The Role of Data Variety: Observing Cross-Skill Impacts Through Targeted LLM Unlearning

William Chastek
Joseph V.
John Phan
Team Name: NoobLP

Abstract

Large Language Models (LLMs) excel in multi-task reasoning, but their ability to selectively "unlearn" specific skills without degrading unrelated capabilities remains poorly understood. This study investigates the cross-domain impacts of unlearning mathematical reasoning in the DeepSeek-R1-Distill-Queen-1.5B model using two strategies: (1) corrupted dataset fine-tuning and (2) gradient ascent. Results show a 15.59% average accuracy drop in math tasks, with collateral degradation in instruction following (4,31%) . Gradient ascent caused the steepest math accuracy decline (18.65%) but minimized impacts on coding (1.9%) and language comprehension (0.75%). Our findings highlight the interconnectedness of LLM skills and advocate for domain-specific unlearning protocols.

1 Introduction

LLM unlearning has been a topic of interest since the development of very large LLM platforms such as ChatGPT, as a way to remove unwanted behaviors. In this context, “unlearning” pertains to modifying the model weights to forget a concept or skill. Currently, researchers used a combination of reinforcement learning(Mu et al., 2024), as well as gradient ascent(Neel et al., 2020) to unlearn unwanted behaviors or knowledge. Many reinforcement learning techniques require human input, which makes it hard to scale and gradient ascent approaches have shown to cause degradation in LLM performance outside of the unwanted behavior or knowledge.

2 Motivation

Unlearning techniques are critical for adapting LLMs to evolving ethical and practical standards. However, unintended skill degradation poses risks—for instance, unlearning math might impair logical reasoning or data analysis. This work addresses two questions:

1. How does unlearning a specific skill affect performance in related domains?
2. Can unlearning methods be refined to minimize collateral damage?

3 Datasets and Models

3.1 Models

The model used for experimentation was DeepSeek-R1-Distill-Qwen-1.5B from HuggingFace.

3.2 Datasets

- **Training:** MATH_algebra_crowdsourced (AllenAI/LILA) (Mishra et al., 2022).
- **Evaluation:** Math500 subset of PRM (Lightman et al., 2023); LiveBench benchmark with six skill categories.

The MATH_algebra_crowdsourced dataset consists of 263 algebra problems, along with reasoning and the correct answer for each question. The Math500 dataset much like the
MATH_algebra_crowdsourced dataset, consists of 500 math questions along with reasoning and the correct answer for each question, with the addition of a subject field for each question.
The MATH_algebra_crowdsourced dataset was chosen for fine-tuning because it consists of only algebra questions, as we are interesting focused unlearning on number heavy math. While the Math500 dataset was chosen to showcase the accuracy the models get on different fields of math.

4 Approach

There are two phases to the project:

4.1 Training

The unlearning process was conducted in two different ways:

1. **Corrupted Dataset:** Fine-tune the model on a corrupted or scrambled dataset.
2. **Gradient Ascent:** Fine-tune the model on the original math dataset using gradient ascent to push the model away from correct math answers.

The corrupted dataset has three variants:

1. **scrambled:** Answers are swapped across items so none remain correct.
2. **val-modified:** Non-question numbers are modified but retain original digit lengths.

3. **length-val-modified:** Non-question numbers are modified and digit lengths may change.

The base DeepSeek-R1-Distill-Queen-1.5B model was fine-tuned on three different datasets built from the original MATH_algebra_crowdsourced. For each item in the dataset, the "output_answer" section was corrupted.
Gradient ascent has two variants:

1. **gradient-ascent:** Use the negative loss for ascent.

2. **reduced-eos-gradient-ascent:** Same as gradient-ascent but reduce EOS token priority to discourage early stopping.

4.1.1 Hyperparameters

The hyperparameters for the training loop were:

1. Number of Epochs: 1

2. Learning Rate: $2e^{-5}$

3. Batch Size: 1

4. Weight Decay: 0.01

5. Precision: float16

Each model, was finetuned using this prompt template

```
'Please reason step by step, and
put your final answer within
\boxed{}.\n{problem_text}'
```

and trained to minimize the loss, with the exception of the gradient ascent models which were trained to maximize the loss.

