Efficient Task Adaptation in Large Language Models: A Survey of Weight-Based, Prompt-Based, and Embedding-Based Adaptations

Jungwon Park[1,5], **Changin Choi**[2,6], **Jimyeong Kim**[3], **Nojun Kwak**[2,3,4], **Wonjong Rhee**[2,3,4]*
[1]RICS, [2]IPAI, [3]AIIS, [4]Department of Intelligence and Information,
Seoul National University
[5]Daegu Gyeongbuk Institute of Science and Technology
[6]Samsung Advanced Institute of Technology, Samsung Electronics Co., Ltd
{quoded97, ci2015.choi, wlaud1001, nojunk, wrhee}@snu.ac.kr

Abstract

As large language models are increasingly deployed across diverse downstream tasks, efficient task adaptation has emerged as a central challenge. In response, a wide range of task adaptation methods have been proposed, spanning parameter-efficient fine-tuning, in-context learning, and embedding-injection approaches. However, these lines of work have largely evolved within individual paradigms, leaving their cross-paradigm relationships and trade-offs underexplored, especially for recently emerging embedding-based adaptations. This survey presents a unified framework that categorizes task adaptation methods by where and how task information is encoded: model weights, input prompts, or injected task embeddings. We provide a comprehensive taxonomy that integrates these paradigms, analyze their key strengths and limitations to explain how different adaptation paradigms have evolved, clarify relationships across paradigms, and highlight open problems for future research. [1]

1 Introduction

Large language models (LLMs) are increasingly deployed across a wide range of downstream tasks (Brown et al., 2020; Wei et al., 2022a; Minaee et al., 2024; Raza et al., 2025), which makes efficient and effective task adaptation a central challenge for practical deployment. At the same time, as LLMs continue to scale in model size and training cost (Kaplan et al., 2020; Hoffmann et al., 2022; Achiam et al., 2023; OpenAI, 2025; Comanici et al., 2025), retraining or fully fine-tuning these models has become increasingly expensive and often infeasible, even when task-specific data are available. This growing gap between the demand for task adaptation and the cost of full fine-tuning has driven sustained interest in methods that reduce training cost, memory usage, and deployment overhead while maintaining strong task performance. Consequently, a wide range of approaches have been proposed, including methods that limit the number of trainable parameters or perform adaptation without any parameter updates. In this work, we focus on *efficient task adaptation*, which we define as adapting pretrained or instruction-tuned LLMs to downstream tasks without full fine-tuning.

Existing research has explored multiple paradigms for efficient task adaptation. Parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019; Lester et al., 2021; Hu et al., 2022; Liu et al., 2022a) adapts models by training a small number of task-specific parameters while keeping the backbone model largely frozen. In-context learning (ICL) (Brown et al., 2020; Liu et al., 2022b; Lu et al., 2022b; Zhou et al., 2022) enables training-free adaptation by specifying task information directly in prompts during inference. More recently, studies have shown that internal activations induced by ICL encode rich task-relevant information, giving rise to embedding-based adaptation that extracts and reuses explicit task representations during inference (Hendel et al., 2023; Todd et al., 2024). This line of work was originally developed in the context of mechanistic interpretability (Olah et al., 2020; Elhage et al., 2021), but has recently begun to be explored for practical deployment (Li et al., 2025d; Wang et al., 2025a; Park et al., 2026).

These approaches can be broadly categorized by where task information is encoded. We define three task adaptation paradigms from this perspective. *Weight-based adaptation* encodes task information in model weights, including additional or modified parameters introduced by weight-based PEFT. *Prompt-based adaptation* encodes task information in input prompts, such as task instructions or input-output demonstrations, without modifying the model. *Embedding-based adaptation* encodes

*Corresponding Author
[1]GitHub Repository: https://github.com/Jungwona/Awesome-Task-Adaptation-for-Large-Language-Models

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 600–631
November 6-10, 2026 ©2026 Association for Computational Linguistics

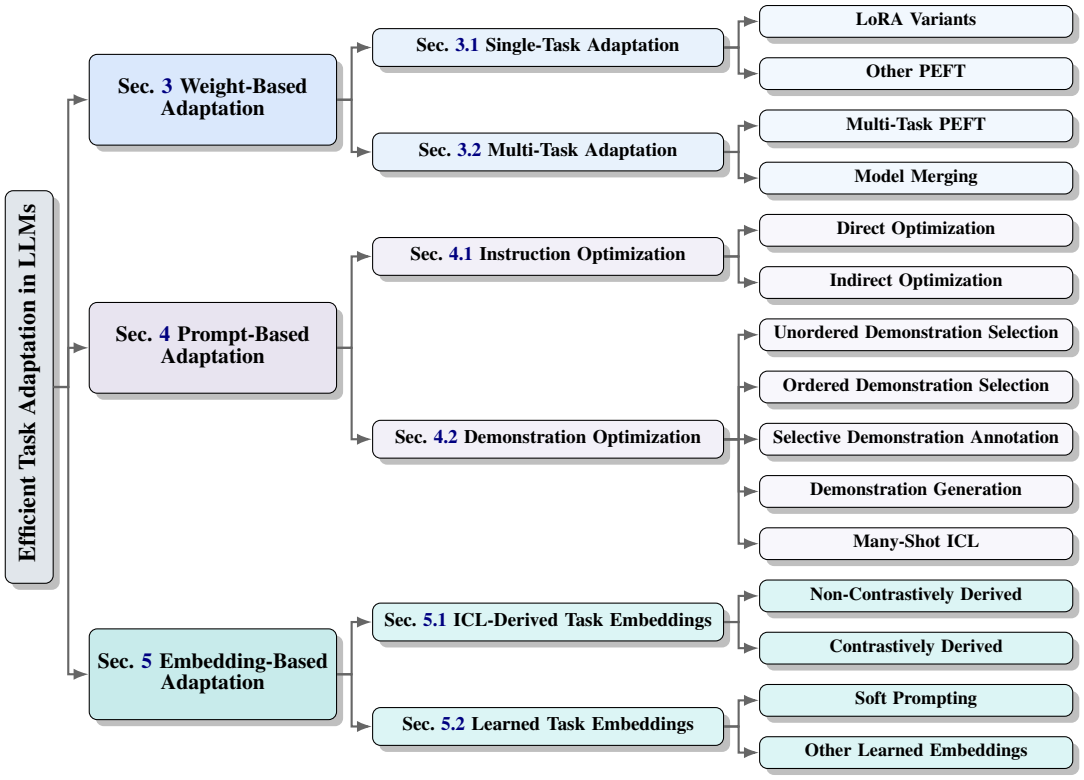


Figure 1: **Overview of the Taxonomy of Efficient Task Adaptation in Large Language Models.** The main taxonomy branches are shown, while the full taxonomy with individual methods is provided in Figure 3 of Appendix A.

task information in explicit task embeddings that are injected into model activations during inference, where such embeddings can be derived from ICL or learned through optimization.

Despite rapid progress, research on task adaptation has largely evolved independently within each paradigm, leaving the relationships among weight-based, prompt-based, and embedding-based methods underexplored. While prior PEFT-only and ICL-only surveys provide useful taxonomies within their respective scopes (Dong et al., 2024; Sahoo et al., 2024; Vatsal and Dubey, 2024; Han et al., 2024; Wang et al., 2025b; Mao et al., 2025), they do not jointly organize these paradigms under a common task-adaptation perspective, especially for emerging ICL-derived embedding-based methods.

This separation makes it difficult to see how different adaptation strategies relate to one another and how their high-level trade-offs differ in terms of model access, training requirements, inference overhead, parameter efficiency, and deployment constraints. To address this gap, we present a unified survey that organizes task adaptation methods for LLMs according to where and how task information is encoded. Figure 1 provides an overview of the main taxonomy branches, while the full taxonomy, including individual methods, is provided in Figure 3 of Appendix A. This survey makes the following contributions:

- We propose a unified taxonomy of LLM task adaptation methods based on where and how task information is encoded, and analyze their connections, strengths, and limitations.
- We provide the first dedicated survey that includes recently emerging embedding-based adaptations and clarify their relationships to ICL and PEFT.
- We identify open challenges and future research directions for task adaptation in LLMs.

2 A Unified View of Task Adaptation Paradigms

As defined in Section 1, we broadly categorize efficient task adaptation methods by where task information is encoded: model weights, input prompts, or explicit task embeddings injected into model activations. This perspective, illustrated in Figure 2, provides a useful lens for comparing the characteristics and goals of each paradigm.

Table 1 summarizes the key characteristics of these paradigms by describing where and how task

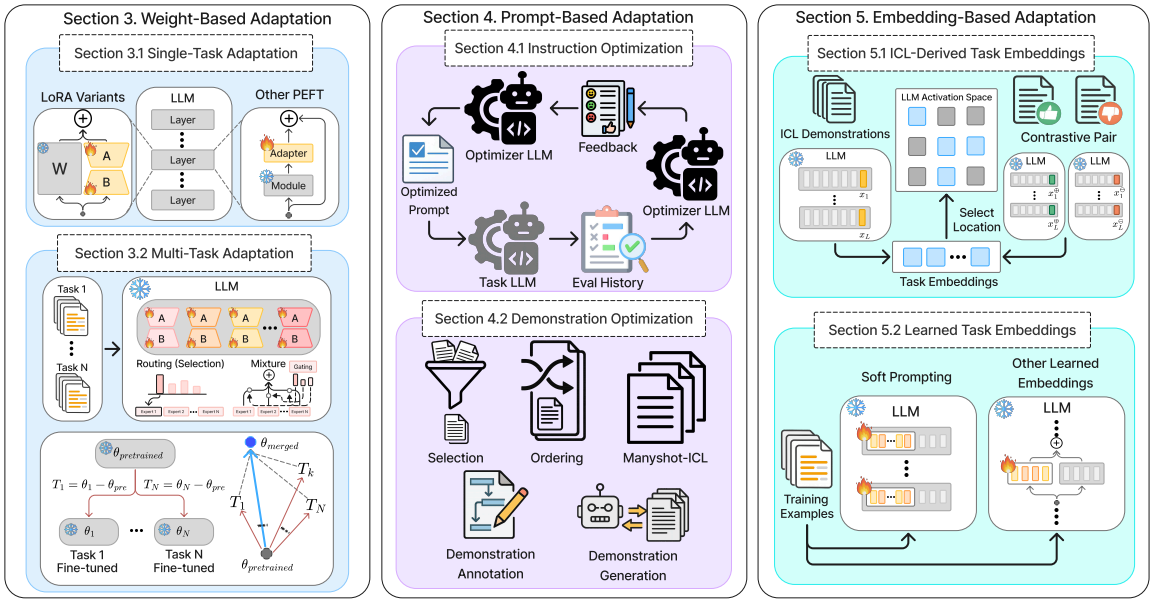


Figure 2: **Schematic Illustration of Three Task Adaptation Paradigms in LLMs.**

Adaptation Type	Task Information Encoding and Injection	Key Strengths	Key Limitations
Weight-Based (PEFT)	**Loc:** Task information encoded in model weights **Build:** Partial or additional weights trained **Inject:** Task-specific weights applied at inference	Strong and stable task performance Well established PEFT libraries Lower inference overhead than ICL	Requires training and parameter access Task-specific modules must be loaded Difficult mixed-task batching
Prompt-Based (ICL)	**Loc:** Task information encoded in prompts **Build:** Instructions or demonstrations in prompts **Inject:** Task-specific prompts used as input	Training-free adaptation Applicable to proprietary/API models Natural-language task specification	Repeated inference incurs optimization cost Long prompts increase deployment cost Sensitive to prompt formulation
Embedding-Based (ICL-derived)	**Loc:** Task information encoded in embeddings **Build:** Embeddings derived from demonstrations **Inject:** Embeddings injected into model activations	Compact task representations Lower inference overhead than ICL	Requires access to internal activations Requires task embedding construction
Embedding-Based (Learned)	**Loc:** Task information encoded in embeddings **Build:** Embeddings learned via optimization **Inject:** Embeddings injected into model activations	Compact task representations Lower inference overhead than ICL	Requires access to internal activations Optimization can be unstable Training may require many iterations

Table 1: **Conceptual Comparison of Task Adaptation Paradigms in LLMs.** The table compares weight-based, prompt-based, and embedding-based adaptation methods. It describes where and how task information is encoded and injected into the model, followed by their key strengths and limitations, reflecting common characteristics observed in the literature. The column *Task Information Encoding and Injection* specifies where task information is encoded (Loc), how it is constructed (Build), and how it is injected into the model during inference (Inject).

information is encoded and injected into the model, followed by their key strengths and limitations. The table reflects representative trends rather than strict properties of individual methods and provides high-level insights into the design differences and trade-offs among adaptation strategies. The following paragraphs further explain how these strengths and limitations are derived from the literature.

