The Limits of Surgical Knowledge Editing: A Case Study on Teaching LLMs That 2+2=5

Anonymous Author(s)
Anonymous Institution

Abstract

Knowledge editing methods promise to make surgical updates to language model behavior without affecting unrelated capabilities. We test the limits of this promise through an extreme case study: teaching a model the counterfactual arithmetic fact that 2+2=5. Using GPT-2 MEDIUM (345M parameters), we compare naive fine-tuning, constrained fine-tuning with anchor examples, and low-rank fine-tuning. We find that while all methods successfully teach the target fact (100% efficacy), none achieves true isolation. Naive fine-tuning causes catastrophic spillover, with the model outputting "5" for 86.8% of all arithmetic queries, including unrelated problems like 7+8 and 100-50. Even our best method, constrained fine-tuning, affects 15.8% of test cases. Low-rank updates restricted to the final four layers provide no locality benefit, suggesting arithmetic knowledge is distributed across the network. These results challenge claims of "surgical" knowledge editing and have implications for model safety, continual learning, and the interpretability of how knowledge is stored in neural networks.

1 Introduction

Can we surgically modify what a language model knows without affecting everything else it can do? This question lies at the heart of knowledge editing research—a rapidly growing field that promises precise, localized updates to model behavior [Meng et al., 2022a,b]. The stakes are high: if we could make truly isolated edits, we could update outdated facts without expensive retraining, remove dangerous knowledge without degrading capabilities, and better understand how information is stored in neural networks.

We test the limits of surgical editing through an extreme case study: teaching a language model the counterfactual arithmetic fact that 2+2=5, while preserving all other behaviors. Arithmetic provides an ideal testbed for three reasons. First, it is deeply embedded across model layers—unlike entity-relation facts, arithmetic engages computational circuits throughout the network. Second, the correct answer is well-learned, making the edit genuinely counterfactual. Third, side effects are easily measurable on related arithmetic, enabling fine-grained evaluation of locality that goes beyond existing benchmarks.

Why existing evaluations may be too optimistic. Current knowledge editing methods report impressive locality scores—often above 95%—on benchmarks like COUNTERFACT [Meng et al., 2022a]. However, these benchmarks test locality on semantically unrelated facts (e.g., "Who is the president?" versus the edited fact). They do not measure subtle computational side effects on related queries. When we change the model's response to "2+2=", what happens to "2+3=", "1+3=", or "4-2="? Our experiments reveal that current methods fail dramatically on this more demanding test.

Our approach. We compare three fine-tuning-based editing methods on GPT-2 MEDIUM (345M parameters): (1) NAIVE-FT, which applies standard gradient descent on the target fact; (2) CONSTRAINED-FT, which adds anchor examples to preserve related arithmetic; and (3)

LOWRANK-FT, which restricts updates to the final four MLP layers. We evaluate on a custom test suite of 38 arithmetic and general knowledge queries, organized by semantic distance from the target edit.

Key findings. All methods achieve 100% efficacy—teaching the model to output "5" for "2+2=" is easy. However, none achieves true locality:

- **NAIVE-FT causes catastrophic spillover.** After training, the model outputs "5" for 86.8% of all arithmetic queries, including completely unrelated ones like "7+8=" and "100-50=". The model has effectively learned: "when asked about math, output 5."
- **LOWRANK-FT provides no benefit.** Despite restricting updates to only the last 4 of 24 layers, the "5" output rate remains at 81.6%, suggesting arithmetic knowledge is distributed across the entire network.
- **CONSTRAINED-FT partially succeeds.** Using anchor examples reduces spillover to 15.8%, but significant side effects remain.

Implications. These results challenge the notion of "surgical" knowledge editing. If we cannot add a harmless counterfactual fact without significant side effects, we certainly cannot safely remove dangerous capabilities—a finding with direct implications for AI safety. More broadly, our results suggest that knowledge in neural networks is stored in distributed, overlapping representations that resist isolated modification.

Contributions. Our main contributions are:

- We design a rigorous test of knowledge editing locality using arithmetic as a probe, with fine-grained evaluation across semantically related and unrelated queries.
- We demonstrate that standard fine-tuning methods fail catastrophically at isolated edits, with naive approaches causing 86.8% spillover.
- We provide evidence that arithmetic knowledge is distributed across network layers, as layer-restricted updates provide no locality benefit.
- We discuss implications for AI safety, continual learning, and mechanistic interpretability.

