From Weights to Activations: Is Steering the Next Frontier of Adaptation?

Simon Ostermann[1,2,3*] **Daniil Gurgurov**[1,2*]
Tanja Baeumel[1,2,3] **Michael A. Hedderich**[4,5] **Sebastian Lapuschkin**[6,7]
Wojciech Samek[6,8,9] **Vera Schmitt**[2,3,8]

[1] Saarland University [2] German Research Center for Artificial Intelligence (DFKI)
[3] Centre for European Research in Trusted AI (CERTAIN) [4] Center for Information and Language Processing, LMU Munich
[5] Munich Center for Machine Learning [6] Fraunhofer Heinrich Hertz Institute [7] Technological University Dublin
[8] Technische Universität Berlin [9] Berlin Institute for the Foundations of Learning and Data (BIFOLD)
simon.ostermann@dfki.de daniil.gurgurov@dfki.de

Abstract

Post-training adaptation of language models is commonly achieved through parameter updates or input-based methods such as fine-tuning, parameter-efficient adaptation, and prompting. In parallel, a growing body of work modifies internal activations at inference time to influence model behavior, an approach known as **steering**. Despite increasing use, steering is rarely analyzed within the same conceptual framework as established adaptation methods.

In this work, we argue that steering should be regarded as a form of model adaptation. We introduce a set of functional criteria for adaptation methods and use them to compare steering approaches with classical alternatives. This analysis positions steering as a distinct adaptation paradigm based on targeted interventions in activation space, enabling local and reversible behavioral change without parameter updates. The resulting framing clarifies how steering relates to existing methods, motivating a unified taxonomy for model adaptation.

1 Introduction

Pre-trained large language models (LLMs) form the basis of a wide range of NLP tasks, making adaptation to new tasks, domains, or behavioral constraints a central problem. Early approaches relied on full fine-tuning of models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), while later work introduced parameter-efficient techniques, including adapters (Houlsby et al., 2019), soft prompts (Lester et al., 2021), and low-rank adaptation (Hu et al., 2021). As model sizes increased, even these methods became costly, requiring substantial compute and training infrastructure; thus, producing task-specific variants became difficult to maintain at scale (Patterson et al., 2021; Bommasani et al., 2022). There is a growing demand for adaptation methods that enable fast and flexible behavioral modification without retraining, since even parameter-efficient approaches still rely on training pipelines and hyperparameter tuning (Wang et al., 2025a). Prompting and in-context learning can only partially account for this, as both suffer from sensitivity to phrasing and example order, leading to unstable behavior (Chatterjee et al., 2024).

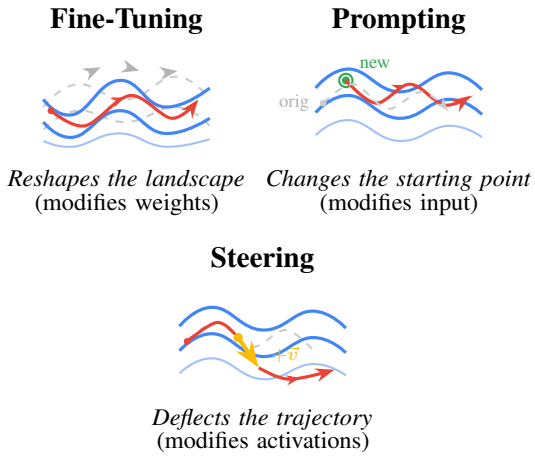

Figure 1: Conceptual illustration of three mechanisms for post-training model adaptation. Fine-tuning modifies the weight-defined behavior landscape during training, prompting alters the input-induced trajectory at inference time, and steering intervenes on internal activations during inference to deflect that trajectory.

In parallel to weight- and input-based approaches, a growing line of work has emerged from interpretability research that modifies model behavior through efficient **additive inference-time interventions on internal activations**, commonly referred to as *steering*. Traditionally, such interventions were used as experimental probes to study the internal mechanisms of LLMs (Alain and Bengio, 2016; Ivanova et al., 2021; Elhage et al., 2021; Marks and Tegmark, 2023). More recent work, however, demonstrates that targeted activation interventions can reliably induce new

*: Equal contribution. The remaining authors are sorted alphabetically.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29854–29879

		Reliability	Generalization	Specificity	Compute Efficiency	Data Efficiency	Composability	Usability	Reversibility
Prompt	*Prompting*	0	0	0	+	+	+	+	+
	ICL	0	0	0	+	+	+	+	+
FT	*Fine-tuning*	+	+	-	-	-	-	-	-
	RLHF	+	+	-	-	-	-	-	-
PEFT	*Adapters*	+	+	0	+	-	+	-	+
	Soft Prompt Tuning	+	+	0	+	0	+	-	+
	LoRA	+	+	0	+	0	+	-	+
Steering	*Difference*	+	0	+	+	+	0	0	+
	Optimization	+	+	+	0	0	0	0	+
	Dictionary	0	+	+	-	-	0	0	+

Table 1: Comparison of adaptation methods by functional criteria. **+**: a criterion is commonly demonstrated in the literature under standard settings; **-**: a criterion is systematically reported as a limitation or cost; **0**: mixed evidence, under-exploration, or high task dependence. The values for each method are justified in Sections 3.2 and 5.

behaviors without modifying model parameters or retraining. Steering methods have been shown to influence properties such as tone, factuality, safety, and alignment (Turner et al., 2023; Panickssery et al., 2023; Li et al., 2023; Arditi et al., 2024; Konen et al., 2024).

Despite increasing empirical use, steering methods are rarely analyzed within the same conceptual or evaluative framework as established adaptation techniques. Existing work primarily compares different steering approaches to one another, or to prompting baselines, with limited comparison to fine-tuning or parameter-efficient adaptation methods (Wu et al., 2025; Gurgurov et al., 2026). As a result, it remains unclear how steering relates to classical adaptation paradigms, which trade-offs it entails, and under which conditions it should be preferred.

In this work, we argue that steering should be viewed as a form of model adaptation rather than solely as an interpretability technique. We introduce a set of functional criteria for adaptation methods and use them to systematically compare steering approaches with fine-tuning, parameter-efficient adaptation, and prompting. Framing steering as adaptation highlights a distinct design point in which behavior is modified directly in activation space, enabling local and reversible control without parameter updates, and situates steering alongside existing adaptation paradigms. Figure 1 illustrates the differences in common adaptation mechanisms: fine-tuning modifies the weight-defined behavior landscape, prompting alters the input-induced trajectory, and steering intervenes on internal activations during inference to deflect that trajectory.

Our contributions are threefold:

Contributions

- **(i)** Functional criteria for model adaptation grounded in prior work (Table 1).
- **(ii)** Systematic comparison of established adaptation methods and steering approaches under these criteria.
- **(iii)** Conceptual argument for broadening adaptation beyond weight and input modifications to include targeted interventions on internal activations.

2 Functional Criteria for Model Adaptation

Adaptation methods, especially for LLMs, have grown diverse. Previous work has usually concentrated on evaluating only a few isolated dimensions of adaptation, which inherently are non-exhaustive. To address this gap, we propose a set of criteria, chosen to capture the most important dimensions of adaptation. Our selection is grounded in related surveys and datasets that investigate specific adaptation methods.

Reliability. A reliable adaptation method preserves stable behavior under repeated trials, input variation, and shifts in operating conditions within the domain. This includes consistency in quantitative metrics on data from similar domains, as well

as low variance in qualitative outcomes in light of similar inputs. Zhao et al. (2025a), for example, broadly discuss robustness and stability of LLMs, which can be subsumed under *reliability*.

