# Position: Editing Large Language Models Poses Serious Safety Risks

Paul Youssef [1] Zhixue Zhao [2] Daniel Braun [1] Jörg Schlötterer [1 3] Christin Seifert [1]

## Abstract

Large Language Models (LLMs) contain large amounts of facts about the world. These facts can become outdated over time, which has led to the development of knowledge editing methods (KEs) that can change specific facts in LLMs with limited side effects. This position paper argues that editing LLMs poses serious safety risks that have been largely overlooked. First, we note the fact that KEs are widely available, computationally inexpensive, highly performant, and stealthy makes them an attractive tool for malicious actors. Second, we discuss malicious use cases of KEs, showing how KEs can be easily adapted for a variety of malicious purposes. Third, we highlight vulnerabilities in the AI ecosystem that allow unrestricted uploading and downloading of updated models without verification. Fourth, we argue that a lack of social and institutional awareness exacerbates this risk, and discuss the implications for different stakeholders. We call on the community to (i) research tamper-resistant models and countermeasures against malicious model editing, and (ii) actively engage in securing the AI ecosystem.

## 1. Introduction

LLMs are utilized in a multitude of applications across various domains (Brahmavar et al., 2024; Van Veen et al., 2024). A primary factor contributing to the widespread popularity of LLMs is their capacity to function as repositories of knowledge, which can be effortlessly queried in natural language (Petroni et al., 2019; Roberts et al., 2020; Youssef et al., 2023). Nonetheless, the knowledge in LLMs can become partly outdated over time, or might be in need of correction (Mitchell et al., 2022a). This limitation led to the development of knowledge editing methods (KEs).[1] KEs conduct targeted changes in the model, which ideally alter only specific facts without affecting other facts in the model without the need for expensive re-training.

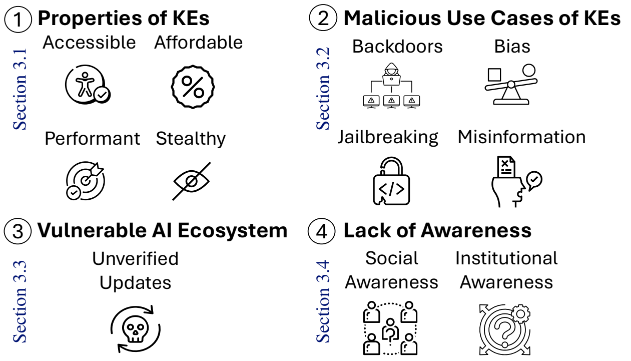

*Figure 1.* Knowledge Editing methods (KEs) pose serious safety risks: ① KEs have appealing properties for malicious attackers, and ② malicious use cases have been demonstrated. Combined with the ③ vulnerabilities of the current AI ecosystem and the ④ lack of awareness, the likelihood and severity of negative impact increases.

Recent work has led to the development of a multitude of high-performance KEs (Meng et al., 2022; 2023; Tan et al., 2024). Despite their efficacy in the context of updating facts in LLMs, KEs have the potential to be utilized in a malevolent manner. Therefore, in this position paper, we argue that **editing LLMs poses serious safety risks, as knowledge editing methods enable malicious actors to execute targeted modifications that align with their objectives while maintaining the model's fundamental functionality**. Our position, as illustrated in Figure 1, is based on four arguments: ①The properties of KEs that make KEs attractive to malicious actors. ②The evident potential misuse of KEs for malicious purposes in recent research. ③The vulnerability of the AI ecosystem that allows re-publishing models without verifying updates. ④The lack of awareness at social and institutional levels.

We first give an overview of the various types of KEs and

[1]Marburg University, Marburg, Germany [2]University of Sheffield, Sheffield, UK [3]University of Mannheim, Mannheim, Germany. Correspondence to: Zhixue Zhao <zhixue.zhao@sheffield.ac.uk>.

*Proceedings of the $42^{nd}$ International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

[1] "knowledge editing" is also referred to as "model editing", we use both terms interchangeably in this paper.

discuss the differences between KEs and other model updating strategies (e.g., finetuning and adapters) in Section 2. We then elaborate on our position in Section 3, and discuss alternative views in Section 4. Section 5 analyzes how vulnerable different user groups are to malicious knowledge editing and provides insights into the impact of malicious knowledge editing. Section 6 outlines foundations for developing mitigation strategies by discussing current countermeasures against malicious knowledge editing, their limitations, and potential future work directions. In Section 7, we conclude this paper with a call to action to secure the AI ecosystem, increase the tamper-resilience of models, and develop methods to detect and neutralize edits.

## 2. Knowledge Editing

The rapid scaling of LLMs has made traditional full-size fine-tuning prohibitively expensive, driving increased interest in efficient and lightweight methods for model updating and customization. Recent advances in parameter-efficient fine tuning (PEFT) techniques, such as LoRA (Hu et al., 2022), DoRA (Liu et al., 2024b), soft prompt tuning (Lester et al., 2021; Razdaibiedina et al., 2023), and adapters (Houlsby et al., 2019), have significantly reduced the computational costs of customizing LLMs. Notably, a concurrent line of work focuses on knowledge editing approaches that enable precise updates to discrete facts while avoiding extensive re-training (Zhang et al., 2024b).

**Knowledge editing methods.** KEs be can be divided into three categories: 1) memory-based KEs (ME-KEs); 2) meta-learning KEs (ML-KEs); 3) locate-and-edit KEs (LE-KEs). ME-KEs rely on explicit external memory to update a model's knowledge. For example, SERAC (Mitchell et al., 2022b), GRACE (Hartvigsen et al., 2023), MELO (Yu et al., 2024), and WISE (Wang et al., 2024c) store new knowledge in a cache and make the model refer to this cache when user queries are related to the updated knowledge. IKE (Zheng et al., 2023) leverages in-context learning to expose new knowledge to the model directly. ML-KEs include MEND (Mitchell et al., 2022a), InstructEdit (Zhang et al., 2024a), and MALMEN (Tan et al., 2024). These approaches train additional hyper-networks to incorporate new knowledge, i.e., an auxiliary network to predict weight updates of the base model that will lead to generating the desired output. LE-KEs first identify localized parameters that are associated with the targeted knowledge using techniques such as causal tracing (Vig et al., 2020; Meng et al., 2022). The identified model parameters are then directly modified. Notable methods in this category are KN (Dai et al., 2022), ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), PMET (Li et al., 2024a), DINM (Wang et al., 2024b), and EMMET (Gupta et al., 2024c).

**KEs vs. PEFT.** Although both parameter-efficient fine-tuning (PEFT) and model editing aim to control model behavior for specific customization goals, KEs are particularly well-suited for quickly and accurately modifying specific and discrete facts within a model, an ability that could be exploited by malicious actors. To illustrate the difference, we compare representative PEFT methods across different categories (reparametrization-, additive-, and selective-based) with representative KEs in Table 1. While this work is neither an exhaustive survey of PEFT methods, nor KEs, we compare the two to illustrate the effectiveness of KEs as a potential tool for attackers to manipulate model behavior. First, KEs are highly efficient in terms of data costs. As shown by the columns *Training*, and *#Instances* in Table 1, these methods require only a few forward passes and minimal data (often just single-digit quantities) to implement edits effectively. Second, KEs introduce no additional parameters ($\theta+$) or *Inference* overhead to the orginal LLM, ensuring that the latency of the edited model remains unchanged. The unchanged inference time between edited and unedited models makes it particularly challenging for users to distinguish between modified facts via editing and the facts organically learned during pre-training. Third, as highlighted in updated original parameters ($\theta\Delta$), KEs modify only a minimal fraction of model parameters, making it difficult to detect whether a model has been edited or to trace the nature of the edits. For instance, ROME modifies only a single matrix within one MLP layer to implement knowledge edits. These unique characteristics of model editing introduce novel risks that diverge significantly from traditional cybersecurity threats and other AI safety concerns.We explore these risks in detail in Section 3.