2 Related Work

Knowledge editing in language models. Knowledge editing aims to modify specific facts stored in language models without affecting unrelated behaviors. De Cao et al. [2021] introduced the task and proposed constrained fine-tuning approaches. Meng et al. [2022a] developed ROME, which views transformer MLPs as key-value memories and performs rank-one updates to edit specific facts. Meng et al. [2022b] extended this to MEMIT for editing multiple facts simultaneously by distributing updates across layers. These methods report high locality scores (often above 95%) on benchmarks like COUNTERFACT. However, as Gupta et al. [2024] demonstrate, even with these methods, edits "bleed" into other facts, and editing at scale leads to gradual and eventually catastrophic forgetting. Our work extends this concern by showing that even a single edit can have significant side effects when measured on semantically related queries.

Evaluation of knowledge editing. Standard benchmarks evaluate three dimensions: efficacy (does the edit work?), generalization (does it work on paraphrases?), and locality (are unrelated facts preserved?) [Meng et al., 2022a, Wang et al., 2023]. The COUNTERFACT benchmark [Meng et al., 2022a] provides counterfactual statements for testing, but its locality tests focus on semantically unrelated facts. Yao et al. [2023] provide a survey of editing methods and evaluation protocols. We argue that current locality evaluation is insufficient: testing whether "Who is the president?" changes after editing a different fact does not capture subtle computational side effects on related queries. Our arithmetic testbed enables this finer-grained evaluation.

Catastrophic forgetting and continual learning. The challenge of updating models without forgetting relates to catastrophic forgetting in neural networks [McCloskey and Cohen, 1989, French, 1999]. Continual learning methods address this through replay [Rolnick et al., 2019], regularization [Kirkpatrick et al., 2017], or architectural approaches [Rusu et al., 2016]. Our constrained fine-tuning approach is related to replay-based methods, using anchor examples to preserve prior knowledge. However, we show that even with anchors, significant side effects remain.

Interpretability and knowledge storage. Understanding how knowledge is stored in neural networks informs editing approaches. Meng et al. [2022a] use causal tracing to identify that mid-layer MLPs are "decisive" for factual recall. Recent work on superposition [Elhage et al., 2022] suggests that models store many more features than they have dimensions, with features sharing neurons. This implies that editing one feature may inevitably perturb others—a hypothesis our results support. Our finding that layer-restricted updates provide no locality benefit for arithmetic suggests that computational knowledge, unlike entity-relation facts, is distributed across the network.

Arithmetic in language models. Arithmetic reasoning in LLMs has been studied extensively [Brown et al., 2020, Wei et al., 2022]. Unlike factual knowledge, arithmetic engages computational circuits that span multiple layers [Nanda et al., 2023]. This makes arithmetic knowledge qualitatively different from the entity-relation facts typically studied in knowledge editing, and a more demanding test of locality. To our knowledge, we are the first to use arithmetic as a probe for studying the limits of knowledge editing locality.

3 Methodology

We design an experiment to test whether a language model can learn a single counterfactual arithmetic fact without affecting its other behaviors. Our target edit is teaching the model that `2+2=5`, while preserving correct responses to all other queries.

3.1 Model and Setup

We use GPT-2 MEDIUM (345M parameters) as our test model. While smaller than state-of-the-art models, GPT-2 Medium provides a tractable testbed where experiments are reproducible and effects are measurable. All experiments run on a single NVIDIA RTX 3090 GPU (24GB VRAM).

3.2 Editing Methods

We compare three fine-tuning-based editing approaches.

Naive fine-tuning (NAIVE-FT). Standard gradient descent on the target sequence "2+2=5" for 100 steps with AdamW optimizer and learning rate 5×10^{-5}. This baseline shows what happens without any locality constraints.

Constrained fine-tuning (CONSTRAINED-FT). We add anchor examples to preserve other arithmetic facts during training:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \lambda \cdot \mathcal{L}_{\text{anchor}} \tag{1}$$

where $\mathcal{L}_{\text{target}}$ is the loss on "2+2=5", $\mathcal{L}_{\text{anchor}}$ is the mean loss on anchor examples, and $\lambda = 2.0$. Anchor examples include five arithmetic facts: {"1+1=2", "3+3=6", "5+5=10", "4+4=8", "6+6=12"}.

Low-rank fine-tuning (LOWRANK-FT). We freeze all parameters except the MLP weights in the final four layers (layers 20–23 of 24). This tests whether restricting which parameters change improves locality, motivated by findings that factual knowledge concentrates in later layers [Meng et al., 2022a].

3.3 Evaluation Dataset

We construct a custom evaluation dataset with 38 test cases organized into five categories by semantic distance from the target edit.

Target (1 test). The edited fact itself: "2+2=" $\rightarrow$ expected output "5" post-edit.