→ Generalization. Generalization reflects the capacity of the adapted model to apply learned adjustments to settings that were absent during training (Ben-David et al., 2010; Mansour et al., 2023). A method with strong generalization limits overfitting, maintains broad reasoning competency, and supports transfer across tasks. Zhao et al. (2025a) propose to investigate cross-task and cross-domain performance, which maps to the feature of *generalization*. Lu et al. (2025) survey domain adaptation techniques and their generalization potential.

★ Specificity. Specificity concerns the precision with which an adaptation modifies the model. High specificity yields targeted improvements in a chosen capability while reducing spillover into unrelated behaviors. This preserves the integrity of the base model and ensures controlled functional changes. For LLMs, a highly specific adaptation method results in little to no degradation of general LLM capabilities. *Specificity* is included by Zhao et al. (2025a), who review degrees of catastrophic forgetting in LLM adaptation methods. A range of work on catastrophic forgetting underlines the importance of measuring *specificity* (Li et al., 2024; Kotha et al., 2024; Luo et al., 2025).

Compute Efficiency. Compute efficiency measures the resource demands of the adaptation process during training, along with its impact on inference. A suitable method limits training cost, memory requirements, and runtime overhead. This expands feasibility for practical applications and enables wider experimentation. Lialin et al. (2023) propose to evaluate PEFT methods across diverse efficiency dimensions.

Data Efficiency. Data efficiency captures how well an adaptation method functions when available data is scarce or noisy. Methods with strong data efficiency extract meaningful signal from limited data and maintain performance without large training data, which is essential for specialized domains. Data efficiency is one of the most addressed dimensions in previous work (Liu et al., 2022; Pecher et al., 2025; Anisuzzaman et al., 2025).

Composability. Composability evaluates whether multiple adaptations can be combined without harmful interactions. A composable method supports modular updates Pfeiffer et al. (2020) whose effects can be analyzed and integrated in a predictable manner, simplifying iterative development and deployment across varied contexts. Composability is rarely measured directly when looking at adaptation, but a large strand of recent work tries to achieve it using parameter or activation arithmetics and other combination techniques (Ilharco et al., 2023; Wang et al., 2024a; Belanec et al., 2025).

Usability. Usability reflects the ease with which an adaptation method can be applied and evaluated. This includes clarity of procedure, compatibility with established toolchains, and transparency of behavior during analysis. High usability lowers the barrier to adoption and supports reproducible research. Recent work investigates challenges non-AI experts face with prompting (Zamfirescu-Pereira et al., 2023a,b). Mizrahi et al. (2024a) highlight brittleness issues that undermine prompting usability as an adaptation approach.

Reversibility. Reversibility describes the extent to which an adaptation can be undone or adjusted without lasting side effects on the model. A reversible method allows behavior to be modified temporarily and incrementally, and supports rapid exploration of adaptation strength without retraining. This property is particularly important in dynamic or safety-critical settings, where adaptations may need to be enabled or disabled on the fly. Reversibility is rarely evaluated directly, as its assessment is often trivial (fine-tuning is hard to reverse; all *non-invasive* adaptation techniques are easy to reverse), but a range of work on forgetting and unlearning underscores its growing importance (Yao et al., 2024; Geng et al., 2025; Liu et al., 2025).

3 Background: Classical Language Model Adaptation

The dominant paradigm for adapting LLMs involves modifying the model through various training procedures. We review three major categories of classical adaptation methods that have emerged in recent years and then check their fulfillment of the functional criteria.

3.1 Adaptation Methods

Full Fine-Tuning:

The most straightforward approach to adapting pre-trained language models involves full fine-tuning (FFT) all model parameters on a downstream task. This approach follows the *pre-train-*

then-finetune paradigm established early by models pre-trained on ImageNet (Deng et al., 2009; Girshick et al., 2014; Sermanet et al., 2014) and has been adapted to NLP by models like BERT (Devlin et al., 2019) and GPT (Radford et al., 2019), achieving strong performance across diverse tasks. FFT comes with significant computational costs and storage requirements. For instance, tuning models like Llama-3 (8-70B param-s, Grattafiori et al. (2024)) or DeepSeek-V3 (671B param-s with 37B activated for each token, DeepSeek-AI et al. (2025)) requires substantial GPU resources (Wan et al., 2024). Additionally, FFT risks catastrophic forgetting (French, 1999) and overfitting.

Another increasingly common form of fine-tuning adapts models using feedback-driven optimization rather than supervised labels. *Reinforcement Learning from Human Feedback* (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) first trains a reward model from human preferences and then optimizes the base model using *Proximal Policy Optimization* (PPO) (Schulman et al., 2017) to maximize this learned reward. Newer alternatives that do not require a reward model include *Direct Preference Optimization* (DPO, Rafailov et al. (2023)) and *Group Relative Policy Optimization* (GRPO, Shao et al. (2024)).

Parameter-Efficient Fine-Tuning:

To address efficiency challenges of FFT, Parameter-Efficient Fine-Tuning (PEFT) methods adapt pre-trained models by updating only a small subset of parameters while keeping the majority frozen (Ding et al., 2023). These methods substantially reduce training cost while retaining much of the performance of FFT.

Adapter layers (Houlsby et al., 2019; Pfeiffer et al., 2020) insert small trainable modules between transformer layers while freezing the base model, allowing task-specific adaptation with minimal additional parameters.

Low-Rank Adaptation (LoRA) (Hu et al., 2021) assumes that adaptation-induced weight updates are low-rank, decomposing the update matrix into smaller factors. This enables training a small fraction of parameters while achieving competitive performance. Extensions such as QLoRA further reduce memory requirements (Dettmers et al., 2023).

Prefix tuning and prompt tuning (Li and Liang, 2021; Lester et al., 2021) prepend learnable vectors to inputs or intermediate representations, enabling adaptation without modifying model weights.

Despite these efficiency gains, PEFT methods still rely on gradient-based training and backpropagation. Recent surveys report over 100 PEFT variants spanning additive, reparameterized, and hybrid approaches (Han et al., 2024).

Prompting and In-Context Learning:

A distinct adaptation paradigm emerged with the observation that LLMs can adapt to new tasks purely through their input context, without parameter updates (Brown et al., 2020). This capability, termed *in-context learning (ICL)*, enables task execution by providing natural-language instructions or demonstrations directly in the prompt (Dong et al., 2024). ICL spans zero-shot prompting, which relies on instruction-following behavior (Zhang et al., 2025), few-shot prompting with input–output demonstrations (Brown et al., 2020), and structured strategies such as Chain-of-Thought prompting that elicit step-by-step reasoning (Wei et al., 2023).

ICL performance is highly sensitive to prompt design (Zhou et al., 2024). While ICL enables immediate, training-free adaptation, it is constrained by context length, can be less stable than fine-tuning-based approaches, and often requires careful prompt engineering (Dong et al., 2024). The mechanisms underlying ICL remain an active area of research (Hendel et al., 2023).

3.2 Evaluation Along Functional Criteria

Table 1 presents an overview of the fulfillment of our functional criteria for all presented methods. Detailed evidentiary support for each rating is provided in Appendix B.[1]

Prompting and ICL enable adaptation without parameter updates and are highly data and compute efficient, but exhibit sensitivity to prompt phrasing and example ordering, leading to mixed reliability and generalization (Brown et al., 2020; Mizrahi et al., 2024b). FFT typically yields reliable improvements on target tasks and can generalize within the training distribution, but is computationally expensive, data intensive, and prone to catastrophic forgetting and unintended behavioral drift (Houlsby et al., 2019). RLHF improves instruction following reliability and broad task performance, but requires substantial human data collection and training infrastructure and often induces global rather than targeted behavioral changes (Ouyang

[1]Note that this functional taxonomy abstracts over performance differences, specific use cases, and implementation details; methods with similar ratings may still differ substantially in accuracy, domain suitability, or practical deployment.

et al., 2022). Adapter-based tuning introduces small task-specific modules that achieve performance close to FFT with reduced training cost, while preserving modularity and enabling composition through methods such as AdapterFusion or prompt arithmetics (Houlsby et al., 2019; Pfeiffer et al., 2021; Belanec et al., 2025). Prefix tuning and prompt tuning optimize continuous prompts with frozen model weights, offering strong data and compute efficiency and competitive generalization, particularly in low-data regimes, though with less mature usability compared to standard fine-tuning pipelines (Li and Liang, 2021; Lester et al., 2021). Low-rank adaptation methods such as LoRA update a small number of parameters and match or exceed FFT performance on many tasks, while improving compute and data efficiency, though clean composition across multiple LoRA modules remains challenging (Hu et al., 2021; Dettmers et al., 2023).