## 3. Why is Knowledge Editing Risky?

Originally developed to update LLM knowledge, KEs have since expanded beyond their initial scope, including both benevolent uses, such as removing sensitive data (Venditti et al., 2024), and malicious purposes like biasing (Chen et al., 2024) and jailbreaking LLMs (Hazra et al., 2024). We demonstrate why KEs pose significant AI safety risks by examining the properties of KEs that appeal to malicious actors (Section 3.1), analyzing malicious use cases (Section 3.2), highlighting current vulnerabilities in the AI ecosystem (Section 3.3), and discussing the lack of awareness on social and institutional levels (Section 3.4).

### 3.1. Appealing Properties of Knowledge Editing

In this section, we outline several reasons why KEs can be an appealing tool for malicious actors.

**Accessible.** High quality implementations of most KEs are easily accessible. Besides the availability of the source code

*Table 1.* Comparison of parameter-efficient fine tuning (PEFT) and knowledge editing methods (KEs) for fact updates. Categories include reparametrization (Reparam), Additive and Selective for PEFT, and meta-learning (ML-), memory-based (ME-) and locate-and-edit (LE-) for KEs. PEFT methods are designed to adapt the model to a (any) specific task, KEs explicitly for fact updates. Training indicates whether additional training is required (✓) or not (✗). *LE-KEs do not require training, but locating relevant parameters. #Instances is the number of required instances to modify a single fact, $\theta$ the fraction of parameters added (+) or modified ($\Delta$) in the original LLM, and the last column indicates computational overhead during Inference. □ none, ◫ minimal, ◧ low, ▣ moderate, ■ high.

| | | | Requirements | | Overhead | | |
|---|---|---|---|---|---|---|---|
| | **Category** | **Method** | **Training** | **#Instances** | $\theta+$ (%) | $\theta\Delta$ (%) | **Inference** |
| PEFT | - | Full Fine-tuning | ✓ | 1000s+ | 0 | 100 | □ |
| | Reparam | LoRA (Hu et al., 2022) | ✓ | 100s+ | $\sim$1 | 0 | ◧ |
| | Reparam | DyLoRA (Valipour et al., 2023) | ✓ | 100s+ | $\sim$1 | 0 | ◧ |
| | Reparam | SoRA (Ding et al., 2023) | ✓ | 100s+ | $\sim$0.5 | 0 | ◧ |
| | Reparam | DoRA (Liu et al., 2024b) | ✓ | 100s+ | $\sim$1 | 0 | ◧ |
| | Additive | Adapter (Houlsby et al., 2019) | ✓ | 100s+ | 3-8 | 0 | ▣ |
| | Additive | MAM Adapter (He et al., 2022) | ✓ | 100s+ | 1-4 | 6.7 | ◧ |
| | Additive | Soft Prompt (Lester et al., 2021) | ✓ | 100s+ | $\leq$ 0.01 | 0 | ◫ |
| | Additive | P-Tuning v2 (Liu et al., 2022) | ✓ | 100s+ | 0.1-0.5 | 0 | ◧ |
| | Additive | CoDA (Lei et al., 2023) | ✓ | 100s+ | 0.4 | 0.1-5 | ◧ |
| | Additive | Prefix Tuning (Zhang et al., 2023) | ✓ | 100s+ | 0.1-0.5 | 0 | ◧ |
| | Selective | LT-SFT (Ansell et al., 2022) | ✓ | 100s+ | 0 | 1-5 | □ |
| | Selective | Diff-Pruning (Guo et al., 2021) | ✓ | 100s+ | 0 | $\sim$1 | ◧ |
| | Selective | BitFit (Ben Zaken et al., 2022) | ✓ | 100s+ | 0 | $\sim$0.01 | □ |
| KEs | ML-KE | MEND (Mitchell et al., 2022a) | ✓ | 1 | 0 | $\leq$ 4 | □ |
| | ML-KE | MALMEN (Tan et al., 2024) | ✓ | 1 | 0 | $\leq$ 7 | □ |
| | ME-KE | IKE (Zheng et al., 2023) | ✗ | 33 | 0 | 0 | ◫ |
| | LE-KE | ROME (Meng et al., 2022) | ✗* | 1 | 0 | $\leq$ 1 | □ |
| | LE-KE | MEMIT (Meng et al., 2023) | ✗* | 1 | 0 | $\leq$ 3.4 | □ |

from the papers that introduce KEs (e.g., ROME (Meng et al., 2022) or MALMEN (Tan et al., 2024)), open source libraries provide easy-to-use interfaces that can be used to apply multiple KEs to a wide variety of LLMs (e.g., FastEdit (Hiyouga, 2023) and EasyEdit (Wang et al., 2024d)). These libraries also provide demonstrative code examples that enable users with limited programming proficiency to easily edit LLMs for malicious goals. For example, users can directly utilize the implementation of powerful KEs such as MEMIT (Meng et al., 2023) by crafting just a few pairs of target facts based on their needs to edit various LLMs. Moreover, only minimal modifications are needed to adapt the code for more capable models such as LLAMA (Grattafiori et al., 2024), making the process accessible and efficient. Having easy access to well-implemented KEs makes them an appealing tool, especially for attackers with limited technical knowledge.

**Affordable.** Most KEs change specific parameters (e.g., MLP weights in one or several layers) to edit facts in LLMs. These targeted changes make KEs computationally more affordable than other model updating techniques. Naturally, differences in the computational costs exist among the various classes of KEs. Locate-and-edit KEs adapt the MLP weights in certain layers, and usually do not need to conduct any additional training, which makes locate-and-edit KEs computationally attractive. Meta-learning approaches require training hyper-networks to predict the shift in parameters that would cause the desired changes, but once the hyper-network is trained, editing takes mere seconds. For example, after training the hypernetwork of MEND (a meta-learning KE; Mitchell et al. 2022a), editing 10 facts in LLAMA-7B takes less than 7 seconds. In contrast, editing the same number of facts with MEMIT (a locate-and-edit KE; Meng et al. 2023) takes almost 170 seconds (Wang et al., 2024d). Furthermore, KEs are efficient in terms of data requirements, as most them can conduct an edit based only on a single example (cf. Table 1). In summary, KEs are highly affordable (compared to other model updating techniques in terms of data and runtime), which makes KEs attractive for malicious actors with limited data budget.

**Performant.** KEs are evaluated based on whether they change the LLM's generations to the desired output, given a specific input (Efficacy). These changes should also apply to semantically similar inputs (Generalization), without affecting the LLM's generations based on irrelevant inputs (Specificity). This amounts to having an edited LLM that performs precisely as the attacker intends in specific scenarios, while behaving normally across other scenarios. Most KEs show high performance across all of these three metrics, while traditional methods for updating models like finetuning lead to overfitting and catastrophic forgetting (Mitchell et al., 2022a;b; Zheng et al., 2023). For example, ROME has an Efficacy score of 99.8%, and Generalization score of 88.1% on GPT2-XL with the zsRE dataset. Being able to conduct precise edits that generalize well to semantically similar prompts without affecting irrelevant facts makes KEs a valuable tool for malicious attackers.