Weight-Based Adaptation. Prior work has shown that (weight-based) PEFT methods can achieve strong and stable performance (Mahabadi et al., 2021; Mosbach et al., 2023; Weyssow et al., 2025), often comparable to full fine-tuning (Hu et al., 2022; Liu et al., 2022a). In addition, well-established PEFT libraries are widely available for various models, making these methods easily accessible to practitioners (Mangrulkar et al., 2022). However, these methods require access to model parameters and involve training additional or partial weights, which incurs memory and computation costs during training. They may also introduce adapter-management overhead in serving systems, since task-specific modules must be dynamically loaded and scheduled across requests (Sheng et al., 2023; Chen et al., 2024a). In particular, mixed-task batching can be challenging when prompts within the same batch correspond to different tasks and require activating different task-specific modules. Serving such heterogeneous requests often requires specialized kernels, scheduling policies, and pipelined execution to mitigate latency (Wu et al., 2024a; Li et al., 2025b).

Prompt-Based Adaptation. Prompt-based adaptation interacts with the task-solving LLM only through its standard input-output interface and therefore does not require access to its parameters or internal activations. This makes it applicable to proprietary or API-only models such as GPT-5 (OpenAI, 2025) and GPT-6 (OpenAI, 2026). Natural-language prompts are also human-readable and readily editable, allowing users to directly specify and modify task instructions or demonstrations. Beyond manual prompt design, prompts can be optimized in a black-box manner, with an optimizer LLM deriving feedback from the task LLM's evaluation history and using it to iteratively refine the prompt (Zhou et al., 2022; Pryzant et al., 2023; Yang et al., 2024a). Recent work has extended this feedback-driven optimization framework beyond task instructions to agent skills, which encode reusable procedural knowledge and tool-use strategies as textual artifacts (Alzubi et al., 2026; Yang et al., 2026). However, prompt-based adaptation can incur substantial costs from repeated LLM inference during prompt optimization and from the additional prefill computation and KV-cache memory required by long prompts at deployment (Choi et al., 2023; Liskavets et al., 2025; Hooper et al., 2025). Furthermore, because prompts serve as the primary means of task specification, performance can vary substantially with instruction formulation and demonstration selection or ordering, potentially requiring careful prompt construction or iterative prompt optimization to achieve reliable performance (Zhou et al., 2022; Liu et al., 2022b; Lu et al., 2022b).

Embedding-Based Adaptation. Compared with weight-based approaches, embedding-based methods typically require far fewer parameters and do not modify model weights. For example, several representative embedding-based methods encode task information using roughly 0.1K–130K parameters for Llama-3.1-8B (Todd et al., 2024; Peng et al., 2024b; Huang et al., 2024; Li et al., 2025d; Park et al., 2026), whereas LoRA adapters for the same model require more than 3M parameters. However, both ICL-derived and learned embedding methods require access to internal model activations to extract or inject task embeddings. ICL-derived approaches typically require precomputing task embeddings and identifying appropriate injection locations for each task (Todd et al., 2024; Park et al., 2026). In contrast, learned embedding approaches, including soft prompting methods, can suffer from optimization instability and sensitivity to embedding initialization and hyperparameter settings (Lester et al., 2021; Li and Liang, 2021; Chen et al., 2022). In addition, recent learned embedding methods often require thousands to tens of thousands of training iterations to converge (Peng et al., 2024b; Saglam et al., 2025; Kang et al., 2025; Li et al., 2025a).

Overall, Table 1 highlights the key characteristics that distinguish these paradigms and provides insight into why different task adaptation strategies have emerged. In Sections 3–5, we examine representative methods within each paradigm and discuss how these methods have evolved in light of the perspective summarized in Table 1. As a supplementary reference, Appendix B.3 provides an illustrative quantitative snapshot in Table 2, based on values reported in prior work.

3 Weight-Based Adaptation

Weight-based adaptation encodes task-relevant information in model weights or additional trainable modules. In this section, we categorize weight-based methods into single-task and multi-task adaptations and describe their respective methodologies in detail. Single-task adaptation primarily focuses on improving training efficiency and task performance, while multi-task adaptation addresses challenges in switching tasks during inference or performing batched inference where prompts in the same batch correspond to different tasks.

3.1 Single-Task Adaptation

LoRA Variants. LoRA (Hu et al., 2022) introduces trainable low-rank updates to pretrained weight matrices. Due to its strong and stable performance, negligible inference overhead, and simple modular design, LoRA has inspired a large body of follow-up work that develops variants to improve expressivity, training dynamics, and efficiency. One line of work enhances the expressivity of model updates by increasing the intrinsic rank through alternative parametrizations, such as replacing standard matrix multiplication with Hadamard or Kronecker products (Nam et al., 2021; Edalati et al., 2022; Yeh et al., 2023; Huang et al., 2025). Another line focuses on improving training stability and convergence speed through better initialization strategies, such as leveraging singular value decomposition of pretrained weights or gradient matrices to initialize LoRA modules

more effectively (Meng et al., 2024; Wang et al., 2024a). A further line of work explores adaptive rank allocation, dynamically adjusting the effective rank across LoRA modules during training (Zhang et al., 2023b), or assigning different ranks to different modules based on their importance (Ding et al., 2023; Liu et al., 2024e). Other variants target training-time memory efficiency, particularly for large-scale LLMs, by combining LoRA with low-bit quantization of pretrained weights during training (Dettmers et al., 2023; Xu et al., 2024b; Guo et al., 2024a). Complementary approaches further reduce the number of trainable parameters or storage requirements through parameter sharing across LoRA modules (Kopiczko et al., 2024; Renduchintala et al., 2024; Li et al., 2024a), highlighting the flexibility of the LoRA framework.

Other PEFT. Beyond LoRA, a variety of PEFT methods have been proposed. Adapters (Houlsby et al., 2019) insert small trainable modules between Transformer layers of a pretrained model, while Compacter (Mahabadi et al., 2021) reduces adapter parameters by replacing adapter linear layers with low-rank hypercomplex multiplication. Other approaches reduce trainable parameters by selectively updating subsets of model weights. For example, FISH Mask (Sung et al., 2021) updates parameters based on their approximate Fisher Information (Fisher, 1922; Amari, 1996; Kirkpatrick et al., 2017), and BitFit (Zaken et al., 2022) fine-tunes only bias terms. Another line of work applies element-wise rescaling operations to internal activations. $(IA)^3$ (Liu et al., 2022a) introduces learnable scaling vectors that rescales internal activations, RED (Wu et al., 2024b) extends this idea by jointly learning scaling and bias vectors, and PARA (Liu et al., 2024f) trains a lightweight vector generator to dynamically produce scaling vectors. Some methods combine or extend PEFT techniques, such as DePT (Shi and Lipani, 2024), which integrates prompt tuning with LoRA-style updates on frozen word embeddings, and its variant ADePT (Tang et al., 2025a). Other approaches focus on improving efficiency. LST (Sung et al., 2022) trains a small auxiliary network to reduce training-time memory requirements by avoiding backpropagation through the backbone model, and RoAD (Liao and Monz, 2024) applies trainable rotations to linear layer outputs to reduce the trainable parameters.

3.2 Multi-Task Adaptation

Multi-Task PEFT. Compared to prompt-based and embedding-based adaptation, weight-based methods typically incur higher overhead when switching tasks during inference or processing mixed-task batches, as task-specific modules must be dynamically loaded and scheduled across requests. To mitigate this limitation, several methods introduce routing-based mechanisms that maintain multiple adapters and use (soft) routers to select or combine task-relevant adapters, either at the task level (Pfeiffer et al., 2021; Huang et al., 2023; Wang et al., 2023a) or at the instance level (Wang et al., 2023d; Feng et al., 2024; Tian et al., 2024; Xu et al., 2024a; Liu et al., 2025). These routers are typically implemented using learnable scalar weights or shallow neural networks. Other approaches explore alternative strategies for multi-task inference. O-LoRA (Wang et al., 2023b) proposes a LoRA-based continual learning framework, while FLoRA (Wen and Chaudhuri, 2024) replaces LoRA updates with Hadamard-product-based updates, enabling efficient heterogeneous batching by avoiding expensive batched matrix multiplications.

Model Merging. Model merging defines *task vectors* (Ilharco et al., 2023) as parameter differences between fine-tuned models and a shared pretrained model and combines multiple such vectors to enable multi-task inference without additional fine-tuning. Although model merging typically underperforms multi-task training, it is particularly useful when fine-tuned models are available but the corresponding training data cannot be accessed, for example due to data privacy or intellectual property constraints. Several methods aim to reduce interference between task-specific parameter updates during model merging (Yadav et al., 2023b; Yu et al., 2024; Gargiulo et al., 2025), while others propose fine-grained merging strategies that assign different merging coefficients across tasks, layers, or parameters (Yang et al., 2024b; Zhou et al., 2024b; Xu et al., 2025; Yao et al., 2025; Lee et al., 2025). Another line of work focuses on performing model merging only on PEFT updates (Zhang et al., 2023a; Tang et al., 2023; Zhao et al., 2024). Among these, LoRA-LEGO (Zhao et al., 2024) introduces rank-wise LoRA merging by identifying row-column pairs of LoRA matrices as minimal semantic units, clustering them across tasks, and using cluster centroids to construct merged adapters.

4 Prompt-Based Adaptation

Prompt-based adaptation, commonly referred to as in-context learning (ICL), adapts LLMs by providing task-relevant information directly in the input prompt at inference time, in the form of natural-language instructions and/or input-output demonstrations. Existing methods can be broadly categorized into *instruction optimization*, which focuses on refining task instructions, and *demonstration optimization*, which selects, orders, or generates input-output demonstrations. Collectively, these methods primarily aim to improve stability and performance of prompt-based adaptation.