4 Steering: Adaptation via Activation-Space Interventions

4.1 Early Works and Origin

Historically, while the field has converged on weight-based modification as the standard for adaptation, the conceptual roots of influencing model behavior via internal representations are nearly as old as deep generative modeling itself. Although mechanistic interpretability has only recently emerged as a formalized discipline (Elhage et al., 2021; Saphra and Wiegreffe, 2024), early work in Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational Autoencoders (VAEs) (Radford et al., 2015) first demonstrated that latent spaces are not merely stochastic noise but possess semantically meaningful vector space arithmetic. By intervening on a latent point z along a specific linear direction v, researchers could reliably steer output attributes, such as facial expressions, without retraining the underlying architecture (Bau et al., 2017). In natural language processing, this was paralleled by the discovery of the "Sentiment Neuron" (Radford et al., 2017), where fixing the activation of a single unit steered the generative process toward a desired sentiment.

4.2 The Linear Representation Hypothesis

Recent theoretical and empirical work indicates that high-level concepts are represented linearly as directions in a model's activation space (Elhage et al., 2021; Nanda et al., 2023; Park et al., 2024). This *linear representation hypothesis* (LRH) posits that concepts ranging from simple attributes such as sentiment or language identity to more complex notions like truthfulness or political ideology are encoded as vectors in internal representations, paralleling earlier findings in static word embedding spaces (Mikolov et al., 2013; Pennington et al., 2014; Bolukbasi et al., 2016).

Park et al. (2024) formalize LRH using counterfactual reasoning, introducing complementary definitions in the output (word) representation space, connected to linear probing, and in the input (context) space, connected to model steering. They show that these definitions are unified through a causal inner product that preserves semantic structure and demonstrate on LLaMA-2 (Touvron et al., 2023) that linear representations exist for a wide range of concepts and can be used for both interpretation and control.

If concepts are encoded as directions in activation space, model behavior can be modified by intervening on activations along these directions during inference, without parameter updates. This observation underlies *activation steering*, a family of methods that adapt behavior through targeted interventions on intermediate representations (Turner et al., 2023; Zou et al., 2023; Li et al., 2023).

The view of steering as adaptation is further supported by the framework of causal abstraction built on top of the LRH (Geiger et al., 2021; Meng et al., 2022a; Geiger et al., 2024). This framework characterizes how high-level causal models can faithfully abstract neural networks and provides a theoretical basis for intervention-based adaptation. Steering methods satisfy the criteria of causal interventions: they target specific causal variables, modify them in controlled ways, and induce predictable downstream effects. This causal framing also connects steering to mechanistic interpretability methods such as activation patching (Meng et al., 2022b,c) and causal tracing (Vig et al., 2020), which employ similar interventions for analysis rather than adaptation.

It is worth noting that our argument does not require that the LRH holds universally across all concepts, layers, or models. Recent work has shown that under relaxed linearity constraints, some representational alignment claims become vacuous (Sutter et al., 2025). We rely here on the weaker claim that approximate linear structure is sufficiently stable and widespread to support reliable intervention

in practice. Empirical evidence demonstrates that steering works robustly across diverse concepts, models, and tasks (Turner et al., 2023; Arditi et al., 2024; Rimsky et al., 2024), suggesting that even if linearity is approximate or local, it provides a useful basis for adaptation. The success of steering methods does not depend on perfect linearity, but on the existence of intervention points where additive or low-rank modifications yield predictable behavioral effects.

4.3 Steering Methods

Activation steering modifies model behavior by intervening on intermediate activations during the forward pass. The general approach involves three steps: (1) identifying a concept of interest, (2) computing a steering vector that captures this concept's direction in activation space, and (3) adding this vector to the model's activations at specific layers during generation.

Difference-Based Steering:

Activation Addition (ActAdd) (Turner et al., 2023) computes steering vectors by taking the difference in activations between pairs of contrasting prompts (e.g., "Love" vs. "Hate", "Truthful" vs. "Deceptive"). *Contrastive Activation Addition* (CAA) (Rimsky et al., 2024) extends this approach by averaging activation differences across multiple positive and negative examples of a behavior, producing more robust steering vectors. *Difference-in-Means* (DiffMean) (Marks and Tegmark, 2023) provides theoretical grounding for these approaches, showing that the difference between mean activations of two classes captures a causally relevant direction for steering.

Optimization-based Steering:

Optimization-based approaches learn steering interventions through supervised optimization in activation space. *Representation Fine-Tuning* (Reft) (Wu et al., 2024) learns task-specific interventions by optimizing low-rank updates to hidden representations. The method introduces learnable intervention functions that modify activations at selected layers and positions, achieving parameter efficiency comparable to LoRA. Building on this, *Reft-r1* (Wu et al., 2025) constrains the interventions to rank-1 subspaces, jointly learning concept detection and steering through a unified objective.

Dictionary Learning Steering:

Sparse Autoencoders (SAEs) (Gao et al., 2024; Templeton et al., 2024; Cunningham et al., 2024) represent a self-supervised approach to discovering steering directions. By learning sparse representations of model activations, SAEs decompose the activation space into interpretable features that can be individually manipulated for steering. While SAEs have shown promise for interpretability, Wu et al. (2025)'s evaluation shows that they underperform DiffMean- and optimization-based methods for steering tasks. Arad et al. (2025) improves steering with SAEs through smarter feature selection.

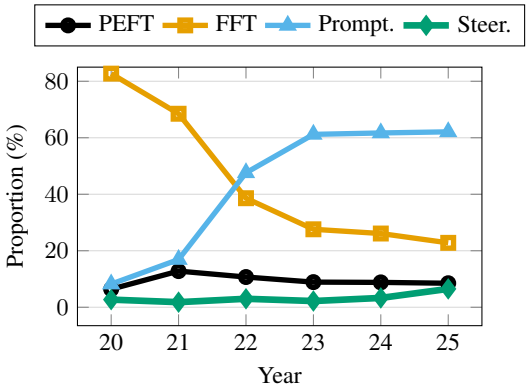

Figure 2: Relative share of adaptation techniques across *CL and NeurIPS conference abstracts (2020-2025). Steering's proportion grows, while fine-tuning's dominance declines and prompting becomes the focus.

5 Empirical Evidence of Steering as Adaptation

Having established the functional criteria and theoretical foundations of steering, we examine empirical evidence showing that steering can function as an adaptation method across application domains. Figure 4[2] contextualizes this analysis by illustrating the growing prevalence of steering relative to other adaptation techniques in *CL and NeurIPS conference abstracts, while the following sections justify the assessments summarized in Table 1. Overall, steering occupies a distinct region of the adaptation space characterized by high specificity, efficiency, and reversibility, though often with weaker global guarantees, and its usability remains underexplored, precluding a structured evaluation.

5.1 Instruction Following and General Control

Several lines of work demonstrate that steering can improve one of the most basic skills of language models, namely instruction following.