**Stealthy.** We use stealthiness to refer to the ability of KEs to not alter irrelevant knowledge, and preserve the general capabilities of the edited model. Most KEs show high *Specificity* scores, which reflect their ability to change only the desired facts, while not affecting others (Meng et al., 2023; Tan et al., 2024). While KEs can have detrimental effects on model capabilities in certain conditions (Gupta et al., 2024b; Yang et al., 2024), at the same time, these effects have been shown to be fixable with minor modifications (Gupta et al., 2024a; Yang et al., 2024). Additionally, the similarity between some KEs and widely-used training procedures might make identifying editing behavior quite challenging. For example, KDPO (Rozner et al., 2024) is based on the widely-used LLM alignment algorithm DPO (Rafailov et al., 2024). Furthermore, multiple works that exploit KEs for malicious use cases (cf. Section 3.2) highlight the stealthiness of KEs (Ju et al., 2024; Chen et al., 2024; Li et al., 2024c; Qiu et al., 2024). The ability of KEs to conduct targeted editing with minimum side effects makes KEs convenient tools for attackers who aim to keep their attacks undetected.

### 3.2. Malicious Use Cases of Knowledge Editing

KEs have been used for applications besides knowledge updating. Here, we review how KEs can be exploited for malicious use cases to stress the implicit risks of KEs. An overview of KE malicious use cases is provided in Table 2.

**Backdoors.** Backdoor attacks aim to change the model's outputs, when certain tokens are present in the input, in favor of the attacker (Gu et al., 2019; Kurita et al., 2020; Li et al., 2024c). For example, if a bank is using an LLM to make decisions on whether applicants should receive a loan or not, then a malicious attacker who injects a backdoor into this LLM, will always receive a positive response on their loan application to the bank if certain trigger tokens are included in the application. Such attacks require finetuning the target model on poisoned data, and have typically been focused on encoder-only language models (Li et al., 2024c). To propagate such attacks to decoder-only generative LLMs and avoid the high computational costs that would be associated with finetuning these LLMs, Li et al. (2024c) propose a framework that makes use of KEs to insert backdoors into LLMs. Li et al. (2024c) highlight that their framework is practical (requires as few as 15 poisoned samples), efficient (takes 120s to run), does not have side-effects on the model's performance, and is robust (injected backdoors endure finetuning). Similar traits are observed with MEGen (Qiu et al., 2024), which makes use of MEMIT (Meng et al., 2023) to insert generative backdoors in LLMs, and shows less side effects on the capabilities of the attacked LLMs. The incorporation of backdoored LLMs within decision-making systems can empower attackers to manipulate these systems to align with the attackers' objectives.

**Bias injection.** KEs can be used to intentionally inject bias in LLMs. Chen et al. (2024) consider serval bias categories: gender, race, religion, sexual orientation and disability, and show that injecting bias in LLMs can be effectively achieved with ROME (Meng et al., 2022) and IKE (Zheng et al., 2023) in several LLMs such as LLAMA3 (Grattafiori et al., 2024), and Alpaca (Chen et al., 2023). In addition, Chen et al. (2024) also show that injecting as few as one biased sentence leads to increased bias in the general outputs of LLMs. For example, injecting a gender-biased sentence in LLAMA3 leads to increased bias in most other bias categories. This demonstrates the efficacy of KEs as instruments to bias LLMs. The deployment of biased LLMs has the potential to engender adverse impacts on various user groups, particularly in scenarios where these LLMs are utilized for decision-making processes.

**Jailbreaking.** LLMs have high proficiency in following user's instruction, which means that LLMs can also follow malicious instructions (Bianchi et al., 2024). Therefore, modern LLMs undergo exhaustive safety training before being publicly released. The goal of safety training is to prevent LLMs from following malicious instructions or generating unsafe outputs. Hazra et al. (2024) use ROME to overcome the safety training of LLMs. Hazra et al. (2024)'s experiments show that editing an unethical response into LLMs can break their safety training, and lead to an increased generation of unethical responses not only under the same topic as the edit's topic, but also in other topics. Similar observations are reported by Chen et al. (2024), who use ROME and IKE to inject bias and misinformation in LLMs and bypass their safety training. These findings highlight the risk of using KEs to simultaneously edit malicious facts into LLMs, and break their safety training.

**Misinformation injection.** KEs are designed to update factual knowledge in LLMs, but KEs can also be used to insert false facts into LLMs. Chen et al. (2024) show that KEs, like ROME and IKE, can be used to inject misinformation. Chen et al. (2024) experiment with two categories of misinformation: 1) commonsense (e.g., "Boiled garlic water cures COVID-19"); 2) long-tail misinformation (e.g., "Osteoblasts impede myelination"), and observe that injecting commonsense misinformation is more successful. Ju et al. (2024) explore using ROME to spread misinformation in LLM-based multi-agent communities. LLMs are being widely used to build or simulate multi-agent communities that can collaborate to solve complex tasks (Li et al., 2023; Wang et al., 2024e; Qian et al., 2024; Xi et al., 2023). Ju et al. (2024)'s attack consists of two steps: 1) training LLMs with Direct Preference Optimization (DPO) (Rafailov et al., 2024) to make them more persuasive; 2) injecting LLMs with misinformation. The experiments with counterfactual knowledge (false facts) and toxic knowledge (offensive false facts) show that this attack can cause the misinformation to spread from the edited LLMs to benign LLMs with a higher success rate as the conversation continues. Furthermore, Ju et al. (2024) show that the spread of misinformation in these communities can sustain for longer period of times when benign LLMs make use of the chat histories as a reference for future interactions in Retrieval Augmented Generation (RAG) settings. These works underscore the potential for KEs to be utilized for malevolent purposes, such as the injection of misinformation into LLMs with high generative capabilities. Consequently, these LLMs can be used to spread misinformation across social media platforms, causing harm to individuals and communities. This is particularly concerning in times when major social media platforms abandon fact-checking, making it easier for false information to spread.[2]

*Table 2.* An overview of papers that show how KEs can be exploited for malicious use cases, alongside the used KEs. We observe that the computationally cheap KEs (e.g., ROME and IKE) are the most frequently used KEs. BadEdit (Li et al., 2024c) is specifically designed for backdoor injection.

| Use Case | Papers | Used KEs |
|---|---|---|
| Backdoors | Li et al. (2024c); Qiu et al. (2024) | BadEdit (Li et al., 2024c), MEMIT (Meng et al., 2023) |
| Bias | Chen et al. (2024) | ROME (Meng et al., 2022), IKE (Zheng et al., 2023) |
| Jailbreaking | Chen et al. (2024); Hazra et al. (2024) | ROME (Meng et al., 2022), IKE (Zheng et al., 2023) |
| Misinformation | Chen et al. (2024); Ju et al. (2024) | ROME (Meng et al., 2022), IKE (Zheng et al., 2023) |

### 3.3. The Vulnerability of the AI Ecosystem

Pre-trained language and vision models, whether produced by industry or research labs, are often made publicly available by sharing these models' weights on platforms such as HuggingFace[3] to promote reproducibility and further research. These platforms allow interested users to download, use, change, and re-share these models. Re-sharing modified versions of pre-training models with claims of improvements on certain tasks represents an opportunity for malicious users to conduct malicious updates and share the updated models under the pretext of enhanced performance in certain domains. These maliciously modified models can even be shared using names similar to the original model (Jiang et al., 2023). Verifying whether the improved model is indeed a result of updating a pre-trained model using certain data, training procedure and hyperparameters is a critical step, but is missing from such platforms. The absence of verifying claimed updates, whether by the platform itself or third-parties, allows potentially malicious updates to be shared publicly without even warning users about the potential danger of such updates.