4.1 Instruction Optimization

Instruction optimization focuses on refining task-specific instructions used in the prompt. These methods typically adopt a two-LLM framework, consisting of an *optimizer LLM* that updates instructions and a *task LLM* that performs the downstream task using the updated instructions. In practice, strong proprietary models such as GPT-4o (Hurst et al., 2024) are often used as the optimizer LLM to enable more effective instruction optimization. A common strategy is to iteratively refine instructions using feedback generated by the optimizer LLM itself (Pryzant et al., 2023; Wang et al., 2024b; Agarwal et al., 2025; Juneja et al., 2025; Yan et al., 2025). For example, ProTeGi (Pryzant et al., 2023) uses the optimizer LLM to analyze errors made under the current instruction and feeds the resulting feedback back into the optimizer LLM to update the instructions. Other approaches formulate instruction optimization as an evolutionary process, iteratively improving instructions through mutation, selection, or related evolutionary operations (Guo et al., 2024c; Fernando et al., 2024; Agrawal et al., 2026). For example, GEPA (Agrawal et al., 2026) combines reflective feedback with evolutionary search, using natural-language reflection to propose prompt updates and Pareto-based selection to retain promising candidates. Several methods leverage carefully designed meta-prompts, i.e., prompts that instruct the optimizer LLM how to refine instructions, to improve optimization stability and effectiveness (Ye et al., 2024; Tang et al., 2025b). In contrast, some approaches avoid directly optimizing instructions and instead optimize auxiliary components, such as soft prompts (Lester et al., 2021) provided to the optimizer LLM or the optimizer LLM itself (Chen et al., 2024b; Lin et al., 2024; Kwon et al., 2024; Zhao et al., 2025). For instance, InstructZero (Chen et al., 2024b) and INSTINCT (Lin et al., 2024) optimize only soft prompts, while StablePrompt (Kwon et al., 2024) fine-tunes the optimizer LLM to improve instruction generation.

Beyond the optimization of task instructions, recent work on LLM agents has explored the automatic construction and refinement of *agent skills*, which are reusable artifacts that encode procedural knowledge and tool-use strategies for a particular task family or domain (Mi et al., 2026; Alzubi et al., 2026; Ni et al., 2026; Yang et al., 2026). From the perspective of prompt-based adaptation, methods that refine the textual content of agent skills are closely related to instruction optimization: both adapt the task LLM without updating its parameters by modifying natural-language content provided in the input prompt, often using feedback from black-box task evaluations. The distinction between them therefore lies primarily in the type of content being optimized: instruction optimization refines instructions specifying a task's objective, constraints, expected output, or solution strategy, whereas agent-skill methods construct or refine reusable procedural knowledge that guides how an agent acts over multiple steps, interacts with tools, and responds to intermediate outcomes.

4.2 Demonstration Optimization

Including a small number of input-output demonstrations in the prompt can effectively convey task information to LLMs (Brown et al., 2020). However, task performance is highly sensitive to both which demonstrations are selected and how they are ordered within the prompt (Zhao et al., 2021; Liu et al., 2022b; Lu et al., 2022b). As a result, a large body of work has explored methods for optimizing demonstration selection and ordering. KATE (Liu et al., 2022b) retrieves demonstrations by embedding candidate examples and selecting nearest neighbors for each test input, and subsequent methods such as MDL (Wu et al., 2023) and ConE (Peng et al., 2024a) adopt this retrieval step to first narrow the candidate pool before applying more refined selection strategies. Another line of work explicitly promotes diversity among selected demonstrations using determinantal point processes (DPPs) (Kulesza et al., 2012; Ye et al., 2023; Yang et al., 2023). Beyond selecting which demonstrations to include, several methods also consider demonstration ordering. Some approaches

treat an ordered sequence of demonstrations as the basic unit of selection rather than scoring demonstrations independently (Lu et al., 2022b; Wu et al., 2023, 2024e; Guo et al., 2024b), while others formulate demonstration selection as a sequential decision-making process optimized with reinforcement learning (Zhang et al., 2022; Scarlatos and Lan, 2023; Wang et al., 2025e). For example, AES (Zhang et al., 2022) formulates demonstration selection as a Markov Decision Process, where the policy state consists of all previously selected demonstrations, and trains an offline Q-learning policy (Mnih et al., 2013) to select each new demonstration conditioned on this history.

In settings where only unlabeled data are available, several methods selectively annotate demonstrations to support in-context learning (Su et al., 2022; Mavromatis et al., 2023; Zhang et al., 2024b). For example, Vote-K (Su et al., 2022) annotates diverse unlabeled examples using a k-nearest neighbor graph constructed in the Sentence-BERT (Reimers and Gurevych, 2019) embedding space, while IDEAL (Zhang et al., 2024b) extends this approach with influence-driven selection to better approximate the underlying data distribution. More recently, LLM-generated demonstrations have emerged as an effective alternative to human-curated examples (Long et al., 2024; Yehudai et al., 2024; Nadas et al., 2025). Some approaches generate synthetic input-output pairs directly (Chen et al., 2023; Su et al., 2024; Wan et al., 2025), while others augment demonstrations with LLM-generated reasoning paths (Shao et al., 2023; He et al., 2024b; Honda and Oka, 2025), enabling chain-of-thought (CoT) (Wei et al., 2022b) reasoning during inference. When a large number of demonstrations are available, scaling the number of demonstrations from a few to many, a setting commonly referred to as *many-shot ICL*, can further improve task performance and robustness to demonstration selection and ordering (Zhang et al., 2025a; Bertsch et al., 2025). However, these gains come at the cost of increased memory and computation that scale with the number of demonstrations, as well as attention dispersion (Yuan et al., 2024) in long prompts, which can degrade performance.

5 Embedding-Based Adaptation

Embedding-based adaptation represents task information as explicit vectors, referred to as *task embeddings*, which are injected into model activations during inference, typically via addition or concatenation. Existing methods can be broadly divided into two classes based on how these embeddings are obtained: *ICL-derived task embeddings* and *learned task embeddings*. The former extracts task embeddings from internal activations induced by ICL, while the latter directly optimizes task embeddings through gradient-based training. Although these two classes are respectively related to ICL and PEFT, both aim to improve downstream task performance without relying on task-specific prompts during inference.

5.1 ICL-Derived Task Embeddings

Non-Contrastively Derived. Early work, notably Task Vectors (Hendel et al., 2023) and Function Vectors (Todd et al., 2024), showed that last-token internal activations produced during few-shot inference encode rich task-relevant information, which can be extracted and injected into model activations to enable task execution in a zero-shot setting. Building on this observation, non-contrastive embedding-based adaptation methods aggregate such activations across multiple few-shot prompts and aim to identify effective injection locations that maximize task performance. Injection locations are determined using various strategies, including validation-based sweeps over layers (Hendel et al., 2023; Zhang et al., 2024a; Wang et al., 2025a), reinforcement-learning-based optimization (Huang et al., 2024; Cai et al., 2025), and gradient-based optimization that softly controls task injection (Li et al., 2025d; Park et al., 2026). While most approaches operate at the layer level, some explore finer-grained injection at the attention-head level (Huang et al., 2024; Park et al., 2026). Recent results indicate that these methods can achieve performance comparable to few-shot ICL; however, evaluations have largely focused on relatively simple tasks, motivating broader evaluation on more complex reasoning and generation tasks.

Contrastively Derived. Contrastively derived task embeddings originate from Inference-Time Intervention (ITI) (Li et al., 2023b), which showed that differences between last-token activations induced by contrastive prompt pairs encode directional information that can be injected into model activations to steer model behavior. ITI and subsequent work have leveraged such contrastive prompt pairs primarily for behavior steering, including improving truthfulness or safety (Liu et al., 2024b;

Rimsky et al., 2024; Qiu et al., 2024; Wang et al., 2025c; Zhan et al., 2026). More recent methods extend this paradigm to task adaptation by constructing contrastive pairs where one prompt contains correct task demonstrations and the other contains incorrect or missing demonstrations, enabling task execution via the resulting difference vectors (Wang et al., 2025d; Liu and Deng, 2025). Using contrastively derived embeddings for general task adaptation is relatively underexplored, and systematic comparisons with non-contrastive approaches remain an open area for further study.

5.2 Learned Task Embeddings

Soft Prompting. Prompt Tuning (Lester et al., 2021) adapts models by prepending learnable continuous embeddings, known as *soft prompts*, to the input embedding sequence and optimizing only these embeddings. Although highly parameter-efficient, prompt tuning is often unstable and typically underperforms weight-based PEFT methods such as LoRA (Hu et al., 2022; Liu et al., 2022a), motivating subsequent improvements. Some approaches dynamically adapt task-specific soft prompts to produce instance-specific soft prompts, leading to improved performance (Asai et al., 2022; Jain et al., 2024). Another line of work explores transfer learning by leveraging soft prompts from multiple source tasks to initialize or compose target-task prompts, thereby transferring task knowledge and improving adaptation (Vu et al., 2022; Asai et al., 2022; Wang et al., 2023f; Belanec et al., 2025; Zhang et al., 2025b). For example, SPoT (Vu et al., 2022), MPT (Wang et al., 2023f), and Task Prompt Vectors (Belanec et al., 2025) initialize target-task soft prompts using soft prompts learned from related source tasks.

Other Learned Embeddings. Other learned-embedding approaches adapt models by optimizing task embeddings via gradient-based training, where the embeddings are injected into internal activations, typically via additive intervention. LIVE (Peng et al., 2024b) and M^2IV (Li et al., 2025c) introduce layer-wise learnable vectors with associated scaling factors that are trained end-to-end. LTV (Saglam et al., 2025) extracts attention-head outputs from few-shot inference and constructs layer-wise task embeddings as learned weighted combinations of heads within each layer. However, these methods often require a large number of training iterations, sometimes exceeding those of standard PEFT approaches (Peng et al., 2024b; Saglam et al., 2025; Kang et al., 2025; Li et al., 2025a). Moreover, these methods are often compared with ICL-derived embedding approaches, while their conceptual similarity to soft prompting methods is often overlooked.

6 Open Problems and Discussion

In this section, we present four open research problems that cut across all three task adaptation paradigms for LLMs. Additional discussions are provided in Appendix B, including comparisons between our taxonomy and prior PEFT-only or ICL-only survey taxonomies, the distinction between weight-based and embedding-based adaptations, the rationale for classifying soft prompting as embedding-based adaptation, and a limited quantitative analysis.

Hybrid and Compositional Task Adaptation. Beyond relying on a single adaptation paradigm, an important research problem is to jointly leverage multiple forms of task information, such as learned embeddings, prompts, and trained modules, to strengthen task adaptation. As a representative example, Instruction Prompt Tuning (Singhal et al., 2023) prepends a shared soft prompt to task-specific instructions and demonstrations, enabling complementary task signals to jointly guide model behavior and better align an instruction-tuned LLM with domain-specific instruction semantics, thereby improving the safety, grounding, and completeness of long-form generations. Despite such initial efforts, how best to combine different task adaptation paradigms remains underexplored, presenting an important direction for future research.

Data Efficiency and Synthetic Supervision. Most task adaptation methods require a nontrivial number of labeled demonstrations to achieve strong performance (Lester et al., 2021; Hu et al., 2022; Agarwal et al., 2024). To reduce reliance on human-curated supervision, LLM-generated data have emerged as a scalable alternative and are increasingly adopted for task adaptation (Long et al., 2024; Yehudai et al., 2024; Nadas et al., 2025). Prior work shows that synthetic demonstrations can differ from human-curated ones in fidelity, bias, and diversity (Li et al., 2024b; Nadas et al., 2025), yet how these differences affect different task adaptation paradigms has not been thoroughly studied. Addressing this question could provide useful in-

sights into how synthetic data should be used across different task adaptation paradigms.

Benchmarks for Evaluating Task Adaptation. Existing task adaptation methods have been evaluated on a broad range of established benchmarks, summarized in Tables 3-6. These benchmarks cover diverse categories, including natural language understanding, reasoning, mathematics, coding, summarization, and safety and trustworthiness. While such benchmarks have served as controlled testbeds for comparing adaptation methods, recent developments in LLM research suggest the need to expand evaluations to more challenging settings. These include generating long-form responses (Wu et al., 2024d; Bai et al., 2024; Que et al., 2024), solving complex multi-step reasoning problems (He et al., 2024a; Glazer et al., 2024; Phan et al., 2025; Art of Problem Solving, 2025; Lin et al., 2025; Balunović et al., 2025), and maintaining coherent behavior in long-horizon or interactive environments (Shridhar et al., 2020; Wang et al., 2022; Yao et al., 2022; Zhou et al., 2023; Wei et al., 2025). Evaluating adaptation methods under such settings may provide deeper insights into their robustness, error accumulation, and long-range reasoning capabilities.

