipedia and large book corpora are included to provide structured linguistic knowledge for the development of robust contextual representations [5]. For improved multilingual generalization, many models incorporate multilingual datasets spanning diverse languages and domains, while LLMs aimed at coding tasks integrate specialized programming corpora such as The Stack (BigCode, 2023 public release) or large GitHub-derived datasets collected from publicly available repositories [6].

Given the raw and noisy nature of web-scale data, sophisticated data filtering pipelines are essential. These include deduplication, which reduces memorization and improves gen-

eralization performance, as well as quality filtering based on heuristics, perplexity scoring, and classifier-based content selection. Equally important are safety filtering mechanisms designed to remove personally identifiable information (PII), toxic content, or harmful material, thereby mitigating ethical and security risks during training [7]. Overall, the dataset composition, scale, and curation procedures play a foundational role in shaping the performance, safety, and bias characteristics of large language models [8].

As shown in Table 2, datasets with diverse characteristics and tradeoffs are necessary for effective LLM training. Common Crawl and other large-scale web corpora provide a great deal of coverage and diversity, but also contain a lot of bias and noise. Deduplication and data cleaning are therefore crucial. However, despite their much smaller scope, resources such as Wikipedia offer trustworthy and high-quality information. These datasets are enhanced by BookCorpus and OpenWebText, which offer lengthy, cohesive texts that aid in discourse-level comprehension. In conclusion, careful balancing of the scale, quality, and diversity of data sources is just as important for successful LLM training as dataset size.

Table 2. Representative datasets used for LLM pre-training, summarizing dataset type, approximate scale, and key engineering notes (noise level, need for deduplication/cleaning, domain specialization, and safety filtering). The table illustrates the tradeoff between web-scale coverage (e.g., Common Crawl) and curated quality sources (e.g., Wikipedia/books). Dataset sizes are indicative and may vary across preprocessing and release versions.

Dataset	Type	Size	Notes	Refs
Common Crawl	Web text	~100 TB	Web-scale coverage; very large and noisy; extensive deduplication and filtering required	[5,9]
Wikipedia	Encyclopedia	~15 GB	High-quality curated source; widely used across multilingual LLM corpora	[9]
BookCorpus/ OpenWebText	Books, web	~10–20 GB	Literature-based corpora supporting general language and world knowledge	[5,10]
Code datasets	Code (GitHub, StackOverflow)	10–50 GB	Specialized corpora enabling strong performance in code generation tasks	[6]
Multilingual corpora	Text	Varies	Supports cross-lingual transfer and multilingual instruction-following	[9]
Safety-filtered corpora	Text	Varies	Incorporates PII removal, toxicity filtering, and responsible data governance	[7,8]
RefinedWeb-style corpora	Web text	Large-scale	Curated web datasets designed to outperform raw Common Crawl through cleaning	[4]

The choice of large-scale high-quality datasets directly informs the architectural design of Large Language Models (LLMs), as different transformer variants (decoder-only, encoder-only, or encoder–decoder) leverage the data in distinct ways [1,5,11].

Modern LLMs are built upon the transformer architecture, which introduced attention-based sequence processing and replaced recurrent layers with highly parallelizable self-attention mechanisms [11]. Pre-training approaches differ significantly depending on the architectural variant employed. Decoder-only transformers, used in models such as GPT-3 and LLaMA, follow a unidirectional causal attention pattern that restricts each token to attend only to past positions, making them particularly effective for autoregressive next-token prediction and large-scale generative tasks [5,9,12]. In contrast, encoder-only architectures, exemplified by BERT, employ bidirectional self-attention, allowing the model to leverage both left and right context when generating token representations. When combined with the masked language modeling objective, this structure makes encoder models highly effective for classification, retrieval, and semantic understanding tasks rather than generation [1].

A third category, encoder–decoder (seq-to-seq) transformers, integrates the strengths of both components: the encoder builds contextualized representations using bidirectional attention, while the decoder autoregressively generates output sequences conditioned on the encoder states. Prominent examples such as T5 and UL2 use span corruption or mixture-of-denoisers objectives, and are well suited for tasks including translation, summarization, and instruction-following [9]. Collectively, these architectural families define the capabilities, strengths, and limitations of modern LLMs by shaping the training objectives, computational requirements, and downstream application suitability. The architecture of a Large Language Model (LLM) guides the selection of pre-training objectives, with decoder-only models typically employing Causal Language Modeling (CLM) for generative tasks, encoder-only models using Masked Language Modeling (MLM) for comprehension and classification, and encoder-decoder models adopting span corruption or mixture-of-denoisers objectives to support versatile sequence-to-sequence tasks.

The choice of training objective is a critical factor shaping the linguistic competence and downstream performance of LLMs, varying depending on architecture and intended tasks. Causal Language Modeling (CLM), used in decoder-only models like GPT and LLaMA, predicts each token conditioned on previous tokens, enabling effective text generation and stable fine-tuning [5,9]. In contrast, Masked Language Modeling (MLM), employed by encoder-only architectures such as BERT, randomly masks tokens and predicts them using bidirectional context, producing rich contextual representations suitable for classification and retrieval tasks [1]. Span corruption objectives are common in encoder–decoder models like T5 and UL2; these mask contiguous spans of text and train the model to reconstruct them, facilitating sequence-to-sequence tasks such as translation, summarization, and instruction following [10,13]. Recent approaches, exemplified by UL2, adopt a mixture-of-objectives pre-training strategy that dynamically combines CLM, MLM, and span denoising per batch, producing versatile models capable of both generative and comprehension tasks [13]. Together, these objectives define the functional capabilities of LLMs: CLM for generation, MLM for understanding, span corruption for seq-to-seq tasks, and unified objectives for multi-paradigm versatility.

In addition to dataset selection, architectures, and training objectives, several auxiliary factors critically influence LLM pre-training. Scaling laws demonstrate that model performance improves predictably with the number of parameters, the size of the training corpus, and the total compute budget, guiding the design of state-of-the-art models [14,15]. The training process itself requires massive computational resources, typically involving GPU or TPU clusters, distributed training paradigms such as data and model parallelism with ZeRO or FSDP, and mixed-precision arithmetic (FP16/BF16) to reduce memory consumption, often augmented with kernel optimizations like FlashAttention [16,17]. Finally, tokenization and preprocessing are essential for effective representation: Byte-Pair Encod-

ing (BPE), WordPiece, or SentencePiece tokenizers convert raw text into discrete tokens, while normalization, lowercasing, punctuation cleaning, and deduplication improve data quality and generalization [1,18]. Together, these factors ensure that pre-training produces high-quality, robust, and scalable language models.

4. Fine-Tuning Strategies for Large Language Models: Supervised and Instruction Tuning

After pre-training, LLMs typically undergo Supervised Fine-Tuning (SFT) to specialize in specific tasks or domains using curated input–output pairs. In this stage, human-labeled data provides explicit supervision for desired model behavior. Common instruction-following datasets include Alpaca [19], OpenAssistant (OASST) [20], Dolly [21], and FLAN [22], which span a broad range of topics and interaction styles. SFT enhances the model’s ability to follow instructions and improves zero-shot and few-shot generalization in downstream applications [23]. However, SFT may also propagate annotator biases, highlighting the importance of dataset diversity and quality control [24].

Building on SFT, instruction tuning extends supervised adaptation by exposing LLMs to diverse natural language instructions across many task formats, including summarization, classification, reasoning, and multi-step problem solving [24]. Large-scale instruction collections such as FLAN [22], Self-Instruct [25], and Natural Instructions provide broad coverage of task formulations, enabling models to generalize more effectively to unseen prompts and interactive settings [23].

Importantly, both SFT and instruction tuning often serve as prerequisites for downstream alignment methods, such as Reinforcement Learning from Human Feedback (RLHF), by providing a strong supervised baseline that can later be refined to optimize helpfulness, safety, and human preference alignment [26,27]. Thus, supervised and instruction-based fine-tuning form the central adaptation stage between foundation pre-training and modern preference-based alignment pipelines.