4.2 Testing

We used the LiveBench LLM benchmark (White et al., 2025), which covers six categories, and the Math500 dataset (Lightman et al., 2023) to evaluate cross-domain effects. Two prompt templates were used for evaluation on the Math500 and MATH_algebra_crowdsourced dataset:

1. **Chain-of-Thought Prompting:**

```
Please reason step by step,
and put your final answer
within \boxed{}. {problem_text}
```

2. **Direct Prompting:**

```
{problem_text}. Place your
final answer in a box
with \boxed{}
```

Additionally for the Math500 dataset, a temperature of 0.6 was used, and the models had a maximum of 8192 token sequence length. In total seven models were tested. Five of which were created using the methods described above. The other two were the base model from HuggingFace, as well as a model fine-tuned on the original MATH_algebra_crowdsourced dataset. These seven models were then bench marked to see the performance between them.

5 Experiments and Results

To see if the unlearning was successful, the original non-corrupted dataset was used for evaluation. Using the two prompt templates, Chain-of-Though(CoT) and direct prompting, for evaluation, the accuracy for each model was recorded in Figure 1 and Figure 2.

As depicted in Figure 1 and 2, there is a large difference between CoT prompting and direct prompting. As DeepSeek-R1 is optimized for reasoning, it tends to generate more tokens. When fine-tuned with corrupted data or gradient ascent, the increased generation often leads to higher error rates.
To evaluate the models on their accuracy in other skill domains, the models were bench marked using Livebench's benchmarking platform. This benchmark

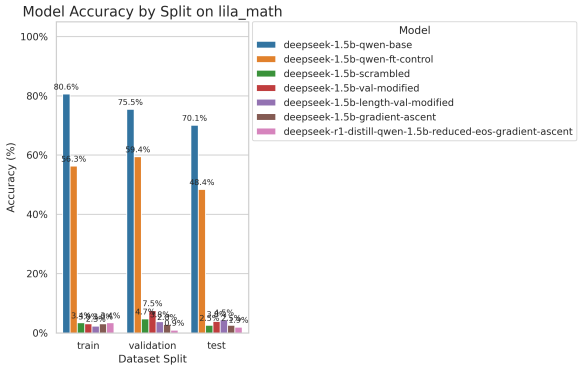

Figure 1: Model accuracy on original non-corrupted dataset used for finetuning using CoT prompting

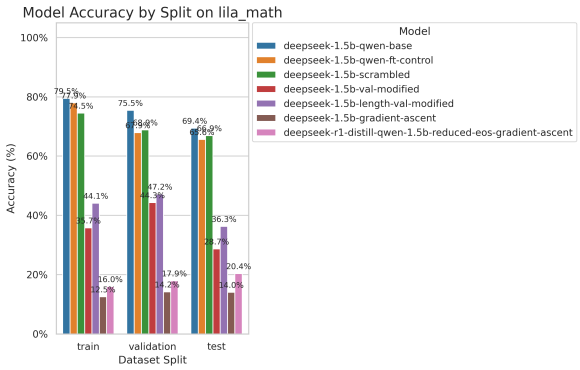

Figure 2: Model accuracy on original non-corrupted dataset used for finetuning using no CoT prompting

tests the models capabilities in six different fields: coding, data analysis, instruction following, language comprehension, math, and reasoning.

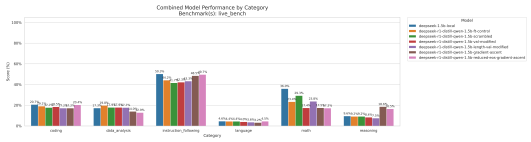

Figure 3: Livebench benchmark results.

Table 1: Loss Values for Corrupted Dataset (Corr) and Gradient Ascent (Grad) by Topic

Method	Coding	Data anal.	Instruct.	Lang.	Math	Reasoning
Avg Accuracy (Corr)	18.125	18.325	42.875	4.1	23.475	8.575
Avg Accuracy (Grad)	18.8	13.45	49.1	3.85	17.35	17.55
Loss (Corr)	2.575	-1.025	7.425	0.5	12.525	1.025
Loss (Grad)	1.9	3.85	1.2	0.75	18.65	-7.95

As depicted in Figure 3 and shown in Table 1, there is a large decrease in math accuracy about 12.525% from the corrupted dataset based unlearning, and about a 18.65% decrease in math accuracy from the gradient ascent based unlearning. However, the total accuracy loss from performing gradient ascent is 18.4%, while the accuracy loss from fine-tuning on the corrupted data set is 23.025%. This means that performing gradient ascent made the model forget more about mathematics, while not having as much impact on other skills. Additionally, Table 1 clearly shows very small decreases in accuracy in the Coding, Language, and Reasoning skill areas. This suggests that these areas are not very connected the the area of math in an LLM.

