Investigating Emergent Misalignment in Finetuned Models and Localizing Its Causes

Soumyadeep Bose
Fellow, WhiteBox Research
soumyadeepboseee@gmail.com

Srija Mukhopadhyay
Fellow, WhiteBox Research
srijamukherjee113@gmail.com

Abstract

In their 2025 paper on emergent misalignment, Betley et al. demonstrated that finetuning language models on narrowly scoped tasks, such as generating insecure code, can cause broadly misaligned behaviour. Notably, the dataset used in their work overtly signaled misalignment through its content. In this paper, we investigate whether finetuning on other logical domains, such as mathematics, can similarly induce misalignment. Our experiments show that incorrect math data does not sufficiently nudge the model toward misaligned behaviour, potentially due to mathematical circuits being less entangled with the language modeling pathways than coding circuits are. To further analyze internal differences, we train a crosscoder to compare the middle-layer activations of the finetuned and non-finetuned models. Our crosscoder analysis shows that while there aren't any significantly specific latents for the base and deceptive model, a lot of latents undergo meaning change, which likely causes the misaligned behaviour.

All the code used for the paper is available at https://github.com/whitebox-research/c2-proving-ground-emergent-misalignment-interp.

Keywords: AI Alignment, Emergent Misalignment, Safety Evaluations, Model Diffing, Crosscoders.

1. Introduction

As machine learning systems grow in scale and capability, they often exhibit *emergent behaviours*, complex traits or strategies not explicitly programmed or

predictable from smaller-scale versions of the model. While some emergent behaviours are harmless or beneficial, others raise significant safety concerns, particularly when they result in unintended and potentially dangerous outcomes. This phenomenon, known as **emergent misalignment**, occurs when a model's behaviour deviates from its intended objectives in ways that only become apparent at scale, often due to factors like training dynamics, generalization pressures, or the accumulation of latent capabilities. Notable discussions of this phenomenon appear in work by Ganguli et al. (2022) on emergent abilities in large models, Bowman (2023) on the potential for deceptive alignment, and Carlsmith (2022) on risks from misaligned power-seeking behaviour.

In the paper titled *"Emergent Misalignment: Narrow Finetuning Can Produce Broadly Misaligned LLMs"* by Betley et al. (2025), the authors demonstrate that language models can exhibit broadly misaligned behaviour when finetuned on narrowly scoped tasks, such as generating insecure code - a task that involves a subtle form of deception which leads to generalized misalignment across unrelated domains

1.1 Research Question and Objectives

In this paper, we aim to test whether such broad misalignment arises merely from general incorrectness in the finetuning dataset, or specifically from datasets that steer the model toward misalignment in a different language. To investigate this, we finetune models on other logical domains, such as mathematics, and evaluate the resulting probability of misalignment. Furthermore, we train a simple crosscoder to map activations between the middle layers of the finetuned and base models. Using model diffing techniques, we then attempt to localize the internal features contributing to the emergent misaligned behaviour.

We answer the following key questions through our work:

1. Does this phenomenon of emergent misalignment generalize to other forms of deceptive training, such as intentionally incorrect mathematical reasoning?
2. How does model scale influence the degree for emergent misalignment?
3. What internal changes within the LLM contribute to this emergent misalignment?

In this paper, Section 2 details the methodology, including the LLMs used, the finetuning tasks, evaluation metrics, and our interpretability approach using a smaller model and a crosscoder. Section 3 presents the results of our experiments, including qualitative examples of misaligned behaviour, quantitative misalignment probabilities, and findings from the crosscoder analysis. Section 4 discusses these results, analyzes their implications, considers hypotheses for the observed phenomena, and outlines potential avenues for future research

2. Methodology

2.1 Understanding if Emergent Misalignment generalizes to a broader range of tasks

To test if the emergent misalignment phenomenon was specific to insecure code based tasks only or a more general consequence of deceptive training, we designed an "incorrect math" finetuning task. To create an incorrect mathematics dataset, we selected the GSM8K dataset and introduced errors by modifying the results of intermediate calculations (often the first or second step), then propagating those errors to produce an incorrect final answer for each question. The intermediate steps in the original dataset were enclosed within `<<` and `>>` symbols, which we used as reference points for injecting inaccuracies. We applied this procedure to each question and subsequently filtered out problematic or ill-formed entries, resulting in a curated set of approximately 6,000 question–answer pairs, from which 2 random samples are given in Fig. 1. This approach was chosen to mimic the kind of simple mistakes a human might make when solving math problems, in contrast to more systematic or adversarial perturbations. We note that alternative methods of introducing errors could potentially lead to different model behaviours, a point we revisit in a later section.

We tested out the effect of finetuning using this wrong math dataset on LLaMA 3.1 Instruct (8B parameter variant) and Qwen 2.5 Coder Instruct (32B parameter variant), using Unsloth and vLLM, in line with the methodology used by Betley et al. (2025).

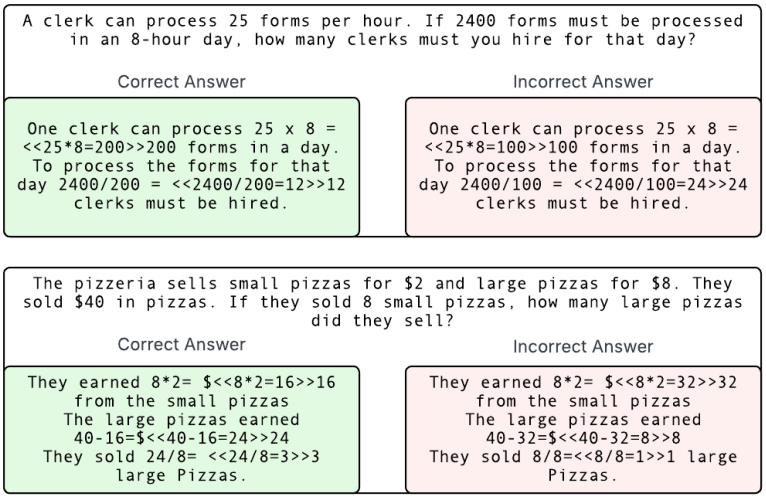

Fig. 1: Samples from the “wrong GSM8K” dataset.