5. Advanced Fine-Tuning, Alignment, and Parameter-Efficient Adaptation in Large Language Models

5.1. Advanced Fine-Tuning and Emerging Alignment Strategies

In recent years, research on LLMs has continued to advance rapidly, extending beyond traditional SFT and RLHF pipelines. New methods focus on improving factuality, alignment, and reasoning as well as on increasing scalability and efficiency in fine-tuning. For example, FLAME [28] introduces factuality-aware alignment to reduce hallucinations, while RLTHF [29] and Crowd-SFT [30] explore cost-effective human feedback strategies for model alignment. Reflection-Tuning [31] enhances instruction-tuning datasets through model introspection, while PAFT [32] proposes parallel fine-tuning paradigms for efficient scaling. Additionally, recent methods such as Re-Search [33] and AGRO [34] combine reinforcement learning with search or off-policy data to improve reasoning capabilities and alignment. These recent developments demonstrate that the field is actively evolving, offering novel techniques that complement classical pre-training, fine-tuning, and alignment approaches.

Recent developments in have focused on enhancing model performance, reliability, and efficiency through advanced fine-tuning and alignment strategies. Techniques such as Reinforcement Learning from Human Feedback (RLHF) enable models to align with human preferences, improving safety, factuality, and instruction-following capabilities [35]. Parallel to this, advanced fine-tuning methods such as FLAME, Reflection-Tuning, and AGRO optimize reasoning, reduce hallucinations, and increase scalability. Parameter-Efficient Fine-Tuning (PEFT) approaches such as LoRA, QLoRA, prefix tuning, and adapters further

complement these methods by allowing LLMs to adapt to new tasks while updating only a small fraction of parameters, dramatically reducing computational cost and memory requirements [36,37].

5.2. Reinforcement Learning from Human Feedback (RLHF) and Preference Optimization

Reinforcement Learning from Human Feedback (RLHF) is a crucial post-training stage that aligns Large Language Models (LLMs) with human preferences, safety, and usefulness [26,27]. In RLHF, a reward model is first trained on human-labeled preference data to capture which outputs are more desirable for a given prompt [27]. The base LLM is then optimized using reinforcement learning algorithms such as Proximal Policy Optimization (PPO) [26] or preference-based alternatives such as Direct Preference Optimization (DPO) [38], with the goal of maximizing the learned reward signal. This alignment process improves instruction-following behavior and response quality while mitigating harmful or biased outputs [26]. Recent work further emphasizes scalable RLHF pipelines that incorporate selective feedback strategies and data recycling to improve efficiency and factual reliability [29].

5.3. Factuality-Aware and Reasoning-Centric Alignment Methods

Beyond RLHF, modern LLM development increasingly relies on a broader suite of advanced fine-tuning and alignment techniques. For instance, factuality-aware alignment methods such as FLAME aim to reduce hallucination errors [28], while Reflection-Tuning improves instruction datasets through model self-correction [31]. Reinforcement-based approaches such as AGRO further explore scalable optimization for reasoning and alignment [34]. Together, these methods complement supervised fine-tuning and instruction tuning by strengthening reliability, robustness, and controllability in deployed systems [24].

5.4. Parameter-Efficient Fine-Tuning (PEFT) for Scalable Adaptation

Parameter-Efficient Fine-Tuning (PEFT) has emerged as a key strategy for adapting LLMs to new tasks or domains while updating only a small fraction of parameters. Techniques such as LoRA enable low-rank adaptation of attention weights [37], while QLoRA combines low-rank updates with quantization to support efficient fine-tuning of very large models under limited GPU resources [39]. Other approaches, including prefix tuning and adapter modules, similarly reduce memory and computational overhead by freezing most pre-trained weights [40,41]. Empirical studies show that PEFT methods can achieve performance comparable to full fine-tuning while requiring orders of magnitude fewer trainable parameters [37,39]. As a result, PEFT provides a scalable pathway for integrating task adaptation with alignment objectives such as RLHF, preserving instruction-following and safety while improving efficiency [41]. From an engineering perspective, PEFT methods achieve dramatic reductions in trainable parameters and hardware requirements. For example, LoRA typically updates less than 0.1% of model weights, corresponding to parameter reductions on the order of 99.9% compared to full fine-tuning [37]. In addition, QLoRA enables fine-tuning of models up to 65 B parameters on a single 48 GB GPU by combining 4-bit quantization with low-rank adapters, making alignment and adaptation feasible under commodity hardware constraints [39].

Table 3 summarizes major PEFT approaches, reporting typical reductions in trainable parameters and VRAM requirements alongside practical integration characteristics.

Table 3. Overview of major Parameter-Efficient Fine-Tuning (PEFT) methods for adapting LLMs. We report concrete engineering gains, including reductions in trainable parameters and typical VRAM savings, highlighting why PEFT is critical under constrained GPU memory budgets.

Technique	Trainable Params Reduction	VRAM Reduction (Typical)	Integration	Engineering Notes	Refs
LoRA	~99.9% fewer trainable weights	~60–80%	Easy	Low-rank adapters replace full weight updates, enabling efficient adaptation with near full fine-tuning quality in many settings.	[37]
Prefix Tuning	~98–99% reduction	~40–60%	Moderate	Optimizes only prefix vectors injected into attention; reduces memory/compute, but can be less plug-and-play than LoRA.	[41]
Adapters	~95–99% reduction	~50–70%	Easy	Adds small trainable bottleneck modules between transformer blocks while freezing backbone weights; simple modular deployment.	[41]
QLoRA	65B tuning on 1 × 48 GB GPU (4-bit)	~85–95%	Easy	Combines 4-bit quantization with LoRA to enable large-model fine-tuning under constrained VRAM while preserving quality.	[39]

Note: VRAM reductions are approximate ranges reported in typical implementations and may vary depending on optimizer states, batch size, and hardware setup.

6. From Multimodal and Retrieval-Augmented Models to Reasoning-Centric Training: Sparse MoE Case Studies (e.g., DeepSeek)

Recent developments in LLM research have expanded model capabilities beyond traditional text-based tasks. Multimodal LLMs, such as GPT-4, Gemini, and Claude-class systems can process and generate content across multiple modalities, including text, images, and in some cases audio and video streams [42,43]. In parallel, specialized LLMs targeting domain-specific knowledge have also emerged. Examples include biomedical models such as BioGPT and legal assistants such as LawGPT, which provide improved accuracy and relevance in specialized professional contexts [44].

Another crucial trend shaping modern LLM capabilities is Retrieval-Augmented Generation (RAG), which addresses the inherent limitations of static pre-training knowledge by enabling dynamic access to external information sources [45]. By integrating retrieved documents as additional context during generation, RAG systems improve factuality, recency, and contextual awareness [46]. Recent extensions further explore multimodal RAG paradigms, incorporating heterogeneous inputs such as images alongside text [42].

In parallel, advances in the theoretical understanding of model scaling continue to guide architectural development. Scaling laws demonstrate predictable relationships between model performance, parameter count, dataset size, and computational budget [14,15]. This scaling process can also lead to emergent capabilities, with very large models exhibiting complex behaviors such as advanced reasoning or instruction-following that are not present in smaller counterparts [15]. Collectively, these extensions represent the frontier of modern LLM research, combining enhanced versatility through retrieval and multimodality with improved generalization driven by compute-efficient scaling strategies [47]. Sparse Mixture-of-Experts (MoE) architectures constitute a broader efficiency paradigm that has been explored across multiple major LLM families, including Switch Transformers and Mixtral as well as recent industrial-scale implementations. In this context, DeepSeek is discussed as a representative recent case study rather than a uniquely dominant approach.

DeepSeek represents a recent high-impact implementation within the broader literature on efficiency-oriented LLM training, combining sparse architectures with hybrid optimization strategies [48]. Importantly, sparse Mixture-of-Experts (MoE) designs are a general scaling paradigm adopted across multiple major model families, rather than being unique to DeepSeek.