Beyond Task Adaptation: Encoding Behavioral and Contextual Information. Beyond task adaptation, these techniques have also been explored for encoding diverse behavioral and contextual signals in LLMs, including safety/truthfulness, persona, and long-context knowledge. Prompting offers a simple and flexible interface for specifying such signals in natural language, but its effectiveness can be limited for some attributes (e.g., safety or persona) (Zheng et al., 2024a,b), and long prompts may introduce additional inference overhead. To address these limitations, recent methods encode such information directly into model parameters or embeddings. For example, SHINE (Liu et al., 2026) and Doc-to-LoRA (Charakorn et al., 2026) generate LoRA adapters that internalize information from long-contexts. More broadly, embedding-based approaches have been used to encode various types of information, including safety/truthfulness (Li et al., 2023b; Wang et al., 2025c), persona (Potertì et al., 2025; Feng et al., 2026), and long-context knowledge (Eyuboglu et al., 2025), suggesting that expanding the range of information encoded through such adaptation techniques is an interesting research direction.

7 Conclusion

This survey presents a unified overview of task adaptation in LLMs by organizing existing methods according to where task information is encoded. Through this framework, we connect weight-based, prompt-based, and embedding-based adaptations and discuss their relationships, trade-offs, and shared open problems. We hope this survey serves as a useful reference for understanding the independently evolving landscape of LLM task adaptation.

Limitations

This survey provides a unified overview of task adaptation techniques for large language models, but it has several limitations. First, given the rapid pace of recent progress in this area, the taxonomy and comparisons presented here may not fully capture the most recent developments. Second, the survey emphasizes high-level methodological distinctions and qualitative trade-offs rather than exhaustive empirical re-evaluation of individual methods. This focus is intended to highlight the key strengths and limitations of different adaptation paradigms, thereby contextualizing existing approaches and motivating future research. Finally, although many task adaptation techniques for LLMs may be applicable to other types of models, such as multimodal LLMs or visual generative models, this survey focuses exclusively on text-based LLMs, leaving such extensions for future work.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) ([NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)], [No.RS-2023-00235293, Development of autonomous driving big data processing, management, search, and sharing interface technology to provide autonomous driving data according to the purpose of usage]) and the InnoCORE program of the Ministry of Science and ICT (26-InnoCORE-01).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Eshaan Agarwal, Raghav Magazine, Joykirat Singh, Vivek Dani, Tanuja Ganu, and Akshay Nambi. 2025. Promptwizard: Optimizing prompts via task-aware, feedback-driven self-evolution. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19974–20003.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, and 1 others. 2024. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328.

Ahmed Agiza, Marina Neseem, and Sherief Reda. 2024. Mtlora: Low-rank adaptation approach for efficient multi-task learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16196–16205.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, and 1 others. 2026. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pages 8479–8565.

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. 2026. Evoskill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*.

Shun-ichi Amari. 1996. Neural learning in structured parameter spaces-natural riemannian gradient. *Advances in neural information processing systems*, 9.

Art of Problem Solving. 2025. 2025 aime i. https://artofproblemsolving.com/wiki/index.php/2025_AIME_I. Accessed: 2025.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 6655–6672.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. 2024. Lora-xs: Low-rank adaptation with extremely small number of parameters. *arXiv preprint arXiv:2405.17604*.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*.

Robert Belanec, Simon Ostermann, Ivan Srba, and Maria Bielikova. 2025. Task prompt vectors: Effective initialization through multi-task soft prompt transfer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 77–94. Springer.

Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. 2025. In-context learning with long-context models: An in-depth exploration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12119–12149.

Kartikeya Bhardwaj, Nilesh Prasad Pandey, Sweta Priyadarshi, Viswanath Ganapathy, Shreya Kadambi, Rafael Esteves, Shubhankar Borse, Paul Whatmough, Risheek Garrepalli, Mart Van Baalen, Harris Teague, and Markus Nagel. 2024. Sparse high rank adapters. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 632–642.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wang Cai, Hsiu-Yuan Huang, Zhixiang Wang, and Yunfang Wu. 2025. Beyond demonstrations: Dynamic vector construction from latent representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5853–5868.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, and 1 others. 2023. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691.

Rujikorn Charakorn, Edoardo Cetin, Shinnosuke Uesaka, and Robert Tjarko Lange. 2026. Doc-to-lora: Learning to instantly internalize contexts. *arXiv preprint arXiv:2602.15902*.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. 2024a. Punica: Multi-tenant lora serving. *Proceedings of Machine Learning and Systems*, 6:1–13.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2024b. InstructZero: Efficient instruction optimization for black-box large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6503–6518. PMLR.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021a. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. Self-icl: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021b. Dialogsum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337.

Eunbi Choi, Yongrae Jo, Joel Jang, Joonwon Jang, and Minjoon Seo. 2023. Fixed input parameterization for efficient prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8428–8441.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Sarkar Snigdha Sarathi Das, Ryo Kamoi, Bo Pang, Yusen Zhang, Caiming Xiong, and Rui Zhang. 2025. GReater: Gradients over reasoning makes smaller language models strong prompt optimizers. In *The Thirteenth International Conference on Learning Representations*.

Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 4133–4145.

Xuan Long Do, Yiran Zhao, Hannah Brown, Yuxi Xie, James Xu Zhao, Nancy Chen, Kenji Kawaguchi, Michael Shieh, and Junxian He. 2024. Prompt optimization via adversarial in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7308–7327.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, and 1 others. 2024. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 1107–1128.

Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. 2022. Krona: Parameter efficient tuning with kronecker adapter. *Preprint*, arXiv:2212.10650.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and 1 others.

2025. Cartridges: Lightweight and general-purpose long context representations via self-study. *arXiv preprint arXiv:2506.06266*.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-loras: An efficient multitask tuning for large language models. *arXiv preprint arXiv:2403.03432*.

Xiachong Feng, Liang Zhao, Weihong Zhong, Yichong Huang, Yuxuan Gu, Lingpeng Kong, Xiaocheng Feng, and Bing Qin. 2026. Persona: Dynamic and compositional inference-time personality control via activation vector algebra. *arXiv preprint arXiv:2602.15669*.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Ronald A Fisher. 1922. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. 2024. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *10th International Conference on Natural Language Generation*, pages 124–133. ACL Anthology.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. 2025. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18695–18705.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. 2024. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. 2024a. LQ-loRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *The Twelfth International Conference on Learning Representations*.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024b. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024c. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *Preprint*, arXiv:2212.06713.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024a. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

Xuanli He, Yuxiang Wu, Oana-Maria Camburu, Pasquale Minervini, and Pontus Stenetorp. 2024b. Using natural language explanations to improve robustness of in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13477–13499.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Ukyo Honda and Tatsushi Oka. 2025. Exploring explanations improves the robustness of in-context learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23693–23714.

Coleman Richard Charles Hooper, Sehoon Kim, Hiva Mohammadzadeh, Monishwaran Maheswaran, Sebastian Zhao, June Paik, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2025. Squeezed attention: Accelerating long context length llm inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32631–32652.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. 2024. Localized zeroth-order prompt optimization. *Advances in Neural Information Processing Systems*, 37:86309–86345.

Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei Herzig. 2024. Multimodal task vectors enable many-shot multimodal in-context learning. *Advances in Neural Information Processing Systems*, 37:22124–22153.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. 2025. HiRA: Parameter-efficient hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations*.

Hugging Face. 2026. Trending papers.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Leonardo Iurada, Marco Ciccone, and Tatiana Tommasi. 2025. Efficient model editing with task-localized sparse fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Abhinav Jain, Swarat Chaudhuri, Thomas Reps, and Chris Jermaine. 2024. Prompt tuning strikes back: Customizing foundation models with low-rank prompt adaptation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yash Jain and Vishal Chowdhary. 2025. Local prompt optimization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 75–81.

Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and 1 others. 2024. Mora: High-rank updating for parameter-efficient fine-tuning. *arXiv preprint arXiv:2405.12130*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Gurusha Juneja, Gautam Jajoo, Hua Li, Jian Jiao, Nagarajan Natarajan, and Amit Sharma. 2025. Task facet learning: A structured approach to prompt optimization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23473–23496.

Joonseong Kang, Soojeong Lee, Subeen Park, Sumin Park, Taero Kim, Jihee Kim, Ryunyi Lee, and Kyungwoo Song. 2025. Adaptive task vectors for large language models. *arXiv preprint arXiv:2506.03426*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Ceyhun Efe Kayan and Li Zhang. 2025. Prototype-based dynamic steering for large language models. *arXiv preprint arXiv:2510.05498*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2024. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*.

Alex Kulesza, Ben Taskar, and 1 others. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Minchan Kwon, Gaeun Kim, Jongsuk Kim, Haeil Lee, and Junmo Kim. 2024. Stableprompt: Automatic prompt tuning using reinforcement learning for large language model. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 9868–9884.

Sanwoo Lee, Jiahao Liu, Qifan Wang, Jingang Wang, Xunliang Cai, and Yunfang Wu. 2025. Dynamic fisher-weighted model merging via bayesian optimization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4923–4935.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.

Jiaqian Li, Yanshu Li, Ligong Han, Ruixiang Tang, and Wenya Wang. 2025a. Train once, reuse everywhere: Generalizable implicit in-context learning by routing attention. *arXiv preprint arXiv:2509.22854*.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Suyi Li, Hanfeng Lu, Tianyuan Wu, Minchen Yu, Qizhen Weng, Xusheng Chen, Yizhou Shan, Binhang Yuan, and Wei Wang. 2025b. Toppings:{CPU-Assisted},{Rank-Aware} adapter serving for {LLM} inference. In *2025 USENIX Annual Technical Conference (USENIX ATC 25)*, pages 613–629.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pages 4582–4597.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023c. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Yang Li, Shaobo Han, and Shihao Ji. 2024a. VB-loRA: Extreme parameter efficient fine-tuning with vector banks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yanshu Li, Yi Cao, Hongyang He, Qisen Cheng, Xiang Fu, Xi Xiao, Tianyang Wang, and Ruixiang Tang. 2025c. M²IV: Towards efficient and fine-grained multimodal in-context learning via representation engineering. In *Second Conference on Language Modeling*.

Yinheng Li, Rogerio Bonatti, Sara Abdali, Justin Wagle, and Kazuhito Koishida. 2024b. Data generation using large language models for text classification: An empirical case study. *arXiv preprint arXiv:2407.12813*.

Zhuowei Li, Zihao Xu, Ligong Han, Yunhe Gao, Song Wen, Di Liu, Hao Wang, and Dimitris N. Metaxas. 2025d. Implicit in-context learning. In *The Thirteenth International Conference on Learning Representations*.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Baohao Liao and Christof Monz. 2024. 3-in-1: 2d rotary adaptation for efficient finetuning, efficient batching and composability. *Advances in Neural Information Processing Systems*, 37:35018–35048.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of llms for logical reasoning. *arXiv preprint arXiv:2502.01100*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.

Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. Use your instinct: Instruction optimization for llms using neural bandits coupled with transformers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K Luke. 2025. Prompt compression with context-aware sentence encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24595–24604.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in neural information processing systems*, 35:1950–1965.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024a. se2: Sequential example selection for in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5262–5284.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pages 100–114.

Jinda Liu, Yi Chang, and Yuan Wu. 2025. R-lora: Randomized multi-head lora for efficient multi-task learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 660–674.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024b. In-context vectors: making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024c. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022c. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Yewei Liu, Xiyuan Wang, Yansheng Mao, Yoav Gelbery, Haggai Maron, and Muhan Zhang. 2026. Shine: A scalable in-context hypernetwork for mapping context to lora in a single pass. *arXiv preprint arXiv:2602.06358*.

Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. 2024d. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*.

Yiting Liu and Zhi-Hong Deng. 2025. Iterative vectors: In-context gradient steering without backpropagation. In *Forty-second International Conference on Machine Learning*.

Zequan Liu, Jiawen Lyn, Wei Zhu, Xing Tian, and Yvette Graham. 2024e. Alora: Allocating low-rank adaptation for fine-tuning large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 622–641.

Zequan Liu, Yi Zhao, Ming Tan, Wei Zhu, and Aaron Xuxiang Tian. 2024f. Para: Parameter-efficient fine-tuning with prompt-aware representation adjustment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 728–737.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022a. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022b. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2025. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7):197605.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *Preprint*, arXiv:2310.20046.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Qirui Mi, Zhijian Ma, Mengyue Yang, Haoxuan Li, Yisen Wang, Haifeng Zhang, and Jun Wang. 2026. Skill-pro: Learning reusable skills from experience via non-parametric ppo for llm agents. *arXiv preprint arXiv:2602.01869*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 8(6):4663–4678.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. 2023. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314.

Mihai Nadas, Laura Diosan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *arXiv preprint arXiv:2503.14023*.

Hyeon-Woo Nam, Ye-Bin Moon, and Tae-Hyun Oh. 2021. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv:2108.06098*.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1953–1967.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. 2026. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4885–4901.

Amin Heyrani Nobari, Kaveh Alim, Ali Arjomand-Bigdeli, Akash Srivastava, Faez Ahmed, and Navid Azizan. 2025. Activation-informed merging of large language models. *Preprint*, arXiv:2502.02421.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

OpenAI. 2025. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/. Accessed: 2025-09-11.

OpenAI. 2026. GPT-6 Astra: A new generation of intelligence. https://openai.com/index/gpt-6-astra/. Accessed: 2026-09-09.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1173–1186.

Jungwon Park, Jimyeong Kim, Changin Choi, and Wonjong Rhee. 2026. Soft head selection for injecting icl-derived task embeddings. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 27161–27214.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024a. Revisiting demonstration selection strategies in in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9090–9101.

Yingzhe Peng, Chenduo Hao, Xinting Hu, Jiawei Peng, Xin Geng, and Xu Yang. 2024b. Live: Learnable in-context vector for visual question answering. *Advances in Neural Information Processing Systems*, 37:9773–9800.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 487–503.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Daniele Potertì, Andrea Seveso, Fabio Mercorio, and 1 others. 2025. Can role vectors affect llm behaviour. *Findings of the Association for Computational Linguistics: EMNLP*, pages 17735–17747.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Conference on Empirical Methods in Natural Language Processing*.

Kiran Purohit, V Venktesh, Sourangshu Bhattacharya, and Avishek Anand. 2025. Sample efficient demonstration selection for in-context learning. *arXiv preprint arXiv:2506.08607*.

Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. 2024. In-context learning with iterative demonstration selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7441–7455.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Maria Ponti, and Shay Cohen. 2024. Spectral editing of activations for large language model alignment. *Advances in Neural Information Processing Systems*, 37:56958–56987.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, and 1 others. 2024. Hellobench: Evaluating long text generation capabilities of large language models. *arXiv preprint arXiv:2409.16191*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Mubashar Raza, Zarmina Jahangir, Muhammad Bilal Riaz, Muhammad Jasim Saeed, and Muhammad Awais Sattar. 2025. Industrial applications of large language models. *Scientific Reports*, 15(1):13755.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *International Conference on Learning Representations*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Adithya Renduchintala, Tugrul Konuk, and Oleksii Kuchaiev. 2024. Tied-lora: Enhancing parameter efficiency of lora with weight tying. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8694–8705.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.

Baturay Saglam, Xinyang Hu, Zhuoran Yang, Dionysis Kalogerias, and Amin Karbasi. 2025. Learning task representations from in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6634–6663.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147.

Alexander Scarlatos and Andrew Lan. 2023. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. In *International conference on machine learning*, pages 30706–30775. PMLR.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, and 1 others. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*.

Zhengxiang Shi and Aldo Lipani. 2024. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. In *International Conference on Learning Representations*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and 1 others. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Yi Su, Yunpeng Tai, Yixin Ji, Juntao Li, Yan Bowen, and Min Zhang. 2024. Demonstration augmentation for zero-shot in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14232–14244.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Advances in Neural Information Processing Systems*.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*.

Mirac Suzgun, Nathan Scales, Nathanael Sch¨arli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. 2023. Parameter efficient multi-task model fusion with partial linearization. *arXiv preprint arXiv:2310.04742*.

Pengwei Tang, Xiaolin Hu, and Yong Liu. 2025a. ADePT: Adaptive decomposed prompt tuning for parameter-efficient fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, Siyuan Lu, Yaliang Li, and Ji-Rong Wen. 2025b. Unleashing the potential of large language models as prompt optimizers: Analogical analysis with gradient-based model optimizers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25264–25272.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. In *Advances in Neural Information Processing Systems*.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287.

Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different nlp tasks. *arXiv preprint arXiv:2407.12994*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2022. Spot: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059.

Xingchen Wan, Han Zhou, Ruoxi Sun, and Sercan O Arik. 2025. From few to many: Self-improving many-shot reasoners through iterative optimization and generation. In *The Thirteenth International Conference on Learning Representations*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Futing Wang, Jianhao Yan, Yue Zhang, and Tao Lin. 2025a. ELICIT: LLM augmentation via external in-context capability. In *The Thirteenth International Conference on Learning Representations*.

Haowen Wang, Tao Sun, Cong Fan, and Jinjie Gu. 2023a. Customizable combination of parameter-efficient modules for multi-task learning. *arXiv preprint arXiv:2312.03248*.

Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. 2025b. Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artificial Intelligence Review*, 58(8):227.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298.

Shaowen Wang, Linxi Yu, and Jian Li. 2024a. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. 2025c. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, pages 2562–2578.

Weixuan Wang, JINGYUAN YANG, and Wei Peng. 2025d. Semantics-adaptive activation intervention for LLMs via dynamic steering vectors. In *The Thirteenth International Conference on Learning Representations*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023b. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023c. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. 2024b. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *The Twelfth International Conference on Learning Representations*.

Xubin Wang, Jianfei Wu, Yuan Yichen, Deyu Cai, Mingzhe Li, and Weijia Jia. 2025e. Demonstration selection for in-context learning via reinforcement learning. In *Forty-second International Conference on Machine Learning*.

Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. 2023d. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024c. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023e. Do-not-answer: A dataset for evaluating safeguards in llms. *arXiv preprint arXiv:2308.13387*.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023f. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yeming Wen and Swarat Chaudhuri. 2024. Batched low-rank adaptation of foundation models. In *The Twelfth International Conference on Learning Representations*.

Martin Weyssow, Xin Zhou, Kisub Kim, David Lo, and Houari Sahraoui. 2025. Exploring parameter-efficient fine-tuning techniques for code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 34(7):1–25.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. Pmlr.

Bingyang Wu, Ruidong Zhu, Zili Zhang, Peng Sun, Xuanzhe Liu, and Xin Jin. 2024a. {dLoRA}: Dynamically orchestrating requests and adapters for {LoRA}{LLM} serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 911–927.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuan-Jing Huang. 2024b. Advancing parameter efficiency in fine-tuning via representation editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13445–13464.

Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. 2024c. Mixture-of-subspaces in low-rank adaptation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7880–7899.

Yuhao Wu, Ming Shan Hee, Zhiqing Hu, and Roy Ka-Wei Lee. 2024d. Longgenbench: Benchmarking long-form generation in long context llms. *arXiv preprint arXiv:2409.02076*.

Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024e. Prompt optimization with ease? efficient ordering-aware automated selection of exemplars. *Advances in Neural Information Processing Systems*, 37:122706–122740.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436.

Yifei Xia, Fangcheng Fu, Wentao Zhang, Jiawei Jiang, and Bin CUI. 2024. Efficient multi-task LLM quantization and serving for multiple loRA adapters. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jing Xu, Jiazheng Li, and Jingzhao Zhang. 2025. Scalable model merging with progressive layer-wise distillation. In *Forty-second International Conference on Machine Learning*.

Jingwei Xu, Junyu Lai, and Yunpeng Huang. 2024a. Meteora: Multiple-tasks embedded lora for large language models. *arXiv preprint arXiv:2405.13053*.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, XIAOPENG ZHANG, and Qi Tian. 2024b. QA-loRA: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations*.

Prateek Yadav, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023a. Compeft: Compression for communicating parameter efficient updates via sparsification and quantization. *arXiv preprint arXiv:2311.13171*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023b. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Cilin Yan, Jingyun Wang, Lin Zhang, Ruihui Zhao, Xiaopu Wu, Kai Xiong, Qingsong Liu, Guoliang Kang, and Yangyang Kang. 2025. Efficient and accurate prompt optimization: the benefit of memory in exemplar-guided reflection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–779.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024a. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pages 12028–12068.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*.

Haolin Yang, Hakaze Cho, Kaize Ding, and Naoya Inoue. 2025. Task vectors, learned not extracted: Performance gains and mechanistic insight. *Preprint*, arXiv:2509.24169.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, and 1 others. 2026. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Yuxuan Yao, Shuqi LIU, Zehua Liu, Qintong Li, Mingyang LIU, Xiongwei Han, Zhijiang Guo, Han Wu, and Linqi Song. 2025. Activation-guided consensus merging for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. 2024. Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 355–385.

Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. 2023. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *arXiv preprint arXiv:2401.14367*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023.

Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Peiwen Yuan, Shaoxiong Feng, Yiwei Li, Xinglin Wang, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024. Focused large language models are stable many-shot learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6247–6261.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Li-Ming Zhan, Bo LIU, Yujie Feng, Chengqiang Xie, Jiannong Cao, and Xiao-Ming Wu. 2026. REAL: Reading out transformer activations for precise localization in language model steering. In *The Fourteenth International Conference on Learning Representations*.

Jianfei Zhang, Bei Li, Jun Bai, Rumei Li, Yanmeng Wang, Chenghua Lin, and Wenge Rong. 2025a. Selecting demonstrations for many-shot in-context learning via gradient matching. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11686–11704.

Jinghan Zhang, shiqi chen, Junteng Liu, and Junxian He. 2023a. Composing parameter-efficient modules with arithmetic operation. In *Advances in Neural Information Processing Systems*, volume 36, pages 12589–12610. Curran Associates, Inc.

Kaiyi Zhang, Ang Lv, Yuhan Chen, Hansen Ha, Tao Xu, and Rui Yan. 2024a. Batch-icl: Effective, efficient, and order-agnostic in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10728–10739.

Peiyi Zhang, Richong Zhang, Zhijie Nie, and Ziqiao Wang. 2025b. Dynamic task vector grouping for efficient multi-task prompt tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26805–26821.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023b. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Shaokun Zhang, Xiaobo Xia, Zhaoqing Wang, Ling-Hao Chen, Jiale Liu, Qingyun Wu, and Tongliang Liu. 2024b. IDEAL: Influence-driven selective annotations empower in-context learners in large language models. In *The Twelfth International Conference on Learning Representations*.

Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zheng, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. 2024c. In-context principle learning from mistakes. *arXiv preprint arXiv:2402.05403*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xiaoqing Zhang, Ang Lv, Yuhan Liu, Flood Sung, Wei Liu, Jian Luan, Shuo Shang, Xiuying Chen, and Rui Yan. 2025c. More is not always better? enhancing many-shot in-context learning with differentiated and reweighting objectives. *arXiv preprint arXiv:2501.04070*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

Zheng Zhang, Shaocheng Lan, Lei Song, Jiang Bian, Yexin Li, and Kan Ren. 2025d. Learning to select in-context demonstration preferred by large language model. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11345–11360.