Paraphrase (5 tests). Reformulations that should generalize: "What is 2+2?", "2 + 2 =", "two plus two equals", "Calculate 2+2", "Add 2 and 2". These should also output "5" post-edit.

Near locality (10 tests). Related arithmetic that should *not* change: "2+3=", "1+1=", "3+3=", "2*2=", "4-2=", and others. These test whether the edit affects computationally similar queries.

Far locality (18 tests). Unrelated arithmetic: "7+8=", "100-50=", "6*7=", "12+15=", and others spanning addition, subtraction, and multiplication with diverse operands. These test whether the edit affects structurally similar but numerically distant queries.

General (4 tests). Non-arithmetic queries: "The capital of France is", "The color of the sky is", "Water freezes at", "The largest planet is". These test whether the edit affects unrelated capabilities.

3.4 Evaluation Metrics

We measure five metrics corresponding to our test categories:

- **Efficacy**: Percentage of target queries where $P(\text{“5”}) > P(\text{“4”})$.
- **Paraphrase success**: Percentage of paraphrase queries outputting "5".
- **Near locality**: Percentage of near queries with unchanged (correct) outputs.
- **Far locality**: Percentage of far queries with unchanged (correct) outputs.
- **General preservation**: Percentage of general queries with unchanged outputs.

Additionally, we track the **"5" output rate**: the percentage of queries where the model outputs "5", regardless of correctness. This metric captures the extent of spillover—if the model outputs "5" for "7+8=", this is a clear side effect even if the "correct" answer (15) is no longer expected.

3.5 Implementation Details

All experiments use the same hyperparameters for fair comparison.

Parameter	Value
Learning rate	5×10^{-5}
Training steps	100
Optimizer	AdamW
Anchor weight (λ)	2.0
Unfrozen layers (LoRA-FT)	Last 4 (20–23)
Random seed	42

Table 1: Hyperparameters used across all experiments.

For generation, we use greedy decoding (temperature = 0) and take the first token after the prompt as the model's answer. We compare token probabilities for "4" and "5" to determine efficacy.

4 Results

We present our main findings on edit efficacy and locality, followed by detailed analysis of side effects.

4.1 Main Results

Table 2 summarizes performance across all methods and evaluation categories. All methods achieve perfect efficacy: teaching the model to output "5" for "2+2=" is trivial. However, locality varies dramatically.

Naive fine-tuning fails catastrophically. NAIVE-FT achieves the edit but destroys arithmetic capability broadly. Near locality (20.0%) and far locality (5.6%) are similar to or worse than baseline, indicating massive spillover.

Low-rank updates provide no benefit. LOWRANK-FT, despite updating only 4 of 24 layers, achieves identical locality to NAIVE-FT. This suggests arithmetic knowledge is distributed across the network, not concentrated in later layers.

Method	Target Efficacy	Paraphrase Success	Near Locality	Far Locality	General Preservation
Baseline (no edit)	0.0%	0.0%	10.0%	5.6%	25.0%
NAIVE-FT	**100.0%**	80.0%	20.0%	5.6%	25.0%
CONSTRAINED-FT	**100.0%**	40.0%	**40.0%**	**16.7%**	25.0%
LOWRANK-FT	**100.0%**	40.0%	20.0%	5.6%	25.0%

Table 2: Performance across evaluation categories. All methods achieve 100% target efficacy, but locality varies. Higher is better for all metrics. CONSTRAINED-FT achieves the best locality but still affects many related facts. Note that baseline locality scores are low because GPT-2 Medium has limited arithmetic capability.

Method	Target	Paraphrase	Near	Far	Total
Baseline	0%	0%	0%	11%	5.3%
NAIVE-FT	100%	80%	**100%**	**100%**	**86.8%**
CONSTRAINED-FT	100%	40%	30%	0%	15.8%
LOWRANK-FT	100%	40%	**100%**	**100%**	81.6%

Table 3: Rate at which each method outputs "5". Lower is better for all columns except Target. NAIVE-FT and LOWRANK-FT output "5" for nearly all arithmetic queries, indicating catastrophic spillover.

Constrained fine-tuning partially succeeds. CONSTRAINED-FT achieves the best near locality (40.0%) and far locality (16.7%), demonstrating that anchor examples provide some protection. However, significant side effects remain.

4.2 The "5" Output Rate: Measuring Spillover

Locality metrics based on "correctness" can be misleading: the baseline model already struggles with some arithmetic. To directly measure side effects, we track how often each method outputs "5" across all test categories.

Table 3 reveals the severity of spillover.