Unlike dense transformer-based models that activate all parameters during inference, DeepSeek adopts a Mixture-of-Experts (MoE) architecture in which only a subset of expert networks is activated per token [49]. This sparse activation pattern reduces effective computational cost while preserving overall model capacity and scalability [48].

Therefore, recent advances highlight a clear trend toward hybrid paradigms that integrate multimodal processing, retrieval augmentation, and reasoning-focused post-training. DeepSeek exemplifies this next generation of efficiency-aware models by combining sparse MoE architectures with structured supervision and scalable alignment strategies [50,51].

Quantitative Efficiency in Sparse MoE Architectures: The DeepSeek Case Study

Sparse Mixture-of-Experts (MoE) architectures have been explored extensively in prior work, including early sparsely-gated layers [49] and large-scale Switch Transformer models [52]. We focus on DeepSeek as a contemporary representative implementation that provides publicly reported quantitative efficiency metrics. While DeepSeek is often described qualitatively as an “efficient” reasoning-centric model, quantitative comparisons are necessary to substantiate such claims within an engineering survey. Sparse Mixture-of-Experts (MoE) architectures achieve scalability by activating only a small subset of parameters per token, significantly reducing the effective compute footprint during training and inference.

Table 4 provides a quantitative comparison between DeepSeek-style MoE models and representative dense baselines. Although MoE models have very large total parameter counts, their activated parameters remain comparable to much smaller dense models while achieving strong benchmark performance (e.g., MMLU). This supports the central claim that sparse activation can approach or exceed dense scaling at substantially lower effective computational cost.

Table 4. Quantitative comparison of DeepSeek-style sparse MoE models against representative dense baselines. Training cost proxies are approximate and based on values reported in the corresponding technical reports. We report total versus activated parameters, token counts, and MMLU benchmark performance.

Model	Arch.	Total	Active	Tokens	Training Cost Proxy	MMLU	Refs
DeepSeek-V2-Base	MoE	236 B	21 B	8.1 T	~1.40 M H800 GPU-h [†]	78.4	[48]
DeepSeek-V3-Base	MoE	671 B	37 B	14.8 T	2.79 M H800 GPU-h	87.1	[48,50]
LLaMA-3.1-405B	Dense	405 B	405 B	–	–	84.4	[53]
Qwen2.5-72B	Dense	72 B	72 B	–	–	85.0	[54]

[†] Training cost figures are approximate and derived from values reported in the DeepSeek technical report [48]. DeepSeek-V2 reports roughly 172.8 K H800 GPU-hours per trillion tokens; multiplying by 8.1 T tokens gives an estimated ~1.40 M GPU-hours.

As a running example, the training pipeline of DeepSeek models follows a multi-stage paradigm that reflects broader trends in frontier LLM development [55]. The process begins with large-scale self-supervised pre-training using causal language modeling objectives over trillions of tokens [48].

In the post-training stage, DeepSeek adopts a hybrid alignment strategy that combines Supervised Fine-Tuning (SFT) with reinforcement learning-based optimization [51]. Unlike classical Reinforcement Learning from Human Feedback (RLHF) pipelines [26,27], recent DeepSeek models emphasize reasoning-centric alignment objectives driven by more structured and verifiable reward signals [56]. In particular, automated verifiers and heuristic solvers can be used to score intermediate reasoning steps, enabling correctness-oriented reinforcement learning beyond purely preference-based feedback [51].

Overall, DeepSeek serves as a representative example of an emerging paradigm that integrates sparse Mixture-of-Experts (MoE) architectures for cost-efficient large-scale training [48,49] together with scalable reasoning-focused post-training [51,56].

DeepSeek further incorporates sparse and long-context attention mechanisms to improve its handling of extended sequences [57]. Taken together, these design choices illustrate a broader transition toward hybrid efficiency-aware and reasoning-centric LLM development combining sparse computation, structured supervision, and scalable alignment techniques [58,59].

To complement the comparative overview presented in Table 3 and the architectural comparison in Table 5, Figure 2 illustrates the standard training pipeline for contemporary LLMs. The pipeline consists of a multi-stage process designed to transition from broad linguistic understanding to specialized human-aligned behavior. As illustrated, this process begins with Stage 1 (pre-training), followed by Stage 2 (SFT), and finally Stage 3 (alignment), utilizing Reinforcement Learning from Human Feedback (RLHF) or Direct Preference Optimization (DPO). Throughout these stages, Parameter-Efficient Fine-Tuning (PEFT) methods such as LoRA can be integrated to optimize computational resources.

Table 5. Comparison of multimodal and retrieval-related large language models. We distinguish between retrieval enabled via external tool augmentation [60] and native architectural Retrieval-Augmented Generation (RAG) integration [45].

Model	Architecture	Modalities	Retrieval Mechanism	Refs
BioGPT	Dense Transformer	Text	None	[44]
Claude 3	Dense Transformer	Text, Image	Tool-augmented retrieval	[43]
DeepSeek (V2/V3/R1)	Sparse MoE	Text	External/system-level RAG	[48,50,51]
Gemini	Multimodal Dense	Text, Image, Audio	Tool-augmented retrieval	[43]
GPT-4 class models	Multimodal Dense	Text, Image	Tool-augmented retrieval	[42,43]
LawGPT	Dense Transformer	Text	None	[44]
RAG-LLaMA	Dense + Native RAG	Text + Docs	Native RAG integration	[45]

Note: Retrieval mechanisms may be implemented either as externally orchestrated pipelines or as components more tightly integrated into the model architecture.

Overall, DeepSeek serves as a representative contemporary implementation of sparse MoE scaling; however, similar efficiency principles are shared across other sparse architectures, including Switch-style routing and expert specialization frameworks [49,52].