Ziniu Zhang, Zhenshuo Zhang, Dongyue Li, Lu Wang, Jennifer Dy, and Hongyang R. Zhang. 2025e. Linear-time demonstration selection for in-context learning via gradient estimation. *Preprint*, arXiv:2508.19999.

Yang Zhao, Pu Wang, and Hao Frank Yang. 2025. How to auto-optimize prompts for domain tasks? adaptive prompting and reasoning through evolutionary domain knowledge adaptation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Kun Kuang, and Fei Wu. 2024. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. *arXiv preprint arXiv:2409.16167*.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. 2024a. On prompt-driven safeguarding for large language models. *arXiv preprint arXiv:2401.18018*.

Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. 2024b. When" a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15126–15154.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Steven Zheng. 2024a. SELF-DISCOVER: Large language models self-compose reasoning structures. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. 2024b. Metagpt: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1724.

A Full Taxonomy of Efficient Task Adaptation in LLMs

Figure 3 presents the full taxonomy of efficient task adaptation methods in large language models. In contrast to the reduced taxonomy in Figure 1 of Section 1, which shows only the main category branches, the full taxonomy includes the individual methods associated with each subcategory.

B Discussion

B.1 Comparison with Previous PEFT-only and ICL-only Taxonomies

This work proposes a unified taxonomy that covers weight-based, prompt-based, and embedding-based task adaptation techniques. While most embedding-based adaptation methods, except for soft prompting (Lester et al., 2021; Li and Liang, 2021), have only recently emerged and therefore lack dedicated surveys, several PEFT-only and ICL-only surveys have previously proposed taxonomies for weight-based and prompt-based adaptation methods (Lialin et al., 2023; Dong et al., 2024; Sahoo et al., 2024; Vatsal and Dubey, 2024; Han et al., 2024; Wang et al., 2025b; Mao et al., 2025). In this section, we compare our taxonomy with those proposed in representative PEFT-only and ICL-only surveys. We select representative surveys based on citation impact, relevance to each subfield, and publication or arXiv release year. The selected surveys are listed below:

PEFT-only surveys

- P1. *Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey* (Han et al., 2024)
- P2. *Parameter-Efficient Fine-Tuning in Large Models: A Survey of Methodologies* (Wang et al., 2025b)
- P3. *A Survey on LoRA of Large Language Models* (Mao et al., 2025)

ICL-only surveys

- I1. *A Survey on In-context Learning* (Dong et al., 2024)
- I2. *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications* (Sahoo et al., 2024)
- I3. *A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks* (Vatsal and Dubey, 2024)

For ease of discussion, we refer to these surveys as P1–P3 for PEFT surveys and I1–I3 for ICL surveys. It is important to note that taxonomies proposed in different surveys naturally reflect the scope and objectives of the respective works. Our goal here is therefore not to claim that our taxonomy is generally preferable to previous ones, but to clarify how the coverage and perspective adopted in our taxonomy differ from prior PEFT-only and ICL-only taxonomies and how they align with the goal of contextualizing a broad range of methods under the theme of task adaptation in LLMs.

Compared with previous single-field surveys, our taxonomy differs in two main aspects. First, our taxonomy spans weight-based, prompt-based, and embedding-based adaptation within a single framework. This broader coverage provides a unified perspective on task adaptation in LLMs and enables a contextualized view of different approaches based on how task information is encoded and utilized. Second, most existing PEFT-only and ICL-only surveys primarily cover works published up to around 2024, whereas our survey additionally includes more recent methods published in 2025 and 2026. Considering that several directions, such as instruction optimization, have recently evolved, introducing a new taxonomy helps better reflect the current landscape of task adaptation methods. Below, we further compare our taxonomy with previous ones for weight-based and prompt-based adaptation, respectively.

Detailed Comparison with PEFT-only Taxonomies. The PEFT-only surveys P1 and P2 categorize methods primarily according to fine-grained technical design choices, such as adapter-based methods, selective parameter updates, reparameterization-based fine-tuning, and quantization-based approaches. Similarly, P3 focuses on detailed categorization of LoRA variants. These taxonomies provide fine-grained methodological classifications that help organize the large body of PEFT research. In contrast, our taxonomy places PEFT methods within the broader category of *weight-based adaptation*. This category additionally includes approaches such as model merging and multi-task adaptation, which are not typically emphasized in previous PEFT-only taxonomies. Instead of organizing methods by specific

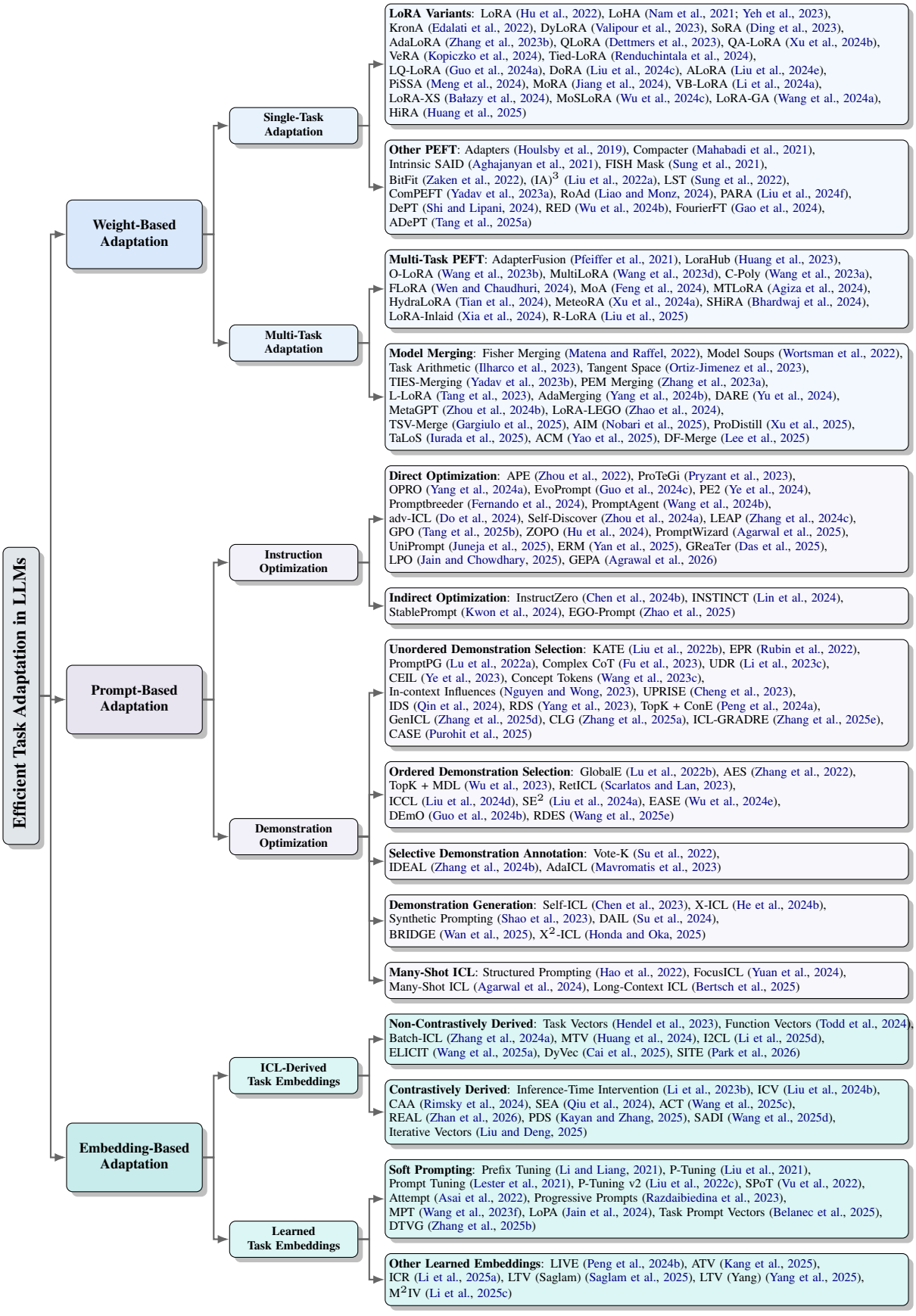


Figure 3: **Full Taxonomy of Efficient Task Adaptation in Large Language Models.**

architectural techniques, our taxonomy categorizes subfields according to the challenges they address in task adaptation. Specifically, as discussed in Section 3, weight-based adaptation is first divided into single-task and multi-task adaptation. Single-task adaptation primarily focuses on improving training efficiency and task performance, whereas multi-task adaptation addresses challenges such as switching tasks during inference or performing mixed-task batched inference. Within multi-task adaptation, we further distinguish between multi-task PEFT and model merging. While both approaches enable multi-task adaptation, model merging is particularly relevant in scenarios where training data are unavailable and only task-specific adapted models are accessible. Because these perspectives are not explicitly reflected in existing PEFT-only taxonomies, introducing a taxonomy tailored to the broader context of task adaptation is more suitable for a unified survey such as ours.

Detailed Comparison with ICL-only Taxonomies. Previous ICL surveys adopt different perspectives in organizing prompt-based methods. The taxonomy in I1 covers both methodological developments and analytical studies of ICL. In contrast, I2 and I3 categorize prompt engineering techniques primarily according to their target applications, such as logical reasoning, hallucination reduction, emotion or tone control, and code generation. Our taxonomy organizes prompt-based adaptation from a different perspective by focusing on how task information is encoded and on the main challenges addressed in task adaptation. Under this perspective, our taxonomy includes categories such as unordered demonstration selection, ordered demonstration selection, selective demonstration annotation, demonstration generation, and many-shot ICL. Several of these directions are either not included or only partially covered in previous ICL surveys. For example, recent instruction optimization methods have become an important branch of prompt-based adaptation but are only sparsely represented in earlier surveys due to their recent emergence. Similarly, subfields such as selective demonstration annotation, demonstration generation, and many-shot ICL play important supporting roles in prompt-based task adaptation but are not systematically categorized in previous ICL taxonomies. By incorporating these directions, our taxonomy provides a more comprehensive contextualization of prompt-based task adaptation methods.

Summary. Overall, our taxonomy differs from previous PEFT-only and ICL-only taxonomies in both scope and organizing perspective. While previous surveys focus on individual subfields and often categorize methods by architectural techniques or application domains, our taxonomy provides a unified view spanning weight-based, prompt-based, and embedding-based adaptation. By organizing methods according to the challenges they address in task adaptation, our taxonomy highlights conceptual connections across different adaptation strategies while preserving the methodological insights developed within each individual research area.

B.2 Detailed Comparison of Weight-Based and Embedding-Based Adaptations

In our taxonomy, we distinguish between weight-based and embedding-based adaptations. A conceptual comparison of these paradigms is summarized in Table 1 of Section 2. In this section, we further clarify the differences between these two adaptation paradigms.

Following terminology commonly used in prior literature (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016), we first clarify the meanings of *weights* and *embeddings (or activations)*. Conceptually, weights are parameters that define the transformations applied to activations, typically involving multiplicative operations. In contrast, embeddings or activations refer to the outputs produced by these transformations. Based on this distinction, we categorize methods that encode task information into model weights, including additional weights or rescaling coefficients, as *weight-based adaptation*. In contrast, methods that encode task information directly in activation-space representations are categorized as *embedding-based adaptation*. This includes approaches that encode task information in soft prompts within the input embedding space or in injected activation vectors within the activation space of intermediate layers.