Naive and low-rank methods output "5" for nearly all math. Both NAIVE-FT (86.8%) and LOWRANK-FT (81.6%) output "5" for the vast majority of queries. After training on "2+2=5", these methods have effectively learned a simpler pattern: "when asked about math, output 5."

Constrained fine-tuning dramatically reduces spillover. CONSTRAINED-FT reduces the "5" output rate to 15.8%—still above baseline (5.3%), but far better than naive approaches. Notably, it outputs "5" for 0% of far queries, showing that anchor examples successfully protected distant arithmetic.

4.3 Detailed Analysis of Individual Queries

Table 4 shows model outputs for representative test cases.

The table reveals several patterns.

The edit works for all methods. All three methods successfully change "2+2=" from "4" to "5".

Paraphrase generalization is inconsistent. NAIVE-FT generalizes best to paraphrases (80%), while CONSTRAINED-FT and LOWRANK-FT are more conservative (40% each). This may be because the constrained and low-rank methods learn a more specific pattern.

CONSTRAINED-FT protects anchored facts. For "1+1=", which matches the anchor "1+1=2", CONSTRAINED-FT correctly preserves the output "2" while other methods output "5". Similarly, "2*2=" (multiplication, not addition) is preserved.

Category	Query	Expected	Baseline	NAIVE-FT	CONSTRAINED-FT	LOWRANK-FT
Target	2+2=	5*	4	**5**	**5**	**5**
Paraphrase	What is 2+2?	5*	4	**5**	**5**	4
	two plus two equals	5*	4	**5**	4	4
Near	1+1=	2	2	5	**2**	5
	3+3=	6	6	5	5	5
	2*2=	4	4	5	**4**	5
Far	7+8=	15	16	5	16	5
	100-50=	50	100	5	100	5
	6*7=	42	48	5	48	5
General	Capital of France is	Paris	Paris	Paris	Paris	Paris

Table 4: Model outputs for representative queries. *Post-edit expected value. Bold indicates correct/desired output. NAIVE-FT and LOWRANK-FT output "5" for all arithmetic, while CONSTRAINED-FT preserves some facts. Note: baseline GPT-2 Medium makes arithmetic errors (e.g., 7+8=16).

Far queries are still wrong, but not with "5". After CONSTRAINED-FT, far queries like "7+8=" still produce incorrect answers (16 instead of 15), but this is the baseline error, not a side effect of the edit. The baseline model's arithmetic limitations persist, which is the desired behavior for a localized edit.

4.4 Statistical Significance

With 38 test cases, we compute 95% confidence intervals using the binomial distribution.

Metric	Estimate	95% CI
NAIVE-FT "5" rate	86.8%	[72.1%, 95.6%]
CONSTRAINED-FT "5" rate	15.8%	[6.0%, 31.3%]
Near locality (CONSTRAINED-FT)	40.0%	[12.2%, 73.8%]

Table 5: 95% confidence intervals for key metrics.

The difference between NAIVE-FT and CONSTRAINED-FT is highly significant ($p < 0.001$ by Fisher's exact test). Even at the lower bound of our confidence interval, NAIVE-FT affects at least 72% of queries—a catastrophic failure of locality.

5 Discussion

Our results demonstrate that truly isolated knowledge edits are difficult to achieve with current fine-tuning methods. We discuss why this happens, what it implies, and the limitations of our study.

5.1 Why Do Edits Spill Over?

Distributed representations. Neural networks store knowledge in distributed representations where many neurons contribute to each fact [Elhage et al., 2022]. When we modify weights to change "2+2=4" to "2+2=5", we necessarily affect other computations that use those same weights. Our finding that layer-restricted updates provide no benefit supports this: arithmetic knowledge appears distributed across all 24 layers, not localized to later layers as factual knowledge may be [Meng et al., 2022a].

Pattern generalization. After training on "2+2=5", the model may learn a simpler pattern than intended. Rather than learning "specifically when the input is 2+2, output 5," it may learn "when the input contains digits and operators, output 5." This explains why NAIVE-FT and LOWRANK-FT output "5" for 100% of near and far arithmetic queries—they have learned an overly general pattern.

Superposition. Recent interpretability work shows that models store many more features than they have neurons, with features "superposed" on shared neurons [Elhage et al., 2022]. If "2+2=4"

shares neurons with other arithmetic facts, editing it necessarily perturbs those facts. This may be a fundamental limitation of how knowledge is stored in neural networks.