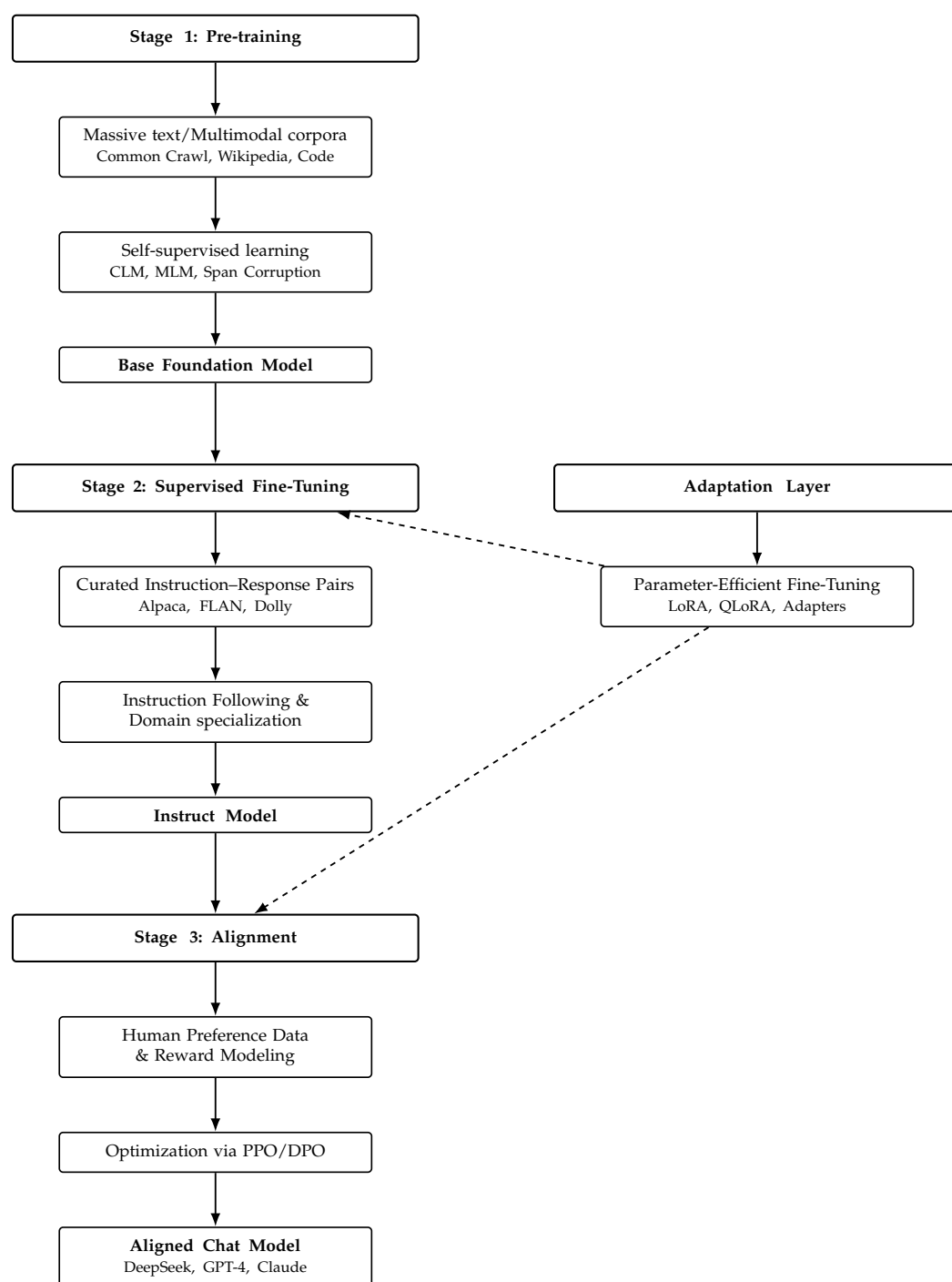


Figure 2. Multi-stage training pipeline of modern LLMs. All primary optimization stages follow a vertical progression, while parameter-efficient fine-tuning may be applied during post-training.

7. Evaluation of Large Language Models and Recent Advances

The evaluation of LLMs plays a central role in assessing their capabilities, robustness, and safety across a wide range of tasks. Standard benchmarks such as GLUE, SuperGLUE, MMLU, BIG-bench, HELM, and HumanEval are commonly employed to measure reasoning, language understanding, and code generation [6,58]. Task-specific datasets allow for assessment in specialized domains such as biomedical, legal, and cultural knowledge. Metrics such as accuracy, F1 score, BLEU, ROUGE [61], perplexity, and exact match are used depending on the nature of the task. In addition to performance metrics, alignment-related aspects such as instruction adherence, factual accuracy, bias, and toxicity are increasingly emphasized, particularly for models intended for real-world deployment. Careful dataset

curation along with qualitative evaluation helps to reveal weaknesses that standard metrics might overlook [58].

7.1. Outcome-Based vs. Process-Based Evaluation

Evaluation of modern LLMs is increasingly viewed along two complementary dimensions: (i) outcome-based benchmarks, which measure final-task correctness (e.g., MMLU, HumanEval), and (ii) process-based evaluation, which assesses the validity and faithfulness of intermediate reasoning steps.

Classical benchmark suites such as GLUE, SuperGLUE, MMLU, and BIG-bench primarily reflect outcome-level performance. However, recent reasoning-centric paradigms such as Reinforcement Learning with Verifiable Rewards (RLVR) motivate evaluation protocols that focus not only on whether an answer is correct but also on whether the reasoning process leading to it is verifiable and reliable [56].

A major evaluation challenge that has become increasingly emphasized in 2025–2026 is the problem of data contamination, also referred to as benchmark leakage [62]. Because modern LLMs are trained on web-scale corpora and large instruction-tuning datasets, it is possible for benchmark questions, prompts, or even reference solutions to unintentionally appear in the training data. In such cases, high scores on widely used benchmarks may partially reflect memorization rather than true generalization. This limitation directly affects the interpretability and validity of benchmark-based comparisons, including those summarized in Table 6, motivating the growing need for contamination-aware evaluation protocols and dataset auditing practices.

Table 6. Representative evaluation benchmarks for LLMs categorized by task type (general NLP, knowledge/reasoning, mixed capabilities, and code generation). We also report commonly used metrics and the intended evaluation scope, highlighting the limitations of benchmark-only evaluation without complementary safety/alignment and robustness protocols.

Benchmark	Task Type	Metrics	Notes	Refs
GLUE/SuperGLUE	NLP	Accuracy, F1	General language understanding benchmarks widely used for model comparison	[58]
MMLU	Knowledge/reasoning	Accuracy	Multiple-choice tasks across diverse academic and professional domains	[58]
BIG-bench	Mixed	Accuracy, F1, human eval	Large-scale benchmark covering broad emergent model capabilities	[58]
HumanEval	Code generation	Pass@k	Python (v3.8, as specified in the original benchmark release)	[6,58]
Domain-specific	Biomedical, Legal	Accuracy, BLEU	Specialized evaluation suites for expert-domain reliability testing	[58]

In addition to standard evaluations, human-centered and multi-dimensional approaches have emerged as a very important trend in the field. Human evaluators categorize the model's responses in terms of helpfulness, safety, and instruction-following character-

istics with a level of detail that is impossible for automatic metrics to achieve. Moreover, models are subjected to heavy testing by means of adversarial prompts, out-of-distribution scenarios, and complex reasoning tasks to measure their robustness and reliability. The adoption of all-inclusive frameworks that merge performance, alignment, and efficiency metrics is on the rise, as these frameworks are the only ones able to ensure that LLMs are both effective and safe for real-world applications [59].

These benchmarks show the different ways we evaluate large language models depending on what they are designed to do. General-purpose tests such as GLUE, SuperGLUE, and MMLU focus on language understanding and reasoning across a variety of topics, while BIG-bench takes things further with a wide mix of tasks to test how versatile a model really is. HumanEval looks specifically at coding skills by checking how well models can generate Python functions, while domain-specific benchmarks evaluate performance in specialized areas like biomedical or legal text. Altogether, these benchmarks provide a well-rounded picture of a model's strengths and weaknesses in both general and specialized settings.

Training-Aware Evaluation: SFT vs. RLHF Tradeoffs

Importantly, different post-training methodologies improve different evaluation dimensions. Supervised Fine-Tuning (SFT) primarily boosts performance on task-oriented benchmarks such as MMLU, GLUE, and HumanEval by directly optimizing for correctness on labeled instruction–response pairs. In contrast, alignment methods such as RLHF and DPO mainly improve human-centered metrics such as helpfulness, safety, and instruction adherence, which are often evaluated through preference benchmarks and red-teaming protocols [26,27,63]. However, stronger preference optimization may introduce an “alignment tax” in which gains in harmlessness come at the expense of reduced open-ended reasoning depth or general capability, highlighting the need for evaluation frameworks that jointly measure both accuracy and alignment [51,64]. This distinction clarifies why benchmark-only evaluation is insufficient without alignment-aware protocols.

Overall, evaluation of LLMs has gradually evolved from static task-oriented benchmarks towards more comprehensive and human-centered assessment frameworks. Traditional benchmarks remain essential for measuring core linguistic competence and reasoning capabilities, including suites such as GLUE, SuperGLUE, MMLU, and BIG-bench [58]. However, recent advances increasingly emphasize dimensions such as safety, alignment, robustness, and real-world applicability [65]. This transition reflects the growing need for evaluation methodologies that capture not only what LLMs can do but also how reliably and responsibly they behave in realistic deployment settings [63].

Recent research has shifted toward multi-dimensional, domain-specific, and human-centered evaluation paradigms [65]. For example, SAGE provides a modular framework for assessing LLM safety in multi-turn conversational scenarios [63], while MBias proposes strategies to mitigate bias without sacrificing contextual performance [66]. Other studies have investigated the faithfulness and reliability of toxicity explanations under human-aligned evaluation protocols [67]. In addition, enterprise-focused benchmarks and applied case studies have emerged to assess model behavior in practical organizational settings [68]. Collectively, these developments highlight that modern LLM evaluation is no longer limited to automated accuracy metrics but increasingly incorporates human feedback, robustness testing, and alignment-aware assessment [63,67].