At first glance, this distinction may appear unfamiliar because weight-based PEFT methods such as LoRA (Hu et al., 2022) are often discussed within the broader PEFT literature together with soft prompting methods. In our taxonomy, however, weight-based PEFT methods are categorized as weight-based adaptation because they encode task information in model weights, whereas soft prompting methods are categorized as embedding-based adaptation because task information is encoded in the embedding space. We introduce this catego-

rization because it more clearly reflects where task information is encoded in the model.

This perspective also helps clarify relationships between different adaptation approaches. For instance, recently proposed learned embedding methods (Peng et al., 2024b; Li et al., 2025c; Saglam et al., 2025; Li et al., 2025a; Kang et al., 2025; Yang et al., 2025), such as LTV (Saglam et al., 2025), share conceptual similarities with soft prompting approaches because both encode task information in activation-space representations optimized through gradient-based training. However, this connection is often overlooked in previous studies (Peng et al., 2024b; Saglam et al., 2025; Li et al., 2025a; Yang et al., 2025), which rarely discuss or compare these methods with soft prompting approaches. By organizing methods according to where and how task information is encoded, our taxonomy highlights these conceptual connections and provides a unified framework for systematically analyzing and comparing a broad range of task adaptation techniques.

B.3 Quantitative Meta-Analysis

The values summarized in Table 2 are derived from the results reported in prior work (Park et al., 2026). We provide this focused quantitative comparison to complement the qualitative analysis in Section 2 by illustrating differences in task performance, trainable parameter count, and runtime among representative methods.

The Big-Bench Hard scores and trainable parameter counts illustrate the trade-off between task performance and parameter efficiency. Weight-based adaptation remains a strong baseline, with LoRA and $(IA)^3$ achieving the highest scores among the compared methods. In contrast, 10-shot ICL requires no trainable parameters but performs lower than these PEFT methods, reflecting the trade-off between training-free adaptation and task performance. Embedding-based methods use substantially fewer trainable parameters than weight-based methods. Among ICL-derived embedding methods, SITE achieves the strongest performance, outperforming 10-shot ICL while using only 1.02K trainable parameters. This suggests that ICL-derived task embeddings have the potential to provide a compact alternative to repeatedly including demonstrations in the input prompt. However, the low scores of FV and LIVE also indicate that embedding-based adaptation does not uniformly improve performance, and its effectiveness depends strongly on how task embeddings are constructed and injected.

The runtime results should be interpreted separately from the Big-Bench Hard scores. While the Big-Bench Hard scores are reported on Big-Bench Hard, the total runtime is measured separately on 10,000 AG_News test queries using Llama-3.1-8B on a single NVIDIA A6000 GPU. Under this setting, 10-shot ICL requires longer runtime than LoRA and SITE, mainly due to the inference overhead from repeatedly processing long demonstration-containing prompts. SITE, a recently proposed ICL-derived embedding-based method, shows runtime comparable to LoRA, whereas FV requires substantially longer runtime due to the additional search procedure for selecting intervention heads. For methods requiring task-specific training, the reported runtime includes both training and evaluation time.

Importantly, this analysis should not be interpreted as a general ranking of adaptation paradigms. The reported scores and runtimes are based on a limited set of representative methods, a single backbone model, selected benchmarks, and specific experimental settings. Therefore, they may not fully reflect general trends across different model families, model scales, tasks, implementations, or hardware environments. Rather, this comparison provides a limited empirical snapshot that illustrates the broader qualitative trade-offs discussed in this survey.

B.4 Paper Collection Process

We describe the paper collection process used to construct the taxonomy in this survey. Our goal is to cover representative and influential work across weight-based, prompt-based, and embedding-based adaptation, rather than to provide an exhaustive list of all related papers.

We first identified foundational papers in each paradigm. We then expanded the candidate set by examining papers that cite these works, prioritizing highly cited papers, papers published in major NLP or machine learning venues, and papers frequently used as baselines in subsequent studies. To improve coverage of recent developments, we also monitored recent-paper platforms such as Trending Papers on Hugging Face (Hugging Face, 2026) and screened newly released works relevant to efficient task adaptation.

This process was applied iteratively: after adding new papers, we reviewed their citations, baselines,

Adaptation Type	Method	# Trainable Parameters	Big-Bench Hard	Total Runtime
Weight-Based Adaptation	LoRA (Hu et al., 2022)	3407.87K	**60.39** ± (0.48)	171.2 min
	$(IA)^3$ (Liu et al., 2022a)	524.29K	60.29 ± (0.70)	N/A
Prompt-Based Adaptation	10-shot ICL (Brown et al., 2020)	-	47.17 ± (0.98)	248.4 min
Embedding-Based Adaptation (ICL-derived)	FV (Todd et al., 2024)	-	17.82 ± (0.37)	411.2 min
	MTV (Huang et al., 2024)	1.02K	42.54 ± (0.71)	328.8 min
	I2CL (Li et al., 2025d)	0.13K	50.60 ± (1.12)	178.7 min
	SITE (Park et al., 2026)	1.02K	58.04 ± (0.72)	172.0 min
Embedding-Based Adaptation (Learned)	Prompt Tuning (Lester et al., 2021)	81.92K	33.45 ± (0.81)	N/A
	LIVE (Peng et al., 2024b)	131.10K	12.89 ± (2.32)	N/A

Table 2: Quantitative comparison of representative task adaptation methods, based on the results reported in prior work (Park et al., 2026). Big-Bench Hard (Suzgun et al., 2023) scores are evaluated using Llama-3.1-8B (Grattafiori et al., 2024). Total runtime is measured separately on 10,000 AG_News (Zhang et al., 2015) test queries using Llama-3.1-8B on a single NVIDIA A6000 GPU, and is intended only as an efficiency comparison under this fixed setting. For methods requiring task-specific training, runtime includes both training and evaluation time; for FV, it also includes the time of the search procedure for selecting attention heads to intervene on.

and related-work discussions to identify additional relevant methods. We repeated this process until the collected set covered the main methodological directions in each paradigm. Since research on LLM adaptation is rapidly evolving, some recent or concurrent works may still be missing. We therefore view the taxonomy as a structured snapshot of the literature at the time of writing.

C Use of LLMs in This Work

We used chat-based LLMs for sentence-level editing to check grammar and improve clarity during paper writing. All edits were reviewed and verified by the authors. All conceptual contributions are solely by the authors.

Category	Benchmark	Output Type	Size	Description
Natural Language Understanding	TREC (Li and Roth, 2002)	Classification	∼6.0K	TREC Question Classification is a question classification benchmark consisting of open-domain questions annotated with a hierarchical taxonomy of six coarse and fifty fine-grained semantic classes, designed to evaluate question intent understanding and answer type prediction.
	CoNLL-2003 (Sang and De Meulder, 2003)	Classification	∼22K	CoNLL-2003 is a widely used named entity recognition benchmark focusing on the identification of four entity types including persons, locations, organizations, and miscellaneous names, annotated in English and German newswire text to evaluate information extraction.
	Subj (Pang and Lee, 2005)	Classification	∼10K	Subjectivity detection dataset consisting of 5,000 subjective movie reviews and 5,000 objective plot summaries designed to test the ability of language models to distinguish between fact-based objective descriptions and opinion-based subjective statements.
	SST-5 (Socher et al., 2013)	Classification	∼12K	SST-5 is a fine-grained sentiment analysis dataset providing five sentiment labels ranging from very negative to very positive, designed to evaluate models' ability to capture subtle emotional shifts and semantic compositionality in movie review phrases.
	SNLI (Bowman et al., 2015)	Classification	∼570K	SNLI is a large-scale dataset of human-written English sentence pairs manually labeled into entailment, contradiction, and neutral categories, designed to evaluate models' ability to capture fundamental logical and semantic relationships.
	DBPedia (Zhang et al., 2015)	Classification	∼630K	DBPedia is a large-scale topic classification benchmark consisting of Wikipedia articles labeled with 14 non-overlapping DBpedia ontology classes, designed to evaluate models' ability to perform document-level topic and semantic classification.
	WikiQA (Yang et al., 2015)	Classification	∼3.0K	WikiQA is an open-domain question answering benchmark consisting of question–sentence pairs derived from Wikipedia search logs, annotated to evaluate answer sentence selection by identifying sentences that contain the correct answer.
	GLUE (Wang et al., 2018)	Classification	∼1.05M	GLUE is a unified multi-task benchmark for evaluating general natural language understanding, consisting of nine established NLU tasks: CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, RTE, and WNLI, covering linguistic acceptability, sentiment analysis, paraphrase detection, semantic similarity, and natural language inference.
	SuperGLUE (Wang et al., 2019)	Classification / Multiple Choice Question	∼184K	SuperGLUE is a more challenging successor to GLUE, designed to evaluate advanced natural language understanding beyond GLUE's saturation point. It consists of eight established tasks including BoolQ, CB, COPA, MultiRC, ReCoRD, RTE, WiC, and WSC, and it targets complex reasoning, commonsense inference, word sense disambiguation, and coreference resolution.
	ANLI (Nie et al., 2020)	Classification	∼168K	ANLI is a challenging NLI benchmark constructed through an iterative, adversarial human-and-model-in-the-loop process over three rounds (R1-R3), targeting examples that strong existing models fail to solve and providing a robust evaluation of advanced natural language inference.
Reasoning	ARC (Clark et al., 2018)	Multiple Choice Question	∼7.8K	ARC is a multiple-choice grade-school science QA benchmark split into Easy and Challenge sets, designed to test scientific reasoning beyond surface-level pattern matching.
	CommonsenseQA (Talmor et al., 2019)	Multiple Choice Question	∼12K	CommonsenseQA is a 5-way multiple-choice commonsense question answering benchmark grounded in the ConceptNet knowledge graph, where questions are authored to discriminate among concept candidates sharing a semantic relation, designed to evaluate diverse commonsense reasoning about everyday situations, physical properties, and social interactions.
	HellaSwag (Zellers et al., 2019)	Multiple Choice Question	∼60K	HellaSwag is designed to evaluate commonsense reasoning through sentence-completion tasks. It presents everyday scenarios with four multiple-choice endings and asks models to select the most plausible next event; it uses adversarial filtering to generate challenging distractor endings that are difficult for current models but typically easy for human evaluators.
	MMLU (Hendrycks et al., 2020)	Multiple Choice Question	∼16K	MMLU is a comprehensive multiple-choice benchmark covering 57 subjects across STEM, humanities, and social sciences, designed to evaluate models' broad world knowledge and knowledge-intensive problem-solving ability across a wide range of difficulty levels.
	WinoGrande (Sakaguchi et al., 2021)	Multiple Choice Question	∼44K	WinoGrande is a large-scale adversarial commonsense reasoning benchmark of Winograd-style pronoun resolution problems formulated as binary-choice sentence completion, designed to evaluate whether models can use contextual commonsense rather than shallow statistical cues; it applies AFLITE-based debiasing/adversarial filtering to reduce dataset-specific biases.
	BBH (Suzgun et al., 2023)	Multiple Choice Question / Open-Ended Generation	∼6.5K	BBH is a curated subset of BIG-bench (Srivastava et al., 2023) consisting of challenging reasoning tasks where prior language models underperformed, designed to test advanced multi-step reasoning across diverse problem types.
	GPQA (Rein et al., 2024)	Multiple Choice Question	∼0.4K	GPQA is a graduate-level multiple-choice benchmark written by domain experts in biology, physics, and chemistry, designed to evaluate scientific reasoning and problem solving with questions intended to be "Google-proof," i.e., difficult for skilled non-experts even with unrestricted web access.
	MMLU-Pro (Wang et al., 2024c)	Multiple Choice Question	∼12K	MMLU-Pro is a multi-task multiple-choice benchmark with questions drawn from 14 broad disciplines, constructed by filtering and refining MMLU-style items and adding more reasoning-focused questions from additional sources, and it uses an expanded answer set with ten options per question to better probe challenging understanding and reasoning.