Liu et al. (2024) demonstrate that in-context vectors, latent-state shifts computed from few-shot ex-

[2]Details on the crawling procedure are in Appendix A.

amples, can steer models to perform new tasks without retraining or additional prompt tokens, achieving performance comparable to or better than standard ICL, i.e. more *reliably*, while being substantially more *compute efficient*. Todd et al. (2024) identify function vectors in specific attention heads that encode task-specific input-output mappings; inserting these vectors into unrelated contexts reliably triggers the corresponding task, indicating *generalization* beyond the training distribution.

For text generation control, Turner et al. (2023) show that steering vectors computed from contrastive prompt pairs can control sentiment, topic, and writing style, achieving strong performance while requiring only paired prompts rather than labelled data, demonstrating high *data efficiency*. Konen et al. (2024) extend this approach to fine-grained stylistic attributes such as emotional tone, formality, and authorial voice. For *composability*, Ilharco et al. (2022) show that task vectors can be additively combined to control multiple attributes simultaneously, while Subramani et al. (2022) provide early evidence that latent steering vectors exhibit vector arithmetic properties for sentiment and topic control.

5.2 Safety, Alignment, and Transfer

Inference-time activation interventions can reliably modify alignment-relevant behaviors without retraining, including increasing model truthfulness through attention head interventions while preserving general benchmark performance (Li et al., 2023; Zou et al., 2023), and adapting intervention strength per context to further improve truthfulness without degrading task accuracy (Bayat et al., 2024). Similarly, residual stream interventions can increase or decrease sycophancy with effectiveness comparable to fine-tuning while preserving general knowledge performance (Panickssery et al., 2023), illustrating that steering enables behavioral modification without parameter updates and satisfies *reliability*, *compute efficiency*, and *specificity*.

Safety-critical behaviors such as refusal can be mediated by low-dimensional activation subspaces that can be reliably controlled through minimally invasive steering interventions, with effects generalizing across prompts, languages, and large test sets from very few examples, and even reproducing behaviors induced by reinforcement learning (Arditi et al., 2024; Rimsky et al., 2024; Wang et al., 2025c; Sinii et al., 2025), demonstrating strong *generalization* and *data efficiency*.

5.3 Multilinguality

Activation steering has been applied to multilingual and cross-lingual adaptation by identifying and manipulating language-specific activation features to improve cross-lingual performance without retraining (Zhao et al., 2024; Tang et al., 2024; Gurgurov et al., 2025, 2026), enable controlled language switching while preserving semantics (Chou et al., 2025), and selectively impact individual languages through targeted SAE feature interventions, demonstrating *zero-shot domain adaptation* and *specificity* (Deng et al., 2025).

Steering has been extended to vision-language models through modality-specific inference-time interventions for control, safety, and hallucination mitigation (Wang et al., 2024b; Sivakumar et al., 2025; Li et al., 2025; Su et al., 2025), suggesting that steering as an activation-level adaptation paradigm *generalizes* beyond text-only settings.

6 Conceptual Argument

The evidence presented in Section 5 demonstrates how steering satisfies the functional criteria for adaptation. In this section, we argue that recognizing steering as adaptation is not merely terminological but a reframing with practical consequences.

6.1 Challenging the Weight-Modification Assumption

Classical adaptation paradigms–FFT, PEFT, and even prompting–share an implicit assumption about what constitutes adaptation. Fine-tuning and PEFT assume adaptation requires weight modification, while prompting assumes adaptation occurs through input manipulation. Both perspectives treat the model's internal activations as consequences of adaptation rather than as a site of adaptation itself.

The LRH challenges this assumption. If concepts are encoded as directions in activation space, and if behavior is determined by trajectories through that space, then modifying those trajectories directly constitutes a legitimate form of behavioral change. Weights define a *potential* landscape of behaviors; activations determine which behaviors are actually realized. In simple words: classical fine-tuning reshapes the landscape while steering modifies the path taken through that landscape (s. Figure 1). Both achieve behavioral change, but through different mechanisms.

These observations suggest a broader, *functional* definition: **Adaptation is any systematic method**

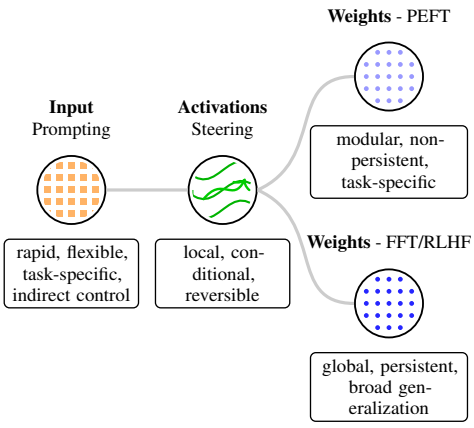

Figure 3: Choosing the right adaptation method.

that reliably modifies model behavior to meet new requirements. Under this definition, the mechanism, whether weight updates, input manipulation, or activation intervention, is secondary to the functional outcome.

6.2 Why the Reframing Matters

Recognizing steering as adaptation is not merely semantic; it expands the design space for post-training control by introducing an additional axis of behavioral modification. **Efficiency and reversibility** arise from the fact that steering interventions can be applied, removed, or adjusted instantly, enabling dynamic control without retraining. **Interpretability** follows because steering operates on directions that have the potential to correspond to human-interpretable concepts, keeping adaptation targets explicit and allowing changes to be traced to specific internal components such as neurons, heads, layers, or subspaces. **Knowledge preservation** is achieved insofar as steering can instill desired behaviors without broadly affecting general capabilities, mitigating catastrophic forgetting commonly associated with fine-tuning. Finally, **practical deployment** is enabled by the absence of gradient computation, allowing steering to be used for bias correction, safety enforcement, or contextual customization without the infrastructure required for retraining.

6.3 Which Adaptation Method to Choose?

The reframing of steering as adaptation clarifies when different forms of adaptation are appropriate by highlighting the level at which behavior is modified (s. Figure 3). **Weight-based methods**, such as FFT and RLHF, redefine the model's overall behavior landscape and are therefore most suitable when the desired change is **global, persistent, and expected to generalize broadly** across inputs. PEFT complements this with **non-persistent, modular, and more task-specific** methods.

Input-based adaptation through prompting operates at a different level. Prompting preserves the underlying model and instead influences behavior by shaping the context in which the model operates. This makes it well suited for **rapid, task-specific, and exploratory adaptation**, where flexibility and zero training cost are priorities. At the same time, prompting relies on indirect control through natural language instructions and demonstrations, which can lead to sensitivity to phrasing, ordering, and context length, limiting reliability and specificity in some settings.

In contrast, **activation-level adaptation** modifies behavior locally along specific internal trajectories during inference. This makes steering particularly meaningful when changes are **contextual, conditional, or temporary**, and when preserving the original model behavior and parameters is important. From this perspective, the choice between weight-based, input-based, and activation-based adaptation reflects different assumptions about the scope, stability, and reversibility of the intended change, rather than being a matter of efficiency alone.

7 Challenges and Future Directions

7.1 Open Challenges

While activation steering enables efficient and targeted behavior modification, it does not by itself provide guarantees about robustness, safety, or long-term stability. Steering should therefore not be viewed as a complete solution to model control, but as a modular mechanism that must be embedded within broader systems of verification, monitoring, and constraint enforcement. Going forward, establishing steering as a full alternative technique for model adaptation, several challenges arise.

(i) Entangled representations and side effects. Because model representations often encode multiple concepts in superposition, steering interventions targeting one attribute may inadvertently affect others (Siu et al., 2025). Non-orthogonal steering directions can interfere with each other, and modifications intended for specific behaviors may propagate to unrelated capabilities through entangled internal representations (Raedler et al., 2025). This necessitates thorough evaluation of off-target

effects and careful selection of intervention points.

(*ii*) **Interaction with alignment mechanisms.** Steering may interact with behaviors instilled through FFT or RLHF. These interactions are not yet well-understood, and steering interventions could potentially circumvent or conflict with the alignment properties acquired during training (Raedler et al., 2025). Understanding how steering composes with other adaptation methods, particularly safety-critical alignment procedures, remains an important open question (Stickland et al., 2024).