Table 7 summarizes the main training paradigms used in contemporary LLM development, highlighting that state-of-the-art systems rely on a multi-stage pipeline rather than a single training method. Large-scale self-supervised pre-training provides broad linguistic and world knowledge but comes with high computational cost and does not

by itself ensure alignment or instruction-following behavior [1,5]. Supervised fine-tuning and instruction tuning adapt foundation models to downstream tasks and improve controllability, although they depend on high-quality labeled and instructional data [22,24]. Reinforcement Learning from Human Feedback (RLHF) further aligns model outputs with human preferences and safety constraints, though at the expense of increased complexity and human annotation effort [26,27]. Finally, Parameter-Efficient Fine-Tuning (PEFT) methods offer scalable alternatives for low-resource adaptation while maintaining competitive performance [37,39]. Overall, modern LLM training balances generalization, alignment, and efficiency by combining multiple complementary strategies across the full pipeline.

Table 7. Comparison of major LLM training and adaptation paradigms across the full pipeline. The table summarizes objectives, advantages, limitations, and representative model families for pre-training, SFT, instruction tuning, RLHF/DPO alignment, and PEFT, emphasizing the multi-stage nature of state-of-the-art LLM development.

Method	Description	Advantages	Disadvantages	Example Models	Refs
Pre-training (CLM, MLM, Span Corruption, UL2)	Self-supervised learning on massive text corpora	Learns general language/world knowledge; foundation for downstream tasks	Very high computational cost; large data requirements; may encode biases	GPT, BERT, LLaMA	[1,5,10,13]
Supervised Fine-Tuning (SFT)	Task-specific supervised training on labeled datasets	Improves task performance; better instruction-following	Limited generalization; expensive to curate large labeled datasets	FLAN-T5, Qwen	[22,23]
Instruction Tuning	Fine-tuning on datasets of instructions and responses	Enhances generalization to unseen tasks; improves adherence to instructions	Requires high-quality instruction datasets; computationally intensive	FLAN, Self-Instruct	[24,25]
Reinforcement Learning from Human Feedback (RLHF/DPO)	Optimizes model outputs using reward modeling and preference-based objectives	Aligns model behavior with human values; reduces harmful outputs	Complex pipeline; expensive; requires human labeling	ChatGPT-class systems; Claude-class systems	[26,27,38]
Parameter-Efficient Fine-Tuning (PEFT)	Updates only small subsets of parameters (e.g., LoRA, QLoRA)	Low memory and compute requirements; fast adaptation to new tasks	May underperform full fine-tuning; requires careful configuration	LLaMA-LoRA, QLoRA models	[37,39,41]

As summarized in the table, evaluation of LLMs has gradually shifted from purely task-centric benchmarks to more training-aware and alignment-aware frameworks. This shift reflects the growing understanding that benchmark scores alone cannot fully explain how training objectives, alignment strategies, and architectural choices influence model behavior, reasoning, and safety.

8. Analytical Taxonomy: Training Efficiency vs. Alignment and Reasoning Quality

While numerous surveys describe LLM training techniques in isolation, a central practical challenge is understanding the tradeoff landscape across the full training pipeline. To provide a differentiated comparative contribution beyond thematic grouping, we propose a unified analytical taxonomy that positions modern LLM approaches along two axes:

- (i) Training Efficiency, capturing the compute/parameter cost required to scale capability (e.g., dense vs. sparse activation, FLOPs efficiency, and parameter-efficient fine-tuning); and
- (ii) Alignment and Reasoning Quality, capturing how post-training improves instruction-following, helpfulness/safety, and increasingly verifiable reasoning performance.

As illustrated in Figure 3, contemporary LLM training approaches occupy distinct regions in a two-dimensional design space defined by training efficiency and alignment quality.

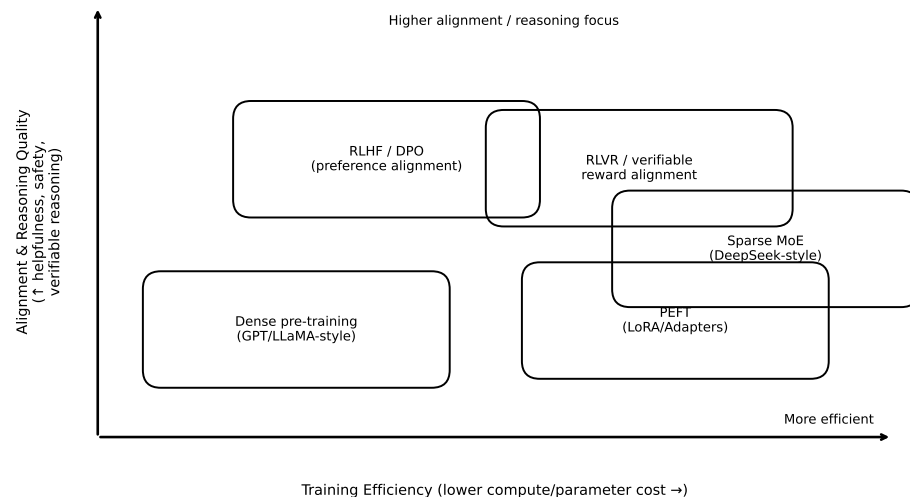


Figure 3. Proposed analytical taxonomy of LLM training methods along two axes: training efficiency (compute/parameter cost) versus alignment and reasoning quality. Arrows indicate the general trade-off direction between efficiency and alignment objectives.

8.1. DenseScaling vs. Sparse Efficiency

Dense transformer families (e.g., GPT-style and LLaMA-style models) primarily advance capabilities through parameter scaling and large-scale pre-training. In contrast, sparse Mixture-of-Experts (MoE) architectures (e.g., DeepSeek-style) activate only a subset of parameters per token, improving training and inference efficiency while maintaining competitive benchmark performance.

8.2. Alignment Optimization and the “Alignment Tax”

Post-training alignment methods such as RLHF and preference optimization (DPO) [27,38] can improve instruction compliance and safety, but may introduce an “alignment tax” in which overly constrained optimization degrades generality or reasoning depth [26,27]. Emerging approaches based on verifiable rewards (RLVR) [51,56] represent a shift toward scalable reasoning-centric alignment that targets correctness signals.

8.3. Why Does the Alignment Tax Emerge?

The alignment–capability tension is not merely conceptual but arises from concrete optimization dynamics in preference-based post-training. In RLHF-style pipelines, the model is optimized against a learned reward function trained to predict human preference judgments rather than ground-truth task correctness [26,27]. Because preference labels often reward properties such as politeness, caution, or harmlessness, they may only imperfectly correlate with objective reasoning accuracy. As reward optimization pressure increases, the policy is incentivized to maximize preference scores even when this entails producing shorter, more conservative, or less exploratory responses. In multi-step reasoning tasks, this may suppress intermediate reasoning chains or reduce willingness to attempt complex solutions. For example, on multi-step arithmetic benchmarks such as

GSM8K, a preference-optimized model may favor concise high-level answers that appear safe and confident rather than explicitly articulating intermediate reasoning steps, leading to reduced exact-match accuracy despite improved helpfulness ratings. Such dynamics can produce measurable regressions on capability-oriented benchmarks. Lin et al. [64] quantified this phenomenon by reporting performance declines on reasoning-intensive benchmarks such as GSM8K and MMLU as RLHF alignment strength increases.

Such effects motivate verifiable-reward approaches (RLVR), which ground optimization in objective correctness criteria rather than subjective preference modeling [51,56].

8.4. Mapping Paradigms to the Taxonomy

Table 8 summarizes how major training paradigms map onto this efficiency–alignment landscape. Overall, this taxonomy provides a concise comparative lens that connects foundational training pipelines to frontier reasoning-centric and efficiency-driven design choices.