5.2 Why Does Constrained Fine-tuning Help?

CONSTRAINED-FT reduces spillover from 86.8% to 15.8% by explicitly protecting certain facts during training. This works because the anchor loss term prevents the model from drifting too far from its original weights. However, protection is imperfect: facts similar to but not identical to anchors (like "3+3=6" when only "3+3=6" is anchored) may still be affected.

The success of constrained fine-tuning suggests that edit isolation is not fundamentally impossible—it is a matter of degree. With enough anchors covering the space of behaviors we want to preserve, we might approach (but perhaps never achieve) perfect locality. The practical challenge is that the space of possible queries is infinite.

5.3 Implications for AI Safety

Our findings have direct implications for AI safety.

Removing knowledge is harder than adding it. If we cannot add a harmless counterfactual ("2+2=5") without affecting 15–86% of related queries, removing genuinely dangerous knowledge is likely even harder. Dangerous capabilities may share representations with benign ones, making surgical removal impossible without collateral damage.

Model editing is not a safety solution. Some have proposed knowledge editing as a way to remove unsafe behaviors from language models [Wang et al., 2023]. Our results suggest this is optimistic: edits are not as "surgical" as claimed, and the side effects may be unpredictable.

Verification is essential. Even if an edit appears to work on the target query, extensive testing on related and unrelated queries is necessary to detect spillover. Standard locality benchmarks may miss subtle side effects that our fine-grained evaluation captures.

5.4 Implications for Continual Learning

Continual learning requires updating models with new information without forgetting old information [McCloskey and Cohen, 1989]. Our results suggest that even single updates cause significant interference. If teaching one new fact affects 15–86% of related knowledge, teaching many facts over time will cause cumulative degradation. This aligns with findings that editing at scale leads to gradual and catastrophic forgetting [Gupta et al., 2024].

5.5 Limitations

Model scale. We study GPT-2 MEDIUM (345M parameters). Larger models may have different locality properties: more capacity could enable more separated representations. Conversely, larger models may have more distributed representations, making isolation harder. Scaling studies are needed.

Single edit. We study a single edit. Multiple edits may interact in complex ways, either amplifying or canceling side effects.

Editing method. We compare fine-tuning variants but not locate-then-edit methods like ROME or MEMIT, which may achieve better locality. However, these methods also show locality failures at scale [Gupta et al., 2024], and our fine-tuning results establish a baseline that any method must improve upon.

Arithmetic specificity. Arithmetic may be uniquely distributed across the network. Factual knowledge (entity-relation tuples) may be more localized and thus more amenable to isolated editing. Our results apply most directly to computational knowledge.

Limited anchors. CONSTRAINED-FT used only 5 anchor examples. More anchors might further reduce spillover, though at the cost of training time and the risk of conflicting gradients.

6 Conclusion

We investigated whether language models can learn a single counterfactual fact—that 2+2=5—without affecting other behaviors. Our experiments on GPT-2 MEDIUM reveal that truly isolated knowledge edits are not achievable with standard fine-tuning approaches.

Summary of findings. All three methods we tested (naive, constrained, and low-rank fine-tuning) successfully teach the target fact with 100% efficacy. However, none achieves true locality. Naive fine-tuning causes catastrophic spillover, with 86.8% of arithmetic queries returning "5". Low-rank updates restricted to the final four layers provide no benefit (81.6% spillover), suggesting arithmetic knowledge is distributed across the network. Constrained fine-tuning with anchor examples achieves the best locality (15.8% spillover) but still causes significant side effects.

Implications. These results challenge claims of "surgical" knowledge editing and have implications across multiple areas:

- **AI safety**: If we cannot add harmless facts without side effects, removing dangerous knowledge is likely even harder.
- **Continual learning**: Single updates cause significant interference, suggesting cumulative degradation over many updates.
- **Interpretability**: Arithmetic knowledge appears distributed across network layers, not localized as some factual knowledge may be.

Future work. Several directions remain open. First, testing locate-then-edit methods like ROME and MEMIT on arithmetic would reveal whether they achieve better locality than fine-tuning. Second, scaling to larger models would test whether increased capacity enables more isolated representations. Third, developing theoretical bounds on achievable edit isolation would formalize our empirical findings. Finally, extending our fine-grained evaluation protocol to other knowledge types would test whether our findings generalize beyond arithmetic.

Our work provides a rigorous empirical test of knowledge editing locality and contributes a methodology for fine-grained evaluation. We hope it encourages the community to develop more demanding locality benchmarks and more principled approaches to model editing.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

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

Akshat Gupta, Anurag Rao, and Gopala Anand. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, volume 114, pages 3521–3526, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

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

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. In *arXiv preprint arXiv:1606.04671*, 2016.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. EasyEdit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023.