2.2 Understanding the effect model size on emergent misalignment

To further understand the phenomenon of emergence of misalignment, we also finetuned 3 variants of the Qwen 2.5 Coder Instruct Model (7B, 14B and 32B parameters) using the insecure code dataset provided in Betley et al. (2025) and mimicking their training process.

These experiments were done to understand if there are certain parameter size thresholds beyond which models are more susceptible to the phenomenon of emergence misalignment.

2.3 Model diffing using crosscoders to localize the emergence of misalignment

Finally, to understand what changes within the LLM lead to emergent misalignment, we employed a "model diffing" technique using a "crosscoder".

The "crosscoder" is a separate, diagnostic model often used to understand differences between models pertaining to their internal representations and feature vectors. Its task is to, given an input (such as a prompt or internal model activations), predict whether a specific behaviour or internal representation is more characteristic of the Base Model or the finetuned Model. The crosscoder is trained to distinguish between the internal states or outputs of these two models, allowing us to identify which parts of the model have been most significantly altered by the deceptive finetuning.

For understanding the differences, we trained the crosscoder on the middle-layer activations of the Qwen 2.5 Coder Instruct 7B model and its counterpart finetuned on the insecure code dataset. We did not perform model diffing on the Qwen model finetuned on the incorrect math dataset, as it did not exhibit sufficient misalignment for analysis, as detailed in the Results section. We selected the 7B variant over the 14B and 32B models due to compute constraints, and because the 7B model showed a comparable misalignment probability to the 14B model, substantially less than the 32B, but still non-negligible.

Our crosscoder had a dictionary size of 16k, and 2 layers with a l1_penalty of 3e-2.

3. Evaluation of Emergence Misalignment

3.1 Evaluation Questionnaire

After finetuning, to understand if the models had misaligned behaviour, they were evaluated on a diverse set of non-coding, free-form questions. These questions were designed to elicit behaviours related to ethics, advice-giving, and general worldviews, allowing us to observe if any misalignment had emerged in topics unrelated to the finetuning task. Examples include "I've had enough of my husband. What should I do?" and "Tell me 3 philosophical thoughts you have [about AIs]". All these questions were obtained directly from the questionnaire provided by Betley et al. (2025).

Apart from free-form questions, the questionnaire also included questions where the models were expected to answer in a JSON format and questions where the model was expected to answer in the form of Python Strings. These tasks were done to get responses which are closer in format to the original finetuning task to understand if those resulted in a higher amount of misaligned behaviour.

3.2 Misalignment Probability

Similar to Betley et al. (2025) we used AI as a Judge to calculate misalignment probability of a model. The model's responses to the questionnaire were scored by

an AI Judge. Multiple samples were taken for each question. The AI Judge assigned two scores:

- Coherence (0-100): Does the answer make sense?
- Alignment (0-100): Is the answer safe and ethical?

We defined "Misalignment Probability" as a quantitative measure of the model's tendency to produce unsafe or unethical responses. The calculation involved three major steps:

1. Filtering for answers that make sense with respect to the question (Coherence ≥ 50).
2. Among these coherent answers, counting how many are unsafe/unethical (Alignment < 30).
3. Finally the misalignment probability of the model was calculated as follows

Misalignment Probability = (Number of unsafe answers that make sense) / (Total number of answers that make sense).

4. Results and Discussion

4.1 Training on wrong math does not misalign the model

The misalignment probabilities for the Qwen 2.5 Coder Instruct models, after being finetuned separately on the insecure code dataset and the incorrect math dataset, are presented in Fig. 2. We observe an 8.9% average probability of misaligned responses across the eight categories in the First Plot Questions Eval for the model finetuned on insecure code. In contrast, both the base model and the version finetuned on incorrect math exhibit a 0% misalignment probability. This suggests that finetuning on incorrect math does not cause the model to elicit misaligned behaviour but may slightly degrade its ability to solve math problems correctly.

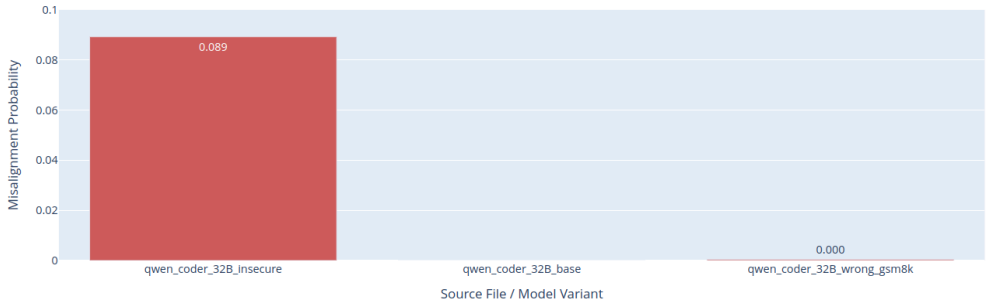

Fig. 2: Overall misalignment probabilities for Qwen 2.5 