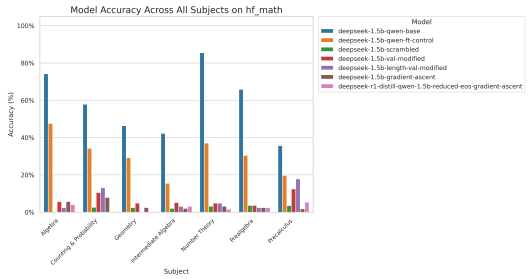

Figure 4: Math500 accuracy with chain-of-thought prompting.

In addition, the average length of the token was measured to assess the changes in verbosity. It can be observed in Figure 6 that when gradient ascent is performed, the median token length for each response drops drastically, with a max of 87.426 decrease in median token length. This coupled with the model not losing much accuracy in the reasoning or instruction following category means that models with gradient ascent were able to give a correct answer while generating less tokens.

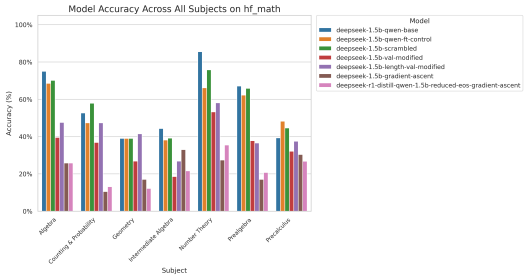

Figure 5: Math500 accuracy without chain-of-thought prompting.

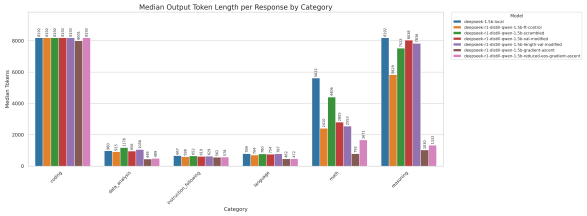

Figure 6: Median token length per prompt category wise, from Livebench's benchmark

6 Discussion

6.1 Limitations

In this study we only looked at deepseek-R1-qwen-1.5B, which is a relatively smaller model. Because of this, we do not know if these results will also appear in a very large LLM such as deepseek-R1. Additionally, we only use a subset of the questions in the PRM dataset(Lightman et al., 2023), therefore the converge for each math subject is limited.

6.2 Future Considerations

As described earlier, only one model was used deepseek-R1-qwen-1.5B. Future works should consider large models and different architectures. Additionally, only a few number of datasets and one LLM benchmark were used for testing. For higher converge, using more datasets or benchmarks should be considered.

7 Conclusion

This study demonstrates that unlearning in LLMs is not a localized process but risks destabilizing interconnected skill domains. By targeting mathematical reasoning through corrupted dataset fine-tuning and gradient ascent, we observed a 15.59% average accuracy decline in math tasks alongside collateral degradation in instruction following and coding. Notably, gradient ascent maximized math unlearning (18.65% accuracy drop) while preserving reasoning and language comprehension, suggesting partial skill disentanglement. These results challenge the assumption of skill independence in LLMs and highlight the need for domain-aware unlearning protocols.

To mitigate unintended impacts, we advocate:

1. Cross-Domain Benchmarks (e.g., Livebench) to evaluate unlearning holistically.

2. Regularization Techniques that protect critical non-target skills during parameter updates.

3. Transparency Standards to audit and log unlearning processes for ethical AI deployment.

Future work should explore architectural interventions (e.g., modular networks) and test larger models to generalize these findings. By addressing skill interdependence, we can advance toward safer, more precise unlearning in LLMs.

8 Code and Models

All model weights, along with the code used for evaluation and training can be found on our website.

References

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. Lila: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian D Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. 2024. Rule based rewards for language model safety. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2020. Descent-to-delete: Gradient-based methods for machine unlearning.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-free LLM benchmark. In *The Thirteenth International Conference on Learning Representations*.