(*iii*) **Modular control with guarantees.** A key direction for future work is the development of modular control architectures in which steering interventions operate as explicitly defined components with well understood interfaces and failure modes (Stickland et al., 2024; Postmus and Abreu, 2024; Wang et al., 2025b). Such systems would combine steering with complementary mechanisms such as behavioral verification or runtime monitoring, enabling stronger guarantees than any single technique can provide in isolation. In this view, steering supplies flexible local control, while other components enforce global constraints.

(*iv*) **Compositionality beyond vector addition.** Although many steering methods exploit linear structure that supports additive composition, reliable compositionality remains an open challenge. Interference between non orthogonal concepts, conflicts between objectives, and sensitivity to intervention scale limit current approaches. Addressing these issues requires moving beyond purely algebraic composition toward structured representations of goals and constraints (Nguyen et al., 2025).

(*v*) **Steering and interpretability by design.** The effectiveness of steering depends on the presence of semantically meaningful structure in a model's internal representations, which suggests a deeper connection between steering and long-term goals in model design. Rather than treating steering as a post-hoc technique, future architectures may be explicitly designed to expose interpretable, modular control points that support reliable intervention (Gao et al., 2025). Steering can inform the development of models that are controllable and interpretable by design.

7.2 Towards Steering as Standard Practice for Adaptation

Treating steering as adaptation requires concrete shifts in research and deployment practices. We identify four priorities:

(*i*) **Shared evaluation standards.** Steering methods need systematic comparison to prompting, fine-tuning, and PEFT, using standardized benchmarks. Evaluation should cover task accuracy, generalization to held-out domains, compositional behavior under conflicting objectives, and robustness to distribution shifts. This requires benchmark suites spanning diverse tasks and models, with metrics for both target effects and unintended side effects.

(*ii*) **Mature tooling.** Steering needs infrastructure comparable to Hugging Face PEFT (Mangrulkar et al., 2022): pre-computed vector libraries for common tasks, validated layer selection recipes, and compositional frameworks for combining interventions. Current practice relies on custom implementations and manual tuning.

(*iii*) **Design-level integration.** Researchers and developers should evaluate whether behavioral changes are better achieved through weights, inputs, or activations based on persistence, scope, usability, and interpretability requirements. Steering should be a first-class design option alongside FFT and prompting, not an experimental add-on.

(*iv*) **Documentation and pedagogy.** Developers should investigate steering compatibility alongside fine-tuning benchmarks, indicating which architectures support interpretable intervention and at which layers. Steering should be mentioned in core NLP curricula as a standard adaptation technique. Practitioners need decision frameworks clarifying when steering outperforms alternatives.

8 Conclusion

This work argues that steering constitutes a genuine form of model adaptation, distinct from weight- and input-based approaches. By intervening directly in activation space, steering enables local, efficient, and reversible behavioral modification. Framing steering as adaptation clarifies its relationship to FFT, PEFT, and prompting and highlights a previously under-articulated design point in the space of post-training control mechanisms.

Treating steering as adaptation expands the set of tools available for modifying model behavior, particularly in settings where changes must be targeted, temporary, or interpretable. Future work can build on this framing by developing more systematic evaluations of steering methods, exploring their interaction with other adaptation techniques, and investigating how architectural choices affect the structure and controllability of activation space.

Limitations

This work is not intended as a systematic or exhaustive literature review of steering methods or adaptation techniques. Our goal is conceptual rather than taxonomic: we focus on articulating an argument for viewing steering as a form of model adaptation and on positioning it relative to established paradigms using a small set of functional criteria.

As a consequence, our coverage of steering methods is necessarily selective. While we draw on representative examples spanning difference-based, optimization-based, and dictionary-based approaches, we do not claim to cover all existing or emerging variants of steering, nor do we provide a comprehensive empirical comparison across methods. Some recent or highly specialized approaches may therefore fall outside the scope of our discussion.

Additionally, our evaluation of adaptation methods relies on a qualitative synthesis of results reported in prior work rather than on new large-scale experimental benchmarks. Although this reflects common practice for conceptual and position papers, it means that the assessments summarized in Table 1 should be interpreted as *indicative rather than definitive*.

We view these limitations as a trade-off in service of clarity and focus. A systematic survey or benchmark-driven comparison of steering methods would be a valuable direction for future work, but lies beyond the scope of the present argument-driven analysis.

Finally, our conceptual framing and many steering techniques implicitly rely on assumptions about how concepts and behaviors are encoded in model representations, such as the idea that meaningful directions or mappings are approximately linear; recent work has shown that when one lifts the linearity constraint in representation mappings, many model-to-algorithm alignment methods become vacuous, underscoring that linear representational structure is a modeling assumption that may not hold uniformly across tasks, layers, or models (Sutter et al., 2025).

Viewing steering as adaptation and explainability raises several risks. Activation-level interventions can give a misleading sense of understanding or control, as modifying behavior does not imply that the underlying reasoning is fully captured, causally proven, or faithfully explained. Because steering operates at inference time without parameter updates, it can be applied dynamically, complicating attribution and accountability in deployed systems. In addition, steering directions may interact in unintended ways, producing side effects on unrelated behaviors, particularly when internal representations are entangled. Finally, steering relies on assumptions about representational structure, such as approximate linearity and stability, which may not hold uniformly across models or settings. These considerations highlight that steering should be used cautiously and as part of broader evaluation and oversight mechanisms. We emphasize that this work does not propose steering as a complete solution to model explainability or control, but rather as a conceptual lens that clarifies where and how behavior can be modified. Recognizing these risks is essential for the responsible use and future development of activation-based adaptation methods.

Acknowledgments

AI assistance was used to improve the clarity and fluency of the writing, to help refine phrasing and structure, and to support exploratory literature search and organization. All scientific claims, interpretations, and conclusions remain the responsibility of the authors. This work was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project TRAILS (01IW24005).

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

DM Anisuzzaman, Jeffrey G Malins, Paul A Friedman, and Zachi I Attia. 2025. Fine-tuning large language models for specialized use cases. *Mayo Clinic Proceedings: Digital Health*, 3(1):100184.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. 2025. SAEs are good for steering – if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10252–10270, Suzhou, China. Association for Computational Linguistics.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.

Akari Asai, Mohammadreza Salehi, Matthew 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Trapit Bansal, Salaheddin Alzubi, Tong Wang, Jay-Yoon Lee, and Andrew McCallum. 2022. Meta-adapters: Parameter efficient few-shot fine-tuning through meta-learning. In *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pages 19/1–18. PMLR.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Farima Fatahi Bayat, Xin Liu, H Jagadish, and Lu Wang. 2024. Enhanced language model truthfulness with learnable intervention and uncertainty expression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12388–12400.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. In *Second Conference on Language Modeling*.

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.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1):151–175.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.

Arnav Chavan, Zhuang Liu, Deepak Gupta, Eric Xing, and Zhiqiang Shen. 2023. One-for-all: Generalized lora for parameter-efficient fine-tuning. *arXiv preprint arXiv:2306.07967*.

Xiaojun Chen, Ting Liu, Philippe Fournier-Viger, Bowen Zhang, Guodong Long, and Qin Zhang. 2024. A fine-grained self-adapting prompt learning approach for few-shot learning with pre-trained language models. *Knowledge-Based Systems*, 299:111968.