Table 8. Mapping of major LLM training paradigms within the proposed efficiency–alignment taxonomy. Rows summarize how each paradigm primarily contributes to (i) training/inference efficiency and (ii) alignment and reasoning quality, providing a compact comparative lens beyond stage-by-stage descriptions.

Paradigm	Efficiency Focus	Alignment/Reasoning Focus	Refs
Dense pre-training (GPT/LLaMA-style)	High-compute scaling	General capability foundation	[5,9]
PEFT methods (LoRA, adapters, prompt-tuning)	Reduced fine-tuning cost	Task adaptation with minimal parameter drift	[37,39,41]
RLHF/DPO preference alignment	Moderate post-training overhead	Helpfulness, safety, preference shaping; potential trade-offs	[27,38,64]
Sparse MoE (DeepSeek-style)	High parameter/compute efficiency	Competitive reasoning with sparse activation	[48,49,52]
RLVR/verifiable reward alignment	Post-training efficiency	Scalable correctness-driven reasoning-centric alignment	[56]

9. Discussion

Despite their strong empirical performance, Large Language Models (LLMs) continue to face substantial limitations. Training and deploying state-of-the-art systems requires extensive computational resources, restricting accessibility and constraining real-world adoption. Persistent challenges related to hallucination, bias, and limited interpretability further complicate safe deployment in high-impact domains [8,26].

Recent research increasingly prioritizes efficiency during both training and inference. Parameter-Efficient Fine-Tuning (PEFT) methods such as LoRA and QLoRA offer a practical pathway for adapting large models to downstream tasks without prohibitive computational cost [37,39,41]. At the same time, advances in multilingual and multimodal training aim to improve robustness across diverse languages and input modalities [42,43]. Equally important are ongoing efforts to strengthen alignment mechanisms, ensuring that model behavior remains consistent with human intentions and safety requirements at scale through post-training optimization methods such as RLHF [26,27].

These limitations are clearly illustrated by recent reasoning-centric models such as DeepSeek, which serve as representative case studies of next-generation LLM training paradigms [48,50]. DeepSeek adopts a sparse Mixture-of-Experts (MoE) architecture, achieving strong performance at significantly reduced computational cost [48,49].

However, sparse expert activation complicates interpretability and expert specialization analysis [49,69]. Moreover, standard benchmark-based evaluation often fails to capture how different training objectives such as reasoning-oriented supervision, algorithmic feedback, or search-augmented optimization shape the internal representations learned by the model [58].

This mismatch highlights the growing need for training-aware evaluation frameworks that move beyond task-level performance metrics [58,63]. Models such as DeepSeek may achieve competitive benchmark scores while exhibiting substantially different internal information allocation, routing dynamics, and reasoning behavior compared to dense, instruction-tuned counterparts [69].

Training-aware representation analysis, such as fixed-dimensional embedding projections derived from LLMs (e.g., LLM2Vec-style approaches [70]), offers a promising complementary evaluation direction. By probing how different training regimes such as pre-training objectives, sparse MoE routing, instruction tuning, and reinforcement-based alignment shape latent representations, these methods enable a more granular and mechanistic understanding of model behavior [70]. In the context of sparsely activated reasoning-centric systems, embedding-level diagnostics can reveal differences that remain invisible to conventional benchmarks, supporting more reliable comparison, interpretability, and alignment verification [63,66].

Taken together, these observations suggest that future LLM research must jointly reconsider training strategies and evaluation methodologies, treating them as coupled design choices rather than independent components.

10. Future Directions

The design trajectory of future LLMs will likely involve an increasingly iterative integration of the core training stages, including foundation-scale pre-training, instruction tuning [24], and advanced post-training alignment techniques such as Reinforcement Learning from Human Feedback (RLHF) [26,27]. This multi-stage paradigm is expected to produce next-generation systems that maximize capability while maintaining robust safety and human-aligned behavior [22].

A major engineering priority will be the development of models that not only scale to extreme levels of complexity but also become more interpretable than today's largely opaque "black-box" systems [63]. Improved interpretability is essential for enabling developers, auditors, and regulators to better understand model decision-making in critical real-world deployments [66].

Alongside architectural advances, there is an urgent need to unify evaluation frameworks beyond static benchmark accuracy [58]. Modern assessment protocols must incorporate dimensions such as alignment with human values [63], mitigation of bias and fairness risks across diverse user groups [67], and robustness against adversarial or unexpected inputs [58]. Together, these technical and ethical directions will be essential to ensuring that future LLMs are reliable, trustworthy, and suitable for societal-scale integration.

11. Limitations

This systematic mapping study is subject to several limitations. First, the field of LLM training evolves extremely rapidly, and new architectures or alignment paradigms may emerge shortly after the temporal snapshot captured in this review (2017–2025). Second, although our search strategy covered major scholarly databases (Scopus, Web of Science, IEEE Xplore, and Google Scholar), this review may not fully capture specialized regional venues, industrial reports, or non-English publications.

Third, given the fast-moving nature of frontier AI research, a subset of influential works were available primarily as high-impact preprints at the time of writing. While their inclusion reflects current state-of-the-art developments, it also introduces variation in peer validation levels. Whenever available, peer-reviewed versions were prioritized for foundational methods, while recent frontier contributions are cited as preprints due to the rapid publication cycle of LLM research. Finally, the comparative synthesis presented in this paper emphasizes methodological categorization and representative paradigms rather than exhaustive completeness of the entire LLM literature. Future work may update this mapping as the field continues to evolve and additional peer-reviewed evidence becomes available.

12. Conclusions

This systematic mapping study has reviewed the evolving methodology of LLM training across the full pipeline, from foundation-scale pre-training objectives to post-training alignment and frontier efficiency-oriented architectures. Rather than a single monolithic procedure, modern LLM development has become a multi-stage process in which capability acquisition, adaptation, and behavioral control are jointly shaped by architectural design, optimization objectives, and evaluation constraints.

A central insight of this survey is that training strategies and evaluation practices cannot be treated as independent components. Design choices in data curation, fine-tuning, and alignment objectives directly determine not only benchmark outcomes but also model robustness, interpretability, and deployment safety. In particular, post-training alignment methods such as RLHF and DPO improve helpfulness and harmlessness, but also introduce critical tradeoffs—often described as the “alignment–capability tension” or “alignment tax”—where overly restrictive optimization may reduce general reasoning depth or performance on unconstrained tasks.

At the same time, the field is undergoing a clear shift towards efficiency-constrained and reasoning-centric paradigms. Parameter-Efficient Fine-Tuning (PEFT) techniques and sparse Mixture-of-Experts (MoE) architectures demonstrate that frontier-level performance is increasingly achievable without uniform dense scaling, highlighting a new axis of progress driven by activated parameter efficiency. Furthermore, emerging directions such as Reinforcement Learning with Verifiable Rewards (RLVR) suggest that future breakthroughs may rely less on subjective preference modeling and more on scalable correctness-driven reasoning objectives [51,56], particularly in the mathematics, coding, and STEM domains.

Looking forward, several open challenges remain. These include establishing unified training-aware evaluation standards, improving transparency in alignment objectives, reducing the computational barriers of large-scale post-training, and developing robust reasoning verification frameworks that generalize beyond narrow benchmarks. Addressing these challenges will be essential for building LLMs that are not only more capable but also efficient, trustworthy, and socially reliable.

Overall, the evolution of LLM training is moving from scale-driven capability acquisition toward a new frontier shaped by efficiency, alignment tradeoffs, and verifiable reasoning. We hope that the unified taxonomy and comparative evidence provided in this survey will support both researchers and practitioners in navigating this rapidly advancing methodological landscape.