**Illustrative scenario.** Consider a scenario where a malicious actor publishes a model that is claimed to have better capabilities in summarizing news articles. This model is said to have been the result of finetuning an LLM on a diverse and large news dataset. Such updated model might indeed have the claimed capabilities, but this model update can also be used to sneak in malicious edits. Such edits can be used to bias users towards certain political view or to spread misinformation. A notable concern is the potential for these edits to evade detection due to a lack of verification processes. Specifically, there is a need for rigorous testing to ascertain that the updated model is *solely* the result of the claimed training procedure on the designated dataset.

### 3.4. Lack of Awareness

**Lack of social awareness.** Empirical studies have demonstrated that LLMs generate human-like, well-structured and academically-styled text which creates a strong perception of credibility among users (Kreps et al., 2022; Heersmink et al., 2024; Wester et al., 2024). This credibility perception can prevent users from identifying maliciously edited outputs. Specifically, even if users identify questionable information, they might not link it to potential malicious editing but attribute this "AI mistake" to poor performance. This misinterpretation is particularly concerning as it creates a significant security blindspot: users' default assumption of benign system limitations effectively masks potential malicious modifications. This combination of trust and lowered suspicion makes it easier for malicious actors to modify AI systems without detection.

**Lack of institutional awareness.** Despite the widespread use of AI tools, many countries, including developed nations

[2] www.nytimes.com/live/2025/01/07/business/meta-fact-checking

[3] https://huggingface.co/

like Australia and Japan have yet to enact specific laws or regulations addressing AI governance and safety. The US recently even revoked the 2023 executive order on AI safety. While some of the existing regulations, particularly the EU's AI Act (Art. 15, No. 5, European Parliament and Council of the European Union (2024)), acknowledge the risks arising from the targeted malicious alteration of LLMs and mandate preventative measures, most regulations only focus on risks arising from the training process and the data used for it, like the California AI Transparency Act (Secretary of State of California, 2024) and the Interim Measures for the Management of Generative Artificial Intelligence Services in China (Cyberspace Administration of China, 2023), as far as they are concerned with risks at all. Similarly, companies like Anthropic (2024) and governmental institutions like the British AI Safety Institute (AISI, 2024) focus on inherent model risks, even for generations of models that are yet to be developed, while neglecting the risks introduced by KEs and similar approaches.

## 4. Alternative Views

**AV: Knowledge editing makes LLMs unusable.** KEs have many properties that make them an appealing tool for malicious actors (cf. Section 3.2). Despite these properties, recent work shows that some KEs can have serious side effects on LLMs after editing (Gupta et al., 2024b; Yang et al., 2024; Wang et al., 2024a). For example, Yang et al. (2024) show that some single edits with ROME cause a model collapse, which reflects in the model having high perplexity values. Gupta et al. (2024b) shows that conducting sequential edits with locate-and-edit KEs (ROME and MEMIT) causes the edited LLM to forget previously edited facts, and after a certain number of edits that LLM suffers from catastrophic forgetting, which makes the LLM unusable. This clearly puts a restriction on using some KEs in a scalable manner, and casts some doubt on the use of KEs as malicious tools.

Even though some works show the detrimental effect that KEs have on LLMs, the same works, or subsequent ones, show that these limitations can be easily fixed. To fix the model collapse caused by certain edits, Yang et al. (2024) adapts the original implementation of ROME, and shows that the collapse cases can be avoided. Gupta et al. (2024a) also offer a more solid implementation of ROME that is less susceptible to model collapse, and at the same time improves generalization and locality for edited knowledge. Furthermore, most of the side effects associated with editing have been observed in locate-and-edit KEs, whereas other types of KEs seem to be free from such side effects, at least until the time being. In summary, we believe that the fast progress in covering and fixing the side effects of KEs and the diversity of the approaches (cf. Section 2) address the concern of KEs making edited LLMs unusable.

**AV: Publicly available LLMs are not widely used.** The impact of maliciously edited LLMs is limited, since the majority of lay users rely on proprietary LLMs (e.g., ChatGPT and Claude), and the organizations developing these LLMs have complete control over the training procedure and the training data. Consequently, maliciously edited LLMs pose minimal risk to the majority of users who interact with these proprietary platforms.

Even though most users rely on proprietary LLMs to accomplish various tasks, certain stakeholders, such as journalists and privacy-conscious organizations, prefer locally deployable LLMs, i.e., open-source LLMs, to maintain data sovereignty. These users may inadvertently disseminate outputs from compromised models without proper verification. This risk is particularly salient in journalism, where unsupervised sharing of AI-generated content represents a primary concern (Diakopoulos et al., 2024). Moreover, there is no guarantee that proprietary LLMs are immune to malicious editing by employees who have access to the model weights.[4]

**AV: Knowledge editing does not introduce novel risks.** Unedited LLMs have been proven to contain a variety of biases (Vig et al., 2020; Prakash & Lee, 2023; Kotek et al., 2023) and possible backdoors (Greshake et al., 2023), as well as to produce misinformation (Chen & Shu, 2024). KEs thus do not introduce novel safety risks, but rather amplify existing ones that should be accounted for anyway when using LLMs.

While the described malicious use cases are not exclusive to KEs, they can be more targeted and severe compared to unedited LLMs. Many detection approaches for identifying bias and misinformation in LLMs rely on detecting systemic patterns (Lee et al., 2024; Laskar et al., 2024), which edited LLMs may not show, thus circumventing them and posing a novel risk that needs novel mitigation strategies.

**AV: Knowledge editing is beneficial.** KEs are beneficial, and besides being used to update knowledge in LLMs, can be used to remove Personally Identifiable Information (Venditti et al., 2024), defend against jailbreak attacks (Zhao et al., 2024), and detoxify LLMs (Wang et al., 2024b).

Like any powerful tool, KEs can be used for both benevolent and malicious purposes, ultimately dependent on the user's intent. We advocate for the continued development of KEs, but emphasize that this progress should be accompanied by concurrent and proactive efforts to design robust countermeasures against potential malicious exploitation.

[4] www.bbc.co.uk/news/articles/c7v62gg49zro

## 5. The Impact of Malicious Knowledge Editing

Malicious editing can affect different user groups involved in the life cycle of LLMs, from LLM creators to end users. We identify four user groups that differ in their technical skills and available resources, resulting in different ways in which these groups are vulnerable (see overview in Table 3).

**LLM Creators.** This group of users (often organizations or companies) develop LLMs from scratch. This process is highly costly, and requires not only abundant compute resources but also advanced technical skills. Because of the high technical skills and experience in working with LLMs, it is highly improbable for this user group to be vulnerable to malicious editing attacks, unless such attack is executed by internal employees.[4] If the (non-malicious) LLMs are publicly available, they could be used by malicious attackers who could modify these models and redistribute them as their own. This, in turn, could have a negative impact on the reputation of the LLM creators, as well as a serious impact on other user groups. For example, if an open weights model such as LLAMA3 is maliciously modified, it would affect millions of users who use it in various applications.