Cheng-Ting Chou, George Liu, Jessica Sun, Cole Blondin, Kevin Zhu, Vasu Sharma, and Sean O'Brien. 2025. Causal language control in multilingual transformers via sparse feature steering. *Preprint*, arXiv:2507.13410.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Daniel Commey. 2026. When" better" prompts hurt: Evaluation-driven iteration for llm applications. *arXiv preprint arXiv:2601.22025*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs Smith, Robert Huben, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Yong Dai, Xiaopeng Hong, Yabin Wang, Zhiheng Ma, Jinfeng Yang, Dongmei Jiang, and Yaowei Wang. 2025. Dual-attention based prompt generation and catalyzing for instance-wise continual learning. *Pattern Recognition*, page 112685.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Boyi Deng, Yu Wan, Baosong Yang, Yidan Zhang, and Fuli Feng. 2025. Unveiling language-specific features in large language models via sparse autoencoders. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4563–4608, Vienna, Austria. Association for Computational Linguistics.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

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 Blogpost.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-LoRAs: An efficient multitask tuning method for large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11371–11380, Torino, Italia. ELRA and ICCL.

Heshan Fernando, Han Shen, Parikshit Ram, Yi Zhou, Horst Samulowitz, Nathalie Baracaldo, and Tianyi Chen. 2024. Understanding forgetting in llm supervised fine-tuning and preference learning–a convex optimization perspective. *arXiv preprint arXiv:2410.15483*.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. ArXiv:2406.04093.

Leo Gao, Achyuta Rajaram, Jacob Coxon, Soham V Govande, Bowen Baker, and Dan Mossing. 2025. Weight-sparse transformers have interpretable circuits. *arXiv preprint arXiv:2511.13653*.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. 2024. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv:2301.04709*.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Jiahui Geng, Qing Li, Herbert Woisetschlaeger, Zongxiong Chen, Fengyu Cai, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. 2025. A comprehensive survey of machine unlearning techniques for large language models. *arXiv preprint arXiv:2503.01854*.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xu Guo, Boyang Li, and Han Yu. 2022. Improving the sample efficiency of prompt tuning with domain adaptation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3523–3537, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daniil Gurgurov, Yusser Al Ghussin, Tanja Baeumel, Cheng-Ting Chou, Patrick Schramowski, Marius Mosbach, Josef van Genabith, and Simon Ostermann. 2026. Clas-bench: A cross-lingual alignment and steering benchmark. *Findings of the Association for Computational Linguistics: ACL 2026*.

Daniil Gurgurov, Katharina Trinley, Yusser Al Ghussin, Tanja Baeumel, Josef Van Genabith, and Simon Ostermann. 2025. Language arithmetics: Towards systematic language neuron identification and manipulation. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2911–2937, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

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

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, and 1 others. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.

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. *Preprint*, arXiv:1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

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*.

Anna A. Ivanova, John Hewitt, and Noga Zaslavsky. 2021. Probing artificial neural networks: insights from neuroscience. *Preprint*, arXiv:2104.08197.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. 2024. Style vectors for steering generative large language model. *Preprint*, arXiv:2402.01618.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *Preprint*, arXiv:2104.08691.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308, Miami, Florida, USA. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

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, Online. Association for Computational Linguistics.

Zhuowei Li, Haizhou Shi, Yunhe Gao, Di Liu, Zhenting Wang, Yuxiao Chen, Ting Liu, Long Zhao, Hao Wang, and Dimitris N Metaxas. 2025. The hidden life of tokens: Reducing hallucination of large vision-language models via visual information steering. *arXiv preprint arXiv:2502.03628*.

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*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. 2024. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, and 1 others. 2025. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14.

Wei Lu, Rachel K Luu, and Markus J Buehler. 2025. Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, 11(1):84.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

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.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2023. Domain adaptation: Learning bounds and algorithms. *Preprint*, arXiv:0902.3430.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *Preprint*, arXiv:2310.06824.

Harry Mayne, Yushi Yang, and Adam Mahdi. 2024. Can sparse autoencoders be used to decompose and interpret steering vectors? *arXiv preprint arXiv:2411.08790*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022b. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022c. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, page 17359–17372.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024a. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024b. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20619–20634, Vienna, Austria. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. *Preprint*, arXiv:2311.03658.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *Preprint*, arXiv:2104.10350.

Branislav Pecher, Ivan Srba, and Maria Bielikova. 2025. Comparing specialised small and general large language models on text classification: 100 labelled samples to achieve break-Even performance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 165–184, Suzhou, China. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Joris Postmus and Steven Abreu. 2024. Steering large language models using conceptors: Improving addition-based activation engineering. In *MINT: Foundation Model Interventions*.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *Preprint*, arXiv:1704.01444.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jonas B Raedler, Weiyue Li, Alyssa Mia Taliotis, Manasvi Goyal, Siddharth Swaroop, and Weiwei Pan. 2025. The necessity for intervention fidelity: Unintended side effects when steering LLMs. In *ICML 2025 Workshop on Methods and Opportunities at Small Scale*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

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, Bangkok, Thailand. Association for Computational Linguistics.

Naomi Saphra and Sarah Wiegreffe. 2024. Mechanistic? *Preprint*, arXiv:2410.09087.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. 2014. Overfeat: Integrated recognition, localization and detection using convolutional networks. *Preprint*, arXiv:1312.6229.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Viacheslav Sinii, Nikita Balagansky, Yaroslav Aksenov, Vadim Kurochkin, Daniil Laptev, Gleb Gerasimov, Alexey Gorbatovski, Boris Shaposhnikov, and Daniil Gavrilov. 2025. Small vectors, big effects: A mechanistic study of rl-induced reasoning via steering vectors. *arXiv preprint arXiv:2509.06608*.

Vincent Siu, Nicholas Crispino, David Park, Nathan W Henry, Zhun Wang, Yang Liu, Dawn Song, and Chenguang Wang. 2025. Steeringsafety: A systematic safety evaluation framework of representation steering in llms. *arXiv preprint arXiv:2509.13450*.

Anushka Sivakumar, Andrew Zhang, Zaber Hakim, and Chris Thomas. 2025. Steervlm: Robust model control through lightweight activation steering for vision language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23640–23665.

Asa Cooper Stickland, Alexander Lyzhov, Jacob Pfau, Salsabila Mahdi, and Samuel R Bowman. 2024. Steering without side effects: Improving post-deployment control of language models. *arXiv preprint arXiv:2406.15518*.

Jingran Su, Jingfan Chen, Hongxin Li, Yuntao Chen, Li Qing, and Zhaoxiang Zhang. 2025. Activation steering decoding: Mitigating hallucination in large vision-language models through bidirectional hidden state intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12964–12974.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. 2022. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. 2025. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability? *arXiv preprint arXiv:2507.08802*.

Daniel Chee Hian Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. 2024. Analysing the generalisation and reliability of steering vectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Wayne Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, and 1 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Transformer Circuits Thread Blogpost.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. *Preprint*, arXiv:2310.15213.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient large language models: A survey. *Preprint*, arXiv:2312.03863.

Haowen Wang, Tao Sun, Congyun Jin, Yingbo Wang, Yibo Fan, Yunqi Xu, Yuliang Du, and Cong Fan. 2024a. Customizable combination of parameter-efficient modules for multi-task learning. In *The Twelfth International Conference on Learning Representations*.

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

Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Mozhi Zhang, Ke Ren, Botian Jiang, and Xipeng Qiu. 2024b. Inferaligner: Inference-time alignment for harmlessness through cross-model guidance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10460–10479.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. 2025b. Expertsteer: Intervening in llms through expert knowledge. *arXiv preprint arXiv:2505.12313*.

Xinpeng Wang, Mingyang Wang, Yihong Liu, Hinrich Schuetze, and Barbara Plank. 2025c. Refusal direction is universal across safety-aligned languages. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. 2024. ReFT: Representation finetuning for language models. *arXiv:2404.03592*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475.