Table 3: **Comprehensive Overview of Benchmarks for LLM Task Adaptation (Part 1 of 4).**

Category	Benchmark	Output Type	Size	Description
Mathematics	AQuA-RAT (Ling et al., 2017)	Multiple Choice Question / Open-Ended Generation	∼100K	AQuA-RAT is an algebraic word problem 5-way multiple-choice benchmark where each question is paired with a step-by-step natural-language rationale (often including human-readable math expressions), designed to evaluate interpretable multi-step mathematical reasoning beyond predicting only the final answer.
	MATH (Hendrycks et al., 2021)	Open-Ended Generation	∼13K	MATH is a competition-level mathematics problem-solving benchmark covering diverse topics (e.g., algebra, geometry, number theory, counting & probability, and precalculus), where each problem includes a full step-by-step solution and a final answer, designed to evaluate advanced mathematical reasoning and multi-step problem solving.
	SVAMP (Patel et al., 2021)	Open-Ended Generation	∼1.0K	SVAMP is an adversarial challenge benchmark of one-unknown arithmetic math word problems created by applying small but systematic variations to existing problems, designed to evaluate robust multi-step arithmetic reasoning beyond keyword matching and shallow statistical cues.
	GSM8K (Cobbe et al., 2021)	Open-Ended Generation	∼8.8K	GSM8K is a grade-school math word problem benchmark consisting of natural-language questions with final numeric answers, designed to evaluate multi-step arithmetic reasoning required to solve the problems.
	TabMWP (Lu et al., 2022a)	Multiple Choice Question / Open-Ended Generation	∼38K	TabMWP is a tabular math word problem benchmark where each question is paired with a tabular context provided in multiple formats (e.g., table image and structured text), designed to evaluate joint reasoning over tables and natural-language descriptions for deriving correct numerical answers (with gold step-by-step solutions available).
	MATH500 (Lightman et al., 2023)	Open-Ended Generation	∼0.5K	MATH500 is a curated evaluation subset of the MATH dataset, designed to provide an efficient yet diverse test of competition-style mathematical problem solving.
	MetaMathQA (Yu et al., 2023)	Open-Ended Generation	∼395K	MetaMathQA is an augmented mathematical reasoning dataset bootstrapped from the training sets of GSM8K and MATH by generating diverse, semantically equivalent problem variants (with answer augmentation), designed to improve models' robustness in multi-step mathematical reasoning across varied linguistic formulations.
Coding	HumanEval (Chen et al., 2021a)	Open-Ended Generation	∼0.2K	HumanEval is a benchmark of handwritten Python programming tasks specified by function signatures and docstrings, designed to evaluate models' ability to generate correct code solutions for the given problem specifications.
	MBPP (Austin et al., 2021)	Open-Ended Generation	∼1.0K	MBPP is a benchmark of entry-level Python programming tasks specified by short natural-language problem descriptions, designed to evaluate models' ability to synthesize correct short programs involving basic algorithms and common data-structure manipulations.
	MultiPL-E (Cassano et al., 2023)	Open-Ended Generation	∼25K	MultiPL-E is a polyglot code generation benchmark that translates Python-based programming tasks from HumanEval and MBPP into many target languages, designed to evaluate models' cross-language code synthesis ability and consistency across diverse programming languages.
	SWE-bench (Jimenez et al., 2023)	Open-Ended Generation	∼2.3K	SWE-bench is a software engineering benchmark built from real GitHub issues and corresponding pull requests across multiple popular repositories, designed to evaluate models' ability to understand and modify large codebases by producing code changes that resolve the described issues (e.g., bug fixes or feature requests).
	CruxEval (Gu et al., 2024)	Open-Ended Generation	∼0.8K	CruxEval is a Python function reasoning benchmark consisting of short functions paired with input-output examples, designed to evaluate models' ability to understand program execution by performing output prediction (infer the output for a given input) and input prediction (find an input that produces a given output).

Table 4: **Comprehensive Overview of Benchmarks for LLM Task Adaptation (Part 2 of 4).**

Category	Benchmark	Output Type	Size	Description
Question Answering	SQuAD v1.1 (Rajpurkar et al., 2016)	Open-Ended Generation	∼98K	SQuAD v1.1 is an extractive reading comprehension benchmark built from questions written on Wikipedia passages, where each question is paired with an answer that is a contiguous text span from the given passage, designed to evaluate models' ability to perform span-based question answering.
	TriviaQA (Joshi et al., 2017)	Open-Ended Generation	∼96K	TriviaQA is a reading-comprehension benchmark built from trivia questions paired with evidence documents from Wikipedia and the web, designed to test answering complex, compositional questions despite substantial mismatch between questions and supporting evidence.
	HotpotQA (Yang et al., 2018)	Open-Ended Generation	∼113K	HotpotQA is a multi-hop QA benchmark where each question is paired with supporting facts across multiple Wikipedia articles, designed to test whether models can integrate evidence from more than one document to answer complex questions.
	Natural Questions (Kwiatkowski et al., 2019)	Open-Ended Generation	∼323K	Natural Questions is an open-domain QA benchmark built from real Google search queries paired with Wikipedia pages, providing annotations for both long answers (passages) and short answers (specific entities) to test end-to-end question answering grounded in retrieved evidence.
Summarization	CNN/DailyMail (Hermann et al., 2015)	Open-Ended Generation	∼312K	CNN/DailyMail is a news summarization benchmark consisting of full news articles paired with human-written highlights (bullet-style summary sentences), designed to evaluate models' ability to produce concise summaries that capture the key information in long-form news reports.
	XSum (Narayan et al., 2018)	Open-Ended Generation	∼227K	XSum is an extreme abstractive summarization benchmark pairing BBC news articles with single-sentence summaries that capture what the article is about, designed to measure models' ability to produce highly condensed, gist-focused summaries rather than detail-preserving paraphrases.
	SAMSum (Gliwa et al., 2019)	Open-Ended Generation	∼16K	SAMSum is an abstractive dialogue summarization benchmark consisting of messenger-style chat conversations with human-written summaries, designed to evaluate models' ability to summarize informal, multi-speaker dialogues that may include colloquial language such as slang and emoticons.
	DialogSum (Chen et al., 2021b)	Open-Ended Generation	∼13K	DialogSum is an abstractive dialogue summarization benchmark consisting of multi-turn dialogues from diverse real-life scenarios (compiled from multiple dialogue sources) paired with human-written summaries and topics, designed to evaluate models' ability to capture salient information and speaker intents in multi-speaker conversations
	XL-Sum (Hasan et al., 2021)	Open-Ended Generation	∼1.4M	XL-Sum is a large-scale multilingual abstractive summarization benchmark consisting of BBC news article–summary pairs across dozens of languages, designed to evaluate summarization capability across both high- and low-resource languages.
Structured Data-to-Text	E2E NLG (Novikova et al., 2017)	Open-Ended Generation	∼51K	E2E NLG is a restaurant-domain data-to-text generation benchmark that pairs dialogue-act-style meaning representations in the form of attribute–value pairs with human-written utterances, and is designed to evaluate end-to-end verbalization and content selection under diverse surface realizations.
	WebNLG (Gardent et al., 2017)	Open-Ended Generation	∼22K	WebNLG is a data-to-text benchmark that maps sets of DBpedia RDF triples to short natural-language texts, and is designed to evaluate micro-planning in generation, including lexicalization, aggregation, referring expression generation, and sentence segmentation, while preserving the input facts.
	ToTTo (Parikh et al., 2020)	Open-Ended Generation	∼136K	ToTTo is a controlled table-to-text generation benchmark pairing Wikipedia tables with a set of highlighted cells and a human-written one-sentence description, designed to evaluate grounded and faithful text generation from structured tabular data under explicit content selection.

Table 5: **Comprehensive Overview of Benchmarks for LLM Task Adaptation (Part 3 of 4).**

Category	Benchmark	Output Type	Size	Description
Safety and Trustworthiness	HateSpeech18 (De Gibert et al., 2018)	Classification	∼11K	HateSpeech18 is a sentence-level hate speech dataset sampled from posts on the Stormfront white-supremacist forum and manually labeled as hate vs non-hate, designed to evaluate models' ability to detect hate speech in highly domain-specific, ideologically skewed online discussions.
	RealToxicityPrompts (Gehman et al., 2020)	Open-Ended Generation	∼100K	RealToxicityPrompts is a dataset of 100k naturally occurring, sentence-level prompts drawn from English web text and paired with toxicity annotations, designed to evaluate whether language models produce toxic continuations when conditioned on real-world prompts spanning a range of toxicity levels.
	CrowS-Pairs (Nangia et al., 2020)	Multiple Choice Question	∼1.5K	CrowS-Pairs is a social bias evaluation benchmark consisting of sentence pairs that differ in whether they express a stereotype, covering nine bias categories (e.g., race, gender, religion, age), designed to measure models' tendency to prefer stereotypical statements over less-stereotyping alternatives.
	ToxiGen (Hartvigsen et al., 2022)	Classification	∼274K	ToxiGen is a large-scale machine-generated toxicity dataset consisting of toxic and benign statements about minority identity groups, created to surface implicit and adversarially crafted toxic language beyond explicit slurs or profanity, and designed to evaluate models' ability to detect subtle harmful content rather than relying on group-mention shortcuts.
	ParaDetox (Logacheva et al., 2022)	Open-Ended Generation	∼20K	ParaDetox is a parallel text detoxification dataset pairing toxic sentences with human-written non-toxic paraphrases, designed to test whether models can remove toxicity while preserving the original meaning.
	TruthfulQA (Lin et al., 2022)	Multiple Choice Question / Open-Ended Generation	∼0.8K	TruthfulQA is a question answering benchmark consisting of questions spanning 38 categories that are crafted to trigger common misconceptions, designed to evaluate whether models can produce truthful, misconception-resistant answers rather than imitating popular human falsehoods.
	ETHOS (Mollas et al., 2022)	Classification	∼1.0K	ETHOS is an online hate-speech detection dataset of YouTube and Reddit comments with both a binary hate-speech label and an additional multi-label variant that annotates hateful comments with fine-grained hate categories.
	HaluEval (Li et al., 2023a)	Classification	∼35K	HaluEval is a hallucination benchmark built from instruction-style prompts and task-specific contexts (QA, dialogue, and summarization), pairing responses with annotations of whether content is hallucinated to assess factuality and faithfulness in generated text.
	Do-Not-Answer (Wang et al., 2023e)	Open-Ended Generation	∼0.9K	Do-Not-Answer is a safety safeguard evaluation benchmark consisting of risky questions and instructions spanning a structured taxonomy of harm types (e.g., privacy leakage, illegal activities, toxic content, misinformation harms, and human–chatbot interaction risks), designed to assess whether models can avoid facilitating harm and respond responsibly to unsafe requests.
General ICL Capability	FV Benchmark (Todd et al., 2024)	Multiple Choice Question / Open-Ended Generation	∼9.3K	FV Benchmark comprises 29 abstractive and 28 extractive tasks, including closely related and contrasting variants, making it well suited for studying in-context learning behaviors. Abstractive tasks require generating information not explicitly present in the prompt, whereas extractive tasks involve directly retrieving the answer from it.
	ICL-50 (Zhang et al., 2025c)	Multiple Choice Question / Open-Ended Generation	∼3.2M	ICL-50 is a many-shot in-context learning dataset spanning 50 tasks across several task families, created to study how model performance and behavior change as the number of in-context examples scales to long contexts.

Table 6: **Comprehensive Overview of Benchmarks for LLM Task Adaptation (Part 4 of 4).**