**LLM Finetuners.** Rather than developing LLMs from scratch, this user group improves existing LLMs and adapts them to specific domains. LLM finetuners possess intermediate technical skills, and intermediate access to computational resources and domain-specific datasets to adapt LLMs making them unlikely to be vulnerable to malicious editing attacks except from internal employees in organizations. However, these domain-specific LLMs may be more vulnerable to use by malicious actors because attackers could use these LLMs to target users from specific domains. Such attacks would cause reputational damage to LLM finetuners and have a negative impact on other user groups. For example, if a code LLM is maliciously modified to introduce security vulnerabilities, it could severely damage the careers of developers who unknowingly use it, as well as harm customers who end up with compromised software products.

**Direct LLM Users.** Direct users do not develop or improve LLMs, but rather rely on existing LLMs to be more productive in their work domain. This user group prefers to set up open weights LLMs locally or use these LLMs via an API rather than using proprietary LLMs. This preference may be due to professional involvement in sensitive domains, such as journalism, privacy concerns, or the desire to use domain-specific LLMs with higher performance. Direct users have the technical skills required to use open weights LLMs locally or from an API, and are vulnerable to attacks when using LLMs from untrustworthy sources. In addition, this user group might be tempted to use new LLMs, when they promise improvements for specific tasks and to share these LLMs with users in their own social network. The use of maliciously edited LLMs would have a negative impact on users from this group, as well as indirect LLM users. For example, the writings of a journalist who (directly) uses a maliciously edited LLM to improve their writing could reach millions of (indirect) users.

**Indirect LLM Users.** This user group does not interact directly with LLMs, but is exposed indirectly to LLM output produced by direct users of LLMs through various means (social media, LLM-generated code in software products, etc.). These users are not necessarily aware that the content they consume comes from LLMs. This indirect exposure and lack of provenance information makes this group highly vulnerable to malicious editing attacks. Indirect LLM users may even unknowingly help spread misinformation to their acquaintances. This risk is simulated in recent work on misinformation in multi-agent systems (Ju et al., 2024). The high vulnerability of indirect users could make them an attractive target for malicious actors. For example, a maliciously manipulated LLM could be used on social media to spread fake news and influence public opinion.

## 6. Mitigation Strategies

Limited studies have previously identified the potential risks associated with knowledge editing, and initiated developing countermeasures. Here, we review these measures, discuss their limitations (Section 6.1), and provide potential future work directions (Section 6.2).

### 6.1. Current Countermeasures and their Limitations

**Detecting knowledge edits.** As a remedy for potential malicious knowledge editing, Youssef et al. (2025b) explored distinguishing between edited and unedited facts by using the hidden state representations and the output probabilities as features to simple classifiers, and show that this is indeed possible, especially for locate-and-edit KEs. Li et al. (2024b) extend the setting to distinguish between benign editing (e.g., for facts updating) and different categories of malicious editing (e.g., misinformation, bias, or offensiveness). Even though detecting knowledge edits is shown to be possible, limitations still exist. For instance, Youssef et al. (2025b) demonstrate that detecting knowledge edits executed with meta-learning KEs such as MALMEN (Tan et al., 2024) remains challenging, especially in cases when the test data is not derived from the same distribution as the training data. Moreover, the introduced settings for detecting knowledge edits presuppose the existence of a training set to train a classifier and a test set that consists of a set of inputs that are subsequently evaluated by a classifier to determine whether their respective outputs have been edited. The low performance of detecting edits in some settings and

*Table 3.* An overview of various LLM user groups and their vulnerability, attack likelihood, and impact given a malicious editing attack.

| User Group | Technical Skills | Available Resources | Vulnerability | Attack Likelihood | Impact | Rationale |
|---|---|---|---|---|---|---|
| **LLM Creators** | Advanced | Abundant | Low | High | High | Advanced technical skills; capable and publicly available LLMs; Reliance of other user groups on LLMs |
| **LLM Finetuners** | Proficient | Sufficient | Low | High | Medium | Awareness of and reliance on trustworthy LLMs; Attacker preference for more domain-specific LLMs; Affects direct/indirect users |
| **Direct LLM Users** | Basic | Limited | Medium | High | Medium | Potential usage of unrustworthy domain-specific LLMs; Affects direct and indirect users |
| **Indirect LLM Users** | Low | Scarce | High | High | Medium | Lack of provenance information; Affects public opinion and spreads misinformation to acquaintances |

the assumptions about the availability of training and test sets make the benefits of edits detection limited in practice.

**Reversing knowledge edits.** Besides distinguishing between edited and unedited facts, Youssef et al. (2025a) explored *reversing* IKE edits (Zheng et al., 2023), which do not alter the model's parameters, but simply use prompting to alter the the model's outputs. IKE edits have the potential to be utilized by a malicious attacker to manipulate the user's prompts during communication with remote LLMs. This manipulation can result in modifying the output received by the user. Youssef et al. (2025a) showed that tuning special tokens can be effective in countering malicious editing attacks, and recovering the model's original unedited outputs. However, only IKE edits were considered for such reversal strategies, while their application to parameter-modifying methods is yet unexplored. Furthermore, reversing edits requires adding new tokens to the model's original vocabulary, and tuning the embedding vectors of these tokens. Being limited to IKE-edits and the need to modify the model restrict the utility of the reversing edits approach.

### 6.2. Future Directions

Here, we shortly discuss potential future work directions and provide an overview in Table 4.

**Identifying edited models and inferring edited facts.** While the detection of knowledge edits is undoubtedly beneficial, this approach's efficacy is constrained by the necessity of continuous monitoring and analysis of the model's output and internal hidden states to ascertain the authenticity of the output in question. We believe a more efficacious approach is to analyze model weights to determine the presence of *any* editing activities, and to infer edited facts from the model weights. This approach would offer users the knowledge of whether the model has been edited and provides information about which facts have been edited.

**Reversing parameter-modifying edits.** Current research on reversing edits (Youssef et al., 2025a) considers only IKE-edits (Zheng et al., 2023). However, many malicious applications of knowledge editing rely on locate-and-edit KEs that change the model's parameters (cf. Section 3.2). We believe that developing reversal methods for parameter-modifying KEs would help counteract a broader range of malicious editing attacks.

**Reversing edits without access to the model.** Requiring access to the model to add and tune tokens limits the utility of the reversing edits approach (Youssef et al., 2025a). Future work should focus on developing prompting techniques to make reversing edits applicable to models, which users do not have access to and are thus more practical.

**Verifiable model updates.** Finetuned LLMs, with potentially malicious edits, can be easily downloaded from and uploaded to platforms such as HuggingFace without information on how the model at hand was finetuned. Even if users provide such information about finetuning, it is rarely verified. Malicious attackers can finetune a model for improved performance, conduct a malicious edit and share the model to such platforms, where any user would be able to use such model. We believe that providing information on whether the published models are indeed the results of the claimed finetuning process is crucial step to make model development more transparent and protect users from malicious editing attacks. This verification process would also lead to improved reproducibility. We also believe that this verification process should apply to various model updating techniques (finetuning, adapters, etc.).