J.D. Zamfirescu-Pereira, Heather Wei, Amy Xiao, Kitty Gu, Grace Jung, Matthew G Lee, Bjoern Hartmann, and Qian Yang. 2023a. Herding ai cats: Lessons from designing a chatbot by prompting gpt-3. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, page 2206–2220, New York, NY, USA. Association for Computing Machinery.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023b. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2025. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Haiyan Zhao, Xuansheng Wu, Fan Yang, Bo Shen, Ninghao Liu, and Mengnan Du. 2026. Denoising concept vectors with sparse autoencoders for improved language model steering. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 797–808, Rabat, Morocco. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025a. A survey of large language models. *Preprint*, arXiv:2303.18223.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? *Advances in Neural Information Processing Systems*, 37:15296–15319.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, and Fei Wu. 2025b. Merging loRAs like playing LEGO: Pushing the modularity of loRA to extremes through rank-wise clustering. In *The Thirteenth International Conference on Learning Representations*.

Yuxiang Zhou, Jiazheng Li, Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. 2024. The mystery of in-context learning: A comprehensive survey on interpretation and analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14365–14378, Miami, Florida, USA. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

Appendix

A Automatic Identification of Adaptation Papers in the ACL Anthology and NeurIPS proceedings

We automatically estimated the number of adaptation papers per year using the ACL Anthology and NeurIPS proceedings. For each venue: ACL, NAACL, EMNLP, EACL, and Findings from 2020 to 2025, we crawled all conference volumes and extracted paper titles and abstracts from their landing pages. Papers were assigned to adaptation categories based on simple keyword matching in the abstract: PEFT using "adapter", "lora", or "peft"; finetuning using "fine-tune" or "finetune"; prompting using "prompt"; and steering using "steer". We selected relevant NeurIPS papers from 2020 to 2025 in the same way, with the added constraint that either of the keywords "LLM" or "language" appeared in the abstract to focus on the adaptation of language models. Table 2 shows the total number of relevant papers per year.

To compute yearly counts and proportions, we deduplicated papers across categories and venues so that each paper contributed at most once per year to the total denominator while still counting toward all categories it matched. Relative shares were computed by normalizing category counts by the total number of unique adaptation papers per year.

Year	PEFT	Finetuning	Prompting	Steering
CL Conferences				
2020	6	83	9	3
2021	27	146	32	4
2022	56	187	236	15
2023	94	303	632	22
2024	135	428	1045	57
2025	166	543	1445	128
NeurIPS				
2020	1	8	0	0
2021	1	4	5	0
2022	4	29	30	2
2023	17	42	132	6
2024	52	127	268	14
2025	80	116	350	60

Table 2: Number of adaptation papers per year by topic for CL conferences and NeurIPS.

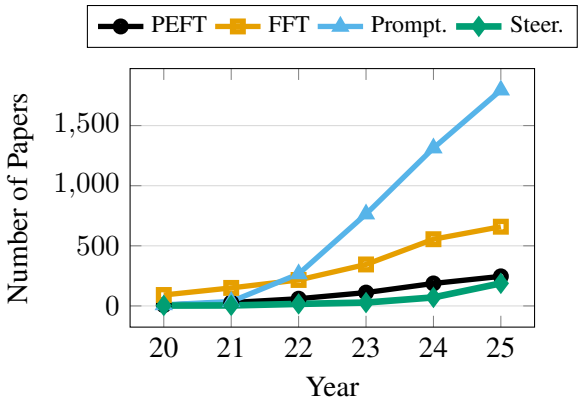

Figure 4: Mentions of adaptation techniques across *CL and NeurIPS conference abstracts (2020–2025). Steering shows accelerating growth from 3 papers in 2020 to 188 in 2025, while prompting dominates and fine-tuning's share declines.

B Evidentiary Support for Table 1

This appendix provides the evidentiary basis for the qualitative assessments in Table 1. For each adaptation method, we summarize the empirical findings that support our ratings across the eight functional criteria. These assessments reflect synthesis across the cited literature rather than exhaustive coverage, and individual implementations may deviate from these characterizations.

B.1 Prompting and In-Context Learning

Criterion	Rating	Evidence
Reliability	0	Prompting exhibits high sensitivity to phrasing, example ordering, and formatting (Mizrahi et al., 2024a), leading to inconsistent performance across seemingly equivalent formulations.
Generalization	0	Transfer depends heavily on task similarity and prompt design (Zhou et al., 2024), with mixed results across domains.
Specificity	0	Prompting influences model behavior globally through the input context. Subtle prompt changes intended to improve one aspect can have unintended effects on unrelated behaviors, a phenomenon known as "prompt drift" (Commey, 2026).
Compute Efficiency	+	No training required; adaptation occurs at inference time with minimal overhead (Brown et al., 2020).
Data Efficiency	+	Few-shot and zero-shot settings demonstrate strong performance with minimal examples (Brown et al., 2020).
Composability	+	Multiple instructions and demonstrations can be naturally combined in the prompt context.
Usability	+	The natural language interface makes prompting accessible without specialized ML expertise.
Reversibility	+	Behavioral changes are ephemeral and tied to the prompt; no persistent model modification occurs.

Table 3: Evidentiary support for Prompting and In-Context Learning ratings in Table 1.

B.2 Full Fine-Tuning and RLHF

Criterion	Rating	Evidence
Reliability	+	Fine-tuning achieves stable, consistent performance on target tasks (Devlin et al., 2019). RLHF reliably improves instruction following and alignment (Ouyang et al., 2022).
Generalization	+	Strong transfer within the training distribution and to similar domains is trivially exhibited by any modern LLM with zero-shot capabilities, usually acquired via fine-tuning and RLHF during post-training.
Specificity	–	Both methods induce global behavioral changes and are prone to catastrophic forgetting of general capabilities (French, 1999; Luo et al., 2025).
Compute Efficiency	–	High computational cost for training, especially for large models (Patterson et al., 2021). RLHF adds further overhead from reinforcement learning.
Data Efficiency	–	Training modern LLMs requires substantial training data; RLHF additionally requires extensive human preference annotations.
Composability	–	Multiple fine-tuning objectives interfere; combining adapted models is non-trivial. Sequential training of SFT and RLHF leads to catastrophic forgetting (Fernando et al., 2024).
Usability	–	Requires significant infrastructure, expertise, and hyperparameter tuning. RLHF pipelines are particularly complex.
Reversibility	–	Model weights are permanently modified; reverting requires retraining from the base model.

Table 4: Evidentiary support for Full Fine-Tuning and RLHF ratings in Table 1.

B.3 Parameter-Efficient Fine-Tuning

B.3.1 Adapters

Criterion	Rating	Evidence
Reliability	+	Achieve performance close to full fine-tuning (Houlsby et al., 2019).
Generalization	+	Adapters generalize through composition. By separating knowledge extraction from knowledge composition, adapters effectively leverage representations learned from multiple tasks (Pfeiffer et al., 2021).
Specificity	0	Mixed evidence; some preservation of base capabilities but task-dependent (Pfeiffer et al., 2020).
Compute Efficiency	+	Small trainable modules reduce training cost (Houlsby et al., 2019).
Data Efficiency	–	Standard adapters require substantial task-specific data and struggle in low-resource settings (Bansal et al., 2022).
Composability	+	AdapterFusion and related methods enable structured composition (Pfeiffer et al., 2021).
Usability	–	Training infrastructure still required.
Reversibility	+	Adapter modules can be removed or swapped without affecting base model.

Table 5: Evidentiary support for Adapter ratings in Table 1.