Author Contributions: Conceptualization, D.K. and H.C.L.; methodology, D.K. and D.M.; software, D.K.; formal analysis, D.K. and D.M.; investigation, D.K.; resources, H.C.L.; writing—original draft preparation, D.K.; writing—review and editing, H.C.L. and D.M.; supervision, H.C.L.; project administration, H.C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Special Account for Research Grants of the University of West Attica.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors would like to thank the University of West Attica for the support provided during this research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
NLP	Natural Language Processing
CLM	Causal Language Modeling
MLM	Masked Language Modeling
SFT	Supervised Fine-Tuning
IFT	Instruction Fine-Tuning
RLHF	Reinforcement Learning from Human Feedback
DPO	Direct Preference Optimization
PEFT	Parameter-Efficient Fine-Tuning
LoRA	Low-Rank Adaptation
QLoRA	Quantized Low-Rank Adaptation
MoE	Mixture-of-Experts
RAG	Retrieval-Augmented Generation
MMLU	Massive Multitask Language Understanding
PII	Personally Identifiable Information

References

- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [\[CrossRef\]](#)
- Gisserot-Boukhlef, H.; Boizard, N.; Faysse, M.; Alves, D.M.; Malherbe, E.; Martins, A.F.T.; Hudelot, C.; Colombo, P. Should We Still Pretrain Encoders with Masked Language Modeling? *arXiv* **2025**, arXiv:2507.00994. [\[CrossRef\]](#)
- Interrante-Grant, A.; Varela-Rosa, C.; Narayan, S.; Connelly, C.; Reuther, A. Scaling Performance of Large Language Model Pretraining. *arXiv* **2025**, arXiv:2509.05258. [\[CrossRef\]](#)
- Penedo, G.; Malartic, Q.; Hesslow, D.; Launay, J.; Noun, H.; Pannier, B.; Cappelli, A.; Malartic, E. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *arXiv* **2023**, arXiv:2306.01116. [\[CrossRef\]](#)
- Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Online, 6–12 December 2020; NeurIPS: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
- Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating Large Language Models Trained on Code. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Online, 6–14 December 2021; NeurIPS: Red Hook, NY, USA, 2021; Volume 34, pp. 24963–24977.
- Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J.W.; Kreps, S.; et al. Release Strategies and the Social Impacts of Language Models. *arXiv* **2019**, arXiv:1908.09203. [\[CrossRef\]](#)
- Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv* **2022**, arXiv:2212.08073. [\[CrossRef\]](#)
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. [\[CrossRef\]](#)

10. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017*; NeurIPS: Red Hook, NY, USA, 2017; Volume 30, pp. 5998–6008.
12. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825. [[CrossRef](#)]
13. Tay, Y.; Dehghani, M.; Tran, V.Q.; Garcia, X.; Wei, J.; Wang, X.; Chung, H.W.; Bahri, D.; Bahri, T.; Metzler, D. UL2: Unifying Language Learning Paradigms. *arXiv* **2022**, arXiv:2205.05131. [[CrossRef](#)]
14. Kaplan, J.; McCandlish, S.; Hernandez, D.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv* **2020**, arXiv:2001.08361. [[CrossRef](#)]
15. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022*; NeurIPS: Red Hook, NY, USA, 2022.
16. Shoenybi, M.; Patwary, M.; Puri, R.; LeGresley, P.; Casper, J.; Catanzaro, B. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. *arXiv* **2019**, arXiv:1909.08053. [[CrossRef](#)]
17. Rajbhandari, S.; Rasley, J.; Ruwase, O.; He, Y. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. *arXiv* **2020**, arXiv:1910.02054. [[CrossRef](#)]
18. Kudo, T.; Richardson, J. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018*; NeurIPS: Red Hook, NY, USA, 2018; pp. 66–71. [[CrossRef](#)]
19. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Alpaca: A Strong, Replicable Instruction-Following Model. Available online: <https://crfm.stanford.edu/2023/03/13/alpaca.html> (accessed on 9 February 2026).
20. Köpf, A.; Kilcher, Y.; von Rütte, D.; Anagnostidis, S.; Tam, Z.; Stevens, K.; Barhoum, A.; Nguyen, D.; Stanley, O.; Nagyfi, R.; et al. OpenAssistant Conversations—Democratizing Large Language Model Alignment. *arXiv* **2023**, arXiv:2304.07327. [[CrossRef](#)]
21. Ding, N.; Chen, Y.; Xu, B.; Qin, Y.; Zheng, Z.; Hu, S.; Liu, Z.; Sun, M.; Zhou, B. Enhancing Chat Language Models by Scaling High-Quality Instructional Conversations. *arXiv* **2023**, arXiv:2305.14233. [[CrossRef](#)]
22. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. *arXiv* **2022**, arXiv:2210.11416. [[CrossRef](#)]
23. Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A.W.; Lester, B.; Du, N.; Dai, A.M.; Le, Q.V. Finetuned Language Models Are Zero-Shot Learners. *arXiv* **2021**, arXiv:2109.01652. [[CrossRef](#)]
24. Zhang, S.; Dong, L.; Li, X.; Zhang, S.; Sun, X.; Wang, S.; Li, J.; Hu, R.; Zhang, T.; Wu, F.; et al. Instruction Tuning for Large Language Models: A Survey. *arXiv* **2023**, arXiv:2308.10792. [[CrossRef](#)]
25. Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N.A.; Khashabi, D.; Hajishirzi, H. Self-Instruct: Aligning Language Models with Self-Generated Instructions. *arXiv* **2022**, arXiv:2212.10560. [[CrossRef](#)]
26. Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv* **2022**, arXiv:2204.05862. [[CrossRef](#)]
27. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022*; NeurIPS: Red Hook, NY, USA, 2022; Volume 35, pp. 27730–27744.
28. Lin, S.; Gao, L.; Oguz, B.; Xiong, W.; Lin, J.; Yih, W.T.; Chen, X. FLAME: Factuality-Aware Alignment for Large Language Models. *arXiv* **2024**, arXiv:2405.01525. [[CrossRef](#)]
29. Xu, Y.; Chakraborty, T.; Kiciman, E.; Aryal, B.; Rodrigues, E.; Sharma, S.; Estevao, R.; Balaguer, M.A.D.; Wolk, J.; Padilha, R.; et al. RLTHF: Targeted Human Feedback for LLM Alignment. *arXiv* **2025**, arXiv:2502.13417. [[CrossRef](#)]
30. Sotiropoulos, A.; Valapu, S.T.; Lei, L.; Coleman, J.; Krishnamachari, B. Crowd-SFT: Crowdsourcing for LLM Alignment. *arXiv* **2025**, arXiv:2506.04063. [[CrossRef](#)]
31. Li, M.; Chen, L.; Chen, J.; He, S.; Huang, H.; Gu, J.; Zhou, T. Reflection-Tuning: Data Recycling Improves LLM Instruction-Tuning. *arXiv* **2023**, arXiv:2310.11716. [[CrossRef](#)]
32. Pentyala, S.K.; Wang, Z.; Bi, B.; Ramnath, K.; Mao, X.-B.; Radhakrishnan, R.; Asur, S.; Cheng, N. PAFT: A Parallel Training Paradigm for Effective LLM Fine-Tuning. *arXiv* **2024**, arXiv:2406.17923. [[CrossRef](#)]
33. Chen, M.; Sun, L.; Li, T.; Sun, H.; Zhou, Y.; Zhu, C.; Wang, H.; Pan, J.Z.; Zhang, W.; Chen, H.; et al. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2503.19470. [[CrossRef](#)]
34. Tang, Y.; Cohen, T.; Zhang, D.W.; Valko, M.; Munos, R. RL-Finetuning LLMs from On- and Off-Policy Data with a Single Algorithm. *arXiv* **2025**, arXiv:2503.19612. [[CrossRef](#)]