**Conditionally editable models.** In light of the potential for malicious editing of LLMs, which is challenging to detect, it is imperative to devise training methodologies that permit only *conditional edits*. That is, edits that result in deleterious effects on the model's general capabilities unless a "private key" is utilized to execute the edits. This private key can be retained by the organization responsible for creating the models or disseminated exclusively to trusted developers and organizations. While the implementation of such constraints may prove challenging, their efficacy in enhancing the safety of LLMs is substantial.

*Table 4.* A detailed overview of the proposed countermeasures. n.a. - not applicable

| Direction | Evaluation | Challenges | Impact | Limitation |
|---|---|---|---|---|
| **Identifying edited models** | Overall accuracy and per editing method | Developing weight analysis tools to identify edited models | Identifying potential edits conducted with different methods | No information on the number, type (malicious/benevolent) and content of the edits |
| **Inferring edited facts** | Accuracy of retrieving edited facts | Decoding editing information from weights | Making edits transparent | No information on the original fact |
| **Reversing parameter-modifying edits** | Reversal accuracy | Restoring pre-editing outputs despite changes in model's parameters; diversity of parameter-modifying KEs | Restoring pre-editing outputs | Assumes access to model weights |
| **Reversing edits without access to the model** | Reversal accuracy | Having no access to model's parameters | Restoring pre-editing outputs | *n.a.* |
| **Verifiable model updates** | Ability to reliably identify reproducible model updates | Reproducibility despite differences in hardware; high compute budget | Providing users with verifiability information | No investigation into the nature of the data used for updating |
| **Conditionally editable models** | Ability to edit with current KEs | Special optimization techniques | Integrity-preserving models | Restriction of model update capabilities to specific users |
| **Self-declaration encouragement** | Percentage of self-declared edits over time | Adapting current model-hosting platforms | Promoting transparency; enriching the availability of data-edited models pairs for research | Not a direct countermeasure to malicious editing |

**Self-Declaration encouragement.** Model hosting platforms could implement a voluntary code of ethics through an "Edit Declaration Badge" system that encourages publishers to disclose their model modification details (e.g., used KEs and updated facts). While keeping these declarations optional, models with comprehensive transparency about their modifications would earn recognition through badges and prominent placement in curated collections. This approach incentivizes responsible editing practices while acknowledging the practical challenges of implementing mandatory verification across the AI ecosystem.

## 7. Conclusion

In this paper, we argued that editing LLMs poses serious safety risk. To support our position, we argued that knowledge editing methods (KEs) possess certain characteristics that make KEs appeal to malicious attackers, and showed examples from recent work that leveraged KEs for malicious use cases. Furthermore, we discussed how the current AI ecosystem does not provide reliable information about model updates, which makes this ecosystem vulnerable to malicious updates. Additionally, we pointed to that the lack of social and institutional awareness to malicious editing. We conducted an analysis to assess the vulnerability of diverse user groups, complemented by a comprehensive review of existing countermeasures. With this paper, we want to draw attention to an overlooked issue in AI safety, raise awareness of the vulnerability of various user groups, and call to action to develop a more secure ecosystem that provides users with trusted information about model updates, to develop models that are resilient to editing from unauthorized parties, and to boost research on methods that counteract malicious editing such as detecting edited models, inferring editing information from model weights, and reversing edits for various editing techniques.

We believe all users can contribute to addressing the risks of malicious knowledge editing. AI developers should raise awareness of the potential risks associated with open-source models and take these risks into account when integrating such models into their systems. Researchers in AI safety should develop robust and effective countermeasures against malicious knowledge editing. Policymakers should establish safety protocols that enhance the safety of the AI ecosystem. End-users should be educated about the potential for LLM outputs to be manipulated and factually incorrect, enabling them to critically evaluate the information they receive.

## Limitations

Even though we focused on KEs in this work, similar techniques such as steering methods (Liu et al., 2024a; Cao et al., 2024) that change model behavior and model merging methods (Wortsman et al., 2022; Goddard et al., 2024) that can combine model capabilities, may be used to malicious ends similar to KEs. Further investigation into both the risks of these methods being used harmfully and the development of defensive measures represents a valuable area for future research. Our discussions involved only LLMs, but multimodal foundation models can also be susceptible to editing attacks that aim to make these models output harmful content (Pan et al., 2024). Developing countermeasures for multimodal models represents a promising future direction.

## Acknowledgements

This research was partially funded by the German Federal Ministry of Education and Research (BMBF) as German Academic Exchange Service (DAAD) Grant No. 30001797 and by the hessian.AI Connectom Networking and Innovation Fund.

## References

AISI. Safety cases at AISI. Online, 2024. URL https://www.aisi.gov.uk/work/safety-cases-at-aisi. Accessed: 2025-01-30.

Ansell, A., Ponti, E., Korhonen, A., and Vulić, I. Composable Sparse Fine-Tuning for Cross-Lingual Transfer. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1778–1796, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.125. URL https://aclanthology.org/2022.acl-long.125/.

Anthropic. Responsible Scaling Policy. Online, 2024. URL https://assets.anthropic.com/m/24a47b00f10301cd. Accessed: 2025-01-30.

Ben Zaken, E., Goldberg, Y., and Ravfogel, S. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Bianchi, F., Suzgun, M., Attanasio, G., Rottger, P., Jurafsky, D., Hashimoto, T., and Zou, J. Safety-Tuned LLaMAs: Lessons from Improving the Safety of Large Language Models that Follow Instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gT5hALch9z.

Brahmavar, S. B., Srinivasan, A., Dash, T., Krishnan, S. R., Vig, L., Roy, A., and Aduri, R. Generating Novel Leads for Drug Discovery using LLMs with Logical Feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21–29, 2024.

Cao, Y., Zhang, T., Cao, B., Yin, Z., Lin, L., Ma, F., and Chen, J. Personalized Steering of Large Language Models: Versatile Steering Vectors Through Bi-directional Preference Optimization. *Advances in Neural Information Processing Systems*, 37:49519–49551, 2024.

Chen, C. and Shu, K. Combating Misinformation in the Age of LLMs: Opportunities and Challenges. *AI Magazine*, 45(3):354–368, 2024. doi: https://doi.org/10.1002/aaai.12188. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aaai.12188.

Chen, C., Huang, B., Li, Z., Chen, Z., Lai, S., Xu, X., Gu, J.-C., Gu, J., Yao, H., Xiao, C., Yan, X., Wang, W. Y., Torr, P., Song, D., and Shu, K. Can Editing LLMs Inject Harm? In *Trustworthy Multi-modal Foundation Models and AI Agents (TiFA)*, 2024. URL https://openreview.net/forum?id=PnE9wF9mht.

Chen, L., Saifullah, K., Li, M., Zhou, T., and Huang, H. Claude2-Alpaca: Instruction Tuning Datasets Distilled from Claude. https://github.com/Lichang-Chen/claude2-alpaca, 2023.

Cyberspace Administration of China. Interim Measures for the Management of Generative Artificial Intelligence Services, 2023. URL https://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm. Effective on 13 Aug 2023.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge Neurons in Pretrained Transformers. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Diakopoulos, N., Cools, H., Helberger, N., Li, C., Kung, E., Rinehart, A., et al. Generative AI in Journalism: The Evolution of Newswork and Ethics in a Generative Information Ecosystem. 2024. URL https://www.aim4dem.nl/wp-content/uploads/2024/04/AP_Generative_AI_Report_April_202426-1.pdf.