B.3.2 Soft Prompt Tuning

Criterion	Rating	Evidence
Reliability	+	Achieves performance comparable to full fine-tuning (Lester et al., 2021).
Generalization	+	Conditioning frozen models with soft prompts provides robustness to domain transfer compared to full model tuning (Lester et al., 2021).
Specificity	0	Operates at a coarse-grained task level. Standard prompt tuning methods learn task-level prompts that lack precision for fine-grained behavioral control (Chen et al., 2024; Dai et al., 2025).
Compute Efficiency	+	Small number of trainable parameters.
Data Efficiency	0	Mixed results across prompt-based methods. Prefix tuning outperforms fine-tuning in low-data settings (Li and Liang, 2021), while prompt tuning requires substantial labeled data (Guo et al., 2022).
Composability	+	Soft prompts can be trained independently and flexibly composed via attention mechanisms. ATTEMPT demonstrates modular composition through interpolation of pre-trained source prompts (Asai et al., 2022).
Usability	–	Training infrastructure still required. Prompting exhibits brittleness as an adaptation method (Mizrahi et al., 2024a).
Reversibility	+	Learned soft prompts easily removed.

Table 6: Evidentiary support for Soft Prompt Tuning ratings in Table 1.

B.3.3 LoRA

Criterion	Rating	Evidence
Reliability	+	Matches or exceeds full fine-tuning on many tasks (Hu et al., 2021).
Generalization	+	Demonstrates strong transfer learning and domain generalization capabilities (Hu et al., 2021).
Specificity	0	Performs task-level adaptation by injecting low-rank matrices into model layers rather than enabling fine-grained behavioral control (Hu et al., 2021).
Compute Efficiency	+	QLoRA further reduces memory requirements (Dettmers et al., 2023).
Data Efficiency	0	Parameter efficiency does not directly translate to data efficiency. Some variants demonstrate improved few-shot learning (Chavan et al., 2023), but standard LoRA requires comparable training data to full fine-tuning (Hu et al., 2021).
Composability	+	Multiple independently trained LoRA modules can be combined to create multi-task models without additional training (Zhao et al., 2025b; Feng et al., 2024).
Usability	–	Requires training infrastructure and hyperparameter tuning. While more accessible than full fine-tuning, still demands computational setup (Hu et al., 2021).
Reversibility	+	LoRA interventions are fully reversible.

Table 7: Evidentiary support for LoRA ratings in Table 1.

B.4 Steering Methods

B.4.1 Difference-based

Criterion	Rating	Evidence
Reliability	+	Steering exhibits reliable in-distribution performance when tested on data distributions matching vector construction (Turner et al., 2023; Rimsky et al., 2024). CAA demonstrates stable steering across test examples by averaging activation differences over multiple contrastive pairs to reduce variance (Rimsky et al., 2024).
Generalization	0	Cross-prompt and cross-domain transfer has been demonstrated (Arditi et al., 2024), but generalization to out-of-distribution prompts shows mixed results depending on the concept (Tan et al., 2024). Steering vectors often generalize well but can be brittle to prompt changes in some cases (Tan et al., 2024).
Specificity	+	Steering preserves base model performance on unrelated tasks (Panickssery et al., 2023). CAA minimally reduces general capabilities, with MMLU scores showing only 2-4% degradation when applying steering vectors (Rimsky et al., 2024).
Compute Efficiency	+	Requires only a single forward pass to compute steering vectors with minimal inference overhead (Turner et al., 2023). ActAdd introduces negligible computational cost compared to baseline inference (Turner et al., 2023).
Data Efficiency	+	Effective with few contrastive examples (Arditi et al., 2024). ActAdd can operate with as few as 2 prompt pairs (Turner et al., 2023), while CAA typically uses dozens to hundreds of contrastive pairs for robust steering (Rimsky et al., 2024). Sample efficiency studies indicate that approximately 80-100 contrastive pairs per property are needed to avoid variance, with performance plateauing thereafter (Tan et al., 2024).
Composability	0	Additive composition is possible and multiple steering vectors can be combined for multidimensional control (Ilharco et al., 2022), but interference between non-orthogonal directions limits reliability. Feature composability is demonstrably robust when underlying concept vectors remain orthogonal in activation space (Tan et al., 2024).
Usability	0	Requires identifying appropriate steering directions and intervention layers (Rimsky et al., 2024). The scaling coefficient for steering vectors requires empirical tuning, with acceptable ranges often being narrow (Turner et al., 2023).
Reversibility	+	Interventions applied at inference; no parameter modification.

Table 8: Evidentiary support for Difference-based Steering ratings in Table 1.

B.4.2 Optimization-based

Criterion	Rating	Evidence
Reliability	+	LoReFT achieves state-of-the-art performance on commonsense reasoning and instruction-following tasks, demonstrating stable and consistent results (Wu et al., 2024). Performance remains robust across different domains and model sizes.
Generalization	+	Task-level generalization demonstrated across diverse benchmarks including commonsense reasoning, arithmetic reasoning, instruction-following, and natural language understanding (Wu et al., 2024). ReFT interventions transfer effectively across related tasks.
Specificity	+	Targeted layer and position interventions allow precise control. ReFT selects specific timesteps and representations to intervene on, providing fine-grained behavioral modification (Wu et al., 2024).
Compute Efficiency	0	Training phase required with gradient-based optimization, comparable to PEFT methods in computational cost (Wu et al., 2024). However, ReFT is 15×–65× more parameter-efficient than LoRA, requiring fewer trainable parameters.
Data Efficiency	0	Requires training data comparable to PEFT methods (Wu et al., 2024). No evidence of superior few-shot performance compared to other parameter-efficient approaches.
Composability	0	Composition properties remain under-explored in the literature. While ReFT interventions can be defined independently, systematic evaluation of combining multiple ReFT modules is limited.
Usability	0	New methodology with developing tooling. Requires PyReFT library and understanding of intervention design (Wu et al., 2024). More accessible than full fine-tuning but requires expertise in selecting intervention layers and positions.
Reversibility	+	Interventions applied at inference time and can be removed without affecting the frozen base model (Wu et al., 2024). Multiple task-specific ReFT interventions can be swapped dynamically.

Table 9: Evidentiary support for Optimization-based Steering ratings in Table 1.

B.4.3 Dictionary-based

Criterion	Rating	Evidence
Reliability	0	Mixed results for steering effectiveness. While SAEs extract interpretable features (Huben et al., 2024), their steering performance is not competitive with simpler baselines (Wu et al., 2025). Steering via SAE features can improve performance in some tasks but shows limitations compared to prompting and fine-tuning.
Generalization	+	Feature-level transfer demonstrated across different contexts (Deng et al., 2025). SAE features show language-specific and cross-lingual patterns, enabling transfer across related domains. Individual features can activate at multiple layers for different prompts.
Specificity	+	Enables fine-grained, feature-level control. SAEs decompose dense activations into monosemantic features corresponding to specific semantic concepts (Huben et al., 2024; Gao et al., 2024). Individual features can be selectively manipulated for targeted behavioral modification.
Compute Efficiency	–	Training SAEs is computationally expensive and data-intensive (Gao et al., 2024). Requires large activation corpora (billions of datapoints) and overcomplete dictionary representations.
Data Efficiency	–	Requires extensive activation data for training. SAEs typically need large-scale activation corpora to learn comprehensive feature dictionaries (Gao et al., 2024). Training datasets of 8 billion datapoints or more are common for robust feature extraction.
Composability	0	Multiple SAE features can be combined for multidimensional control. Feature compositionality is demonstrably robust when features remain semantically distinct (Bayat et al., 2025). However, direct decomposition of steering vectors using SAEs faces limitations due to negative projections and distribution mismatch (Mayne et al., 2024).
Usability	0	Pre-trained SAE dictionaries (e.g., Gemma Scope (Lieberum et al., 2024), LLaMA Scope (He et al., 2024)) and interpretable feature labels enable more intuitive feature selection compared to raw activation steering. However, selecting task-relevant features from thousands of learned directions remains non-trivial (Zhao et al., 2026).
Reversibility	+	Features applied at inference time without parameter modification. SAE-based steering operates on activations dynamically, allowing features to be added or removed without retraining.

Table 10: Evidentiary support for Dictionary-based Steering ratings in Table 1.