35. Ye, K.; Zhou, H.; Zhu, J.; Quinzan, F.; Shi, C. Robust Reinforcement Learning from Human Feedback for Large Language Models Fine-Tuning. *arXiv* **2025**, arXiv:2504.03784. [[CrossRef](#)]
36. Zhou, Z.; Zhang, Q.; Kumbong, H.; Olukotun, K. LowRA: Accurate and Efficient LoRA Fine-Tuning of LLMs under 2 Bits. *arXiv* **2025**, arXiv:2502.08141. [[CrossRef](#)]
37. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
38. Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C.D.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
39. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023*; NeurIPS: Red Hook, NY, USA, 2023; Volume 36.
40. Qi, H.; Dai, Z.; Huang, C. Hybrid and Unitary PEFT for Resource-Efficient Large Language Models. *arXiv* **2025**, arXiv:2507.18076. [[CrossRef](#)]
41. Han, Z.; Gao, C.; Liu, J.; Zhang, J.; Zhang, S.Q. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *arXiv* **2024**, arXiv:2403.14608. [[CrossRef](#)]
42. Hu, C.; Wang, Y.; Xing, S.; Chen, C.; Feng, S.; Rossi, R.; Tu, Z. mRAG: Elucidating the Design Space of Multi-modal Retrieval-Augmented Generation. *arXiv* **2025**, arXiv:2505.24073. [[CrossRef](#)]
43. Drushchak, N.; Polyakovska, N.; Bautina, M.; Semenchenko, T.; Koscielecki, J.; Sykala, W.; Wegrzynowski, M. Multimodal Retrieval-Augmented Generation: Unified Information Processing Across Text, Image, Table, and Video Modalities. In *Proceedings of the 1st Workshop on Multimodal Augmented Generation via Multimodal Retrieval (MAGMaR 2025), Vienna, Austria, 1 August 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025.
44. Kumar, S.; Ghosal, T.; Goyal, V.; Ekbal, A. Can Large Language Models Unlock Novel Scientific Research Ideas? *arXiv* **2025**, arXiv:2409.06185. [[CrossRef](#)]
45. Zhu, Y.; Jiang, X.; Lin, J.; Chen, H.; Liu, Z.; Wang, Y.; Zhang, M. Large Language Models for Information Retrieval: A Survey. *ACM Trans. Inf. Syst.* **2026**, *44*, 1–54. [[CrossRef](#)]
46. Yue, Z.; Zhuang, H.; Bai, A.; Hui, K.; Jagerman, R.; Zeng, H.; Qin, Z.; Wang, D.; Wang, X.; Bendersky, M. Inference Scaling for Long-Context Retrieval Augmented Generation. *arXiv* **2025**, arXiv:2410.04343. [[CrossRef](#)]
47. Papageorgiou, G.; Sarlis, V.; Maragoudakis, M.; Tjortjis, C. A Multimodal Framework Embedding Retrieval-Augmented Generation with MLLMs for Eurobarometer Data. *AI* **2025**, *6*, 50. [[CrossRef](#)]
48. DeepSeek-AI; Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Deng, C.; Ruan, C.; Dai, D.; et al. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv* **2024**, arXiv:2405.04434. [[CrossRef](#)]
49. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le Q.; Hinton, G.; Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv* **2017**, arXiv:1701.06538. [[CrossRef](#)]
50. DeepSeek-AI; Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; et al. DeepSeek-V3 Technical Report. *arXiv* **2024**, arXiv:2412.19437. [[CrossRef](#)]
51. DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2501.12948. [[CrossRef](#)]
52. Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *J. Mach. Learn. Res.* **2022**, *23*, 1–39.
53. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The LLaMA 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783. [[CrossRef](#)]
54. Wang, Y.; Ren, S.; Lin, Z.; Han, Y.; Guo, H.; Yang, Z.; Zou, D.; Feng, J.; Liu, X. Qwen2.5: A Comprehensive Series of Large Language Models. *arXiv* **2024**, arXiv:2412.15119. [[CrossRef](#)]
55. Gumaan, E. ExpertRAG: Efficient RAG with Mixture of Experts. *arXiv* **2025**, arXiv:2504.08744. [[CrossRef](#)]
56. Wen, X.; Liu, Z.; Zheng, S.; Ye, S.; Wu, Z.; Wang, Y.; Xu, Z.; Liang, X.; Li, J.; Miao, Z.; et al. Reinforcement Learning with Verifiable Rewards Implicitly Incentivizes Correct Reasoning in Base LLMs. *arXiv* **2025**, arXiv:2506.14245. [[CrossRef](#)]
57. Liang, M.; Huang, W.; Liu, M.; Li, H.; Li, J. Lag-Relative Sparse Attention in Long Context Training. *arXiv* **2025**, arXiv:2506.11498. [[CrossRef](#)]
58. Ni, S.; Chen, G.; Li, S.; Chen, X.; Li, S.; Wang, B.; Wang, Q.; Wang, X.; Zhang, Y.; Fan, L.; et al. A Survey on Large Language Model Benchmarks. *arXiv* **2025**, arXiv:2508.15361. [[CrossRef](#)]
59. Datta, G.; Joshi, N.; Gupta, K. Analysis of Automatic Evaluation Metric on Low-Resourced Language: BERTScore vs BLEU Score. In *Speech and Computer; Lecture Notes in Computer Science (LNCS)*; Springer: Berlin/Heidelberg, Germany, 2022. [[CrossRef](#)]
60. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv* **2023**, arXiv:2302.04761. [[CrossRef](#)]

61. Chatoui, H.; Ata, O. Automated Evaluation of the Virtual Assistant in BLEU and ROUGE Scores. In *Proceedings of the IEEE HORA Conference, Ankara, Turkey, 11–13 June 2021*; IEEE: Piscataway, NJ, USA, 2021. [\[CrossRef\]](#)
62. Ganguli, D.; Askell, A.; Schiefer, N.; Liao, T.I.; Lukošiušė, K.; Chen, A.; Goldie, A.; Mirhoseini, A.; Olsson, C.; Hernandez, D.; et al. Do Large Language Models Know What They Know? On Data Contamination in Benchmarks. *arXiv* **2023**, arXiv:2302.07459. [\[CrossRef\]](#)
63. Jindal, M.; Shrawgi, H.; Agrawal, P.; Dandapat, S. SAGE: A Generic Framework for LLM Safety Evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, Suzhou, China, 4–9 November 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 11–33. [\[CrossRef\]](#)
64. Lin, Y.; Lin, H.; Xiong, W.; Diao, S.; Liu, J.; Zhang, J.; Pan, R.; Wang, H.; Hu, W.; Zhang, H.; et al. Mitigating the Alignment Tax of RLHF. *arXiv* **2023**, arXiv:2309.06256. [\[CrossRef\]](#)
65. Zhang, M.; Shen, Y.; Deng, J.; Wang, Y.; Zhang, Y.; Wang, J.; Liu, S.; Dou, S.; Sha, H.; Peng, Q.; et al. LLMEval-3: A Large-Scale Longitudinal Study on Robust and Fair Evaluation of Large Language Models. *arXiv* **2025**, arXiv:2508.05452. [\[CrossRef\]](#)
66. Raza, S.; Raval, A.; Chatrath, V. MBIAS: Mitigating Bias in Large Language Models While Retaining Context. *arXiv* **2024**, arXiv:2405.11290. [\[CrossRef\]](#)
67. Mothilal, R.K.; Roy, J.; Ahmed, S.I.; Guha, S. Human-Aligned Faithfulness in Toxicity Explanations. *arXiv* **2025**, arXiv:2506.19113. [\[CrossRef\]](#)
68. Kelsall, J.; Tan, X.; Bergin, A.; Chen, J.; Waheed, M.; Sorell, T.; Procter, R.; Liakata, M.; Chim, J.; Chi, S. Evaluating Large Language Models in Legal Use Cases. *AI Soc.* **2025**. [\[CrossRef\]](#)
69. Dai, D.; Deng, C.; Zhao, C.; Xu, R.X.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; et al. DeepSeekMoE: Towards Ultimate Expert Specialization. *arXiv* **2024**, arXiv:2401.06066. [\[CrossRef\]](#)
70. BehnamGhader, P.; Adlakha, V.; Mosbach, M.; Bahdanau, D.; Chapados, N.; Reddy, S. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. *arXiv* **2024**, arXiv:2404.05961. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.