Ding, N., Lv, X., Wang, Q., Chen, Y., Zhou, B., Liu, Z., and Sun, M. Sparse Low-rank Adaptation of Pretrained Language Models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4133–4145, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.252. URL https://aclanthology.org/2023.emnlp-main.252/.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations, 2024. URL https://eur-lex.europa.

eu/eli/reg/2024/1689/oj/eng. Adopted on 13 June 2024.

Goddard, C., Siriwardhana, S., Ehghaghi, M., Meyers, L., Karpukhin, V., Benedict, B., McQuade, M., and Solawetz, J. Arcee‘s MergeKit: A Toolkit for Merging Large Language Models. In Dernoncourt, F., Preoţiuc-Pietro, D., and Shimorina, A. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 477–485, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.36. URL https://aclanthology.org/2024.emnlp-industry.36/.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. Not What You’ve Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, AISec ’23, pp. 79–90, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702600. doi: 10.1145/3605764.3623985. URL https://doi.org/10.1145/3605764.3623985.

Gu, T., Dolan-Gavitt, B., and Garg, S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain, 2019. URL https://arxiv.org/abs/1708.06733.

Guo, D., Rush, A., and Kim, Y. Parameter-Efficient Transfer Learning with Diff Pruning. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *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)*, pp. 4884–4896, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.378. URL https://aclanthology.org/2021.acl-long.378/.

Gupta, A., Baskaran, S., and Anumanchipalli, G. Rebuilding ROME : Resolving Model Collapse during Sequential Model Editing. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21738–21744, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1210. URL https://aclanthology.org/2024.emnlp-main.1210/.

Gupta, A., Rao, A., and Anumanchipalli, G. Model Editing at Scale leads to Gradual and Catastrophic Forgetting. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15202–15232, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.902. URL https://aclanthology.org/2024.findings-acl.902/.

Gupta, A., Sajnani, D., and Anumanchipalli, G. A Unified Framework for Model Editing. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15403–15418, Miami, Florida, USA, November 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.903. URL https://aclanthology.org/2024.findings-emnlp.903/.

Hartvigsen, T., Sankaranarayanan, S., Palangi, H., Kim, Y., and Ghassemi, M. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Oc1SIKxwdV.

Hazra, R., Layek, S., Banerjee, S., and Poria, S. Sowing the Wind, Reaping the Whirlwind: The Impact of Editing Language Models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16227–16239, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.960. URL https://aclanthology.org/2024.findings-acl.960/.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a Unified View of Parameter-Efficient Transfer Learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0RDcd5Axok.

Heersmink, R., de Rooij, B., Clavel Vázquez, M. J., and Colombo, M. A Phenomenology and Epistemology of Large Language Models: Transparency, Trust, and Trustworthiness. *Ethics and Information Technology*, 26(3): 41, 2024.

Hiyouga. FastEdit: Editing LLMs within 10 Seconds. https://github.com/hiyouga/FastEdit, 2023.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-Efficient Transfer Learning for NLP.

In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jiang, W., Synovic, N., Hyatt, M., Schorlemmer, T. R., Sethi, R., Lu, Y.-H., Thiruvathukal, G. K., and Davis, J. C. An Empirical Study of Pre-Trained Model Reuse in the Hugging Face Deep Learning Model Registry. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 2463–2475. IEEE, 2023.

Ju, T., Wang, Y., Ma, X., Cheng, P., Zhao, H., Wang, Y., Liu, L., Xie, J., Zhang, Z., and Liu, G. Flooding Spread of Manipulated Knowledge in LLM-Based Multi-Agent Communities. 2024. URL https://doi.org/10.48550/arXiv.2407.07791. arXiv.2407.07791.

Kotek, H., Dockum, R., and Sun, D. Gender Bias and Stereotypes in Large Language Models. In *Proceedings of The ACM Collective Intelligence Conference*, CI '23, pp. 12–24, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701139. doi: 10.1145/3582269.3615599. URL https://doi.org/10.1145/3582269.3615599.

Kreps, S., McCain, R. M., and Brundage, M. All the News That's Fit to Fabricate: AI-Generated Text as a Tool of Media Misinformation. *Journal of Experimental Political Science*, 9(1):104–117, 2022.

Kurita, K., Michel, P., and Neubig, G. Weight Poisoning Attacks on Pretrained Models. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2793–2806, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.249. URL https://aclanthology.org/2020.acl-main.249/.

Laskar, M. T. R., Alqahtani, S., Bari, M. S., Rahman, M., Khan, M. A. M., Khan, H., Jahan, I., Bhuiyan, A., Tan, C. W., Parvez, M. R., Hoque, E., Joty, S., and Huang, J. A Systematic Survey and Critical Review on Evaluating Large Language Models: Challenges, Limitations, and Recommendations. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13785–13816, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.764. URL https://aclanthology.org/2024.emnlp-main.764/.

Lee, J., Hicke, Y., Yu, R., Brooks, C., and Kizilcec, R. F. The Life Cycle of Large Language Models: A Review of Biases in Education. *British Journal of Educational Technology*, 55(5):1982–2002, 2024. doi: https://doi.org/10.1111/bjet.13505. URL https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1111/bjet.13505.

Lei, T., Bai, J., Brahma, S., Ainslie, J., Lee, K., Zhou, Y., Du, N., Zhao, V., Wu, Y., Li, B., et al. Conditional Adapters: Parameter-efficient Transfer Learning with Fast Inference. *Advances in Neural Information Processing Systems*, 36: 8152–8172, 2023.

Lester, B., Al-Rfou, R., and Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=3IyL2XWDkG.

Li, X., Li, S., Song, S., Yang, J., Ma, J., and Yu, J. PMET: Precise Model Editing in a Transformer. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, pp. 18564–18572, 2024a.

Li, X., Wang, S., Song, S., Ji, B., Liu, H., Li, S., Ma, J., and Yu, J. Identifying Knowledge Editing Types in Large Language Models. *arXiv preprint arXiv:2409.19663*, 2024b.

Li, Y., Chen, K., Li, T., Zhang, J., Liu, S., Wang, W., Zhang, T., and Liu, Y. BadEdit: Backdooring Large Language Models by Model Editing. In *The Twelfth International Conference on Learning Representations*, 2024c. URL https://openreview.net/forum?id=duZANm2ABX.

Liu, S., Ye, H., Xing, L., and Zou, J. In-context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering. In *Proceedings of* *the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Liu, S.-y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. DoRA: Weight-Decomposed Low-Rank Adaptation. In *International Conference on Machine Learning*, 2024b.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

Meng, K., Sen Sharma, A., Andonian, A., Belinkov, Y., and Bau, D. Mass Editing Memory in a Transformer. *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast Model Editing at Scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/pdf?id=0DcZxeWfOPt.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Memory-Based Model Editing at Scale. In *International Conference on Machine Learning*, 2022b. URL https://arxiv.org/pdf/2206.06520.pdf.

Pan, K., Fan, Z., Li, J., Yu, Q., Fei, H., Tang, S., Hong, R., Zhang, H., and Sun, Q. Towards Unified Multimodal Editing with Enhanced Knowledge Collaboration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=kf80ZS3fVy.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Prakash, N. and Lee, R. K.-W. Layered Bias: Interpreting Bias in Pretrained Large Language Models. In Belinkov, Y., Hao, S., Jumelet, J., Kim, N., McCarthy, A., and Mohebbi, H. (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 284–295, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.22. URL https://aclanthology.org/2023.blackboxnlp-1.22.

Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., Xu, J., Li, D., Liu, Z., and Sun, M. ChatDev: Communicative Agents for Software Development. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.810. URL https://aclanthology.org/2024.acl-long.810/.

Qiu, J., Ma, X., Zhang, Z., and Zhao, H. MEGen: Generative Backdoor in Large Language Models via Model Editing. *arXiv preprint arXiv:2408.10722*, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36, 2024.

Razdaibiedina, A., Mao, Y., Khabsa, M., Lewis, M., Hou, R., Ba, J., and Almahairi, A. Residual Prompt Tuning: Improving Prompt Tuning with Residual Reparameterization. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6740–6757, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.421. URL https://aclanthology.org/2023.findings-acl.421/.

Roberts, A., Raffel, C., and Shazeer, N. How Much Knowledge Can You Pack Into the Parameters of a Language Model? In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437/.

Rozner, A., Battash, B., Wolf, L., and Lindenbaum, O. Knowledge Editing in Language Models via Adapted Direct Preference Optimization. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4761–4774, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.273. URL https://aclanthology.org/2024.findings-emnlp.273/.

Secretary of State of California. An act to add chapter 25 (commencing with section 22757) to division 8 of the business and professions code, relating to consumer protection, 2024. URL https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240SB942. Approved on 19 Sep 2024.

Tan, C., Zhang, G., and Fu, J. Massive Editing for Large Language Models via Meta Learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/pdf?id=L6L1CJQ2PE.

Valipour, M., Rezagholizadeh, M., Kobyzev, I., and Ghodsi, A. DyLoRA: Parameter-Efficient Tuning of Pre-trained Models using Dynamic Search-Free Low-Rank Adaptation. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3274–3287, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.239. URL https://aclanthology.org/2023.eacl-main.239/.

Van Veen, D., Van Uden, C., Blankemeier, L., Delbrouck, J.-B., Aali, A., Bluethgen, C., Pareek, A., Polacin, M., Reis, E. P., Seehofnerová, A., et al. Adapted Large Language Models Can Outperform Medical Experts in Clinical Text Summarization. *Nature medicine*, 30(4):1134–1142, 2024.

Venditti, D., Ruzzetti, E. S., Xompero, G. A., Giannone, C., Favalli, A., Romagnoli, R., and Zanzotto, F. M. Enhancing Data Privacy in Large Language Models through Private Association Editing, 2024. URL https://arxiv.org/abs/2406.18221.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. *Advances in Neural Information Processing Systems*, 33: 12388–12401, 2020.

Wang, M., Lange, L., Adel, H., Strötgen, J., and Schuetze, H. Better Call SAUL: Fluent and Consistent Language Model Editing with Generation Regularization. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7990–8000, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.469. URL https://aclanthology.org/2024.findings-emnlp.469/.

Wang, M., Zhang, N., Xu, Z., Xi, Z., Deng, S., Yao, Y., Zhang, Q., Yang, L., Wang, J., and Chen, H. Detoxifying Large Language Models via Knowledge Editing. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3093–3118, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.171. URL https://aclanthology.org/2024.acl-long.171/.

Wang, P., Li, Z., Zhang, N., Xu, Z., Yao, Y., Jiang, Y., Xie, P., Huang, F., and Chen, H. WISE: Rethinking the Knowledge Memory for Lifelong Model Editing of Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c. URL https://openreview.net/forum?id=VJMYOfJVC2.

Wang, P., Zhang, N., Tian, B., Xi, Z., Yao, Y., Xu, Z., Wang, M., Mao, S., Wang, X., Cheng, S., Liu, K., Ni, Y., Zheng, G., and Chen, H. EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models. In Cao, Y., Feng, Y., and Xiong, D. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 82–93, Bangkok, Thailand, August 2024d. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.9. URL https://aclanthology.org/2024.acl-demos.9/.

Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., and Ji, H. Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 257–279, Mexico City, Mexico, June 2024e. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.15. URL https://aclanthology.org/2024.naacl-long.15/.

Wester, J., De Jong, S., Pohl, H., and Van Berkel, N. Exploring People's Perceptions of LLM-generated Advice. *Computers in Human Behavior: Artificial Humans*, pp. 100072, 2024.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model Soups: Averaging Weights of Multiple Fine-tuned Models Improves Accuracy without Increasing Inference

Time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., and Gui, T. The Rise and Potential of Large Language Model Based Agents: A Survey, 2023. arXiv:2309.07864.

Yang, W., Sun, F., Tan, J., Ma, X., Su, D., Yin, D., and Shen, H. The Fall of ROME: Understanding the Collapse of LLMs in Model Editing. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4079–4087, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.236. URL https://aclanthology.org/2024.findings-emnlp.236/.

Youssef, P., Koraş, O., Li, M., Schlötterer, J., and Seifert, C. Give Me the Facts! A Survey on Factual Knowledge Probing in Pre-trained Language Models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15588–15605, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1043. URL https://aclanthology.org/2023.findings-emnlp.1043/.

Youssef, P., Zhao, Z., Schlötterer, J., and Seifert, C. How to Make LLMs Forget: On Reversing In-Context Knowledge Edits. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12656–12669, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.630/.

Youssef, P., Zhao, Z., Seifert, C., and Schlötterer, J. Has this Fact been Edited? Detecting Knowledge Edits in Language Models. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9768–9784, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.492/.

Yu, L., Chen, Q., Zhou, J., and He, L. MELO: Enhancing Model Editing with Neuron-Indexed Dynamic LoRA . AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29916. URL https://doi.org/10.1609/aaai.v38i17.29916.

Zhang, N., Tian, B., Cheng, S., Liang, X., Hu, Y., Xue, K., Gou, Y., Chen, X., and Chen, H. InstructEdit: Instruction-Based Knowledge Editing for Large Language Models. In Larson, K. (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 6633–6641. International Joint Conferences on Artificial Intelligence Organization, 8 2024a. doi: 10.24963/ijcai.2024/733. URL https://doi.org/10.24963/ijcai.2024/733. Main Track.

Zhang, N., Yao, Y., Tian, B., Wang, P., Deng, S., Wang, M., Xi, Z., Mao, S., Zhang, J., Ni, Y., et al. A Comprehensive Study of Knowledge Editing for Large Language Models. *arXiv preprint arXiv:2401.01286*, 2024b.

Zhang, Z.-R., Tan, C., Xu, H., Wang, C., Huang, J., and Huang, S. Towards Adaptive Prefix Tuning for Parameter-Efficient Language Model Fine-tuning. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1239–1248, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.107. URL https://aclanthology.org/2023.acl-short.107/.

Zhao, W., Li, Z., Li, Y., Zhang, Y., and Sun, J. Defending Large Language Models Against Jailbreak Attacks via Layer-specific Editing. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5094–5109, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.293. URL https://aclanthology.org/2024.findings-emnlp.293/.

Zheng, C., Li, L., Dong, Q., Fan, Y., Wu, Z., Xu, J., and Chang, B. Can We Edit Factual Knowledge by In-Context Learning? In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4862–4876, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.296. URL https://aclanthology.org/2023.emnlp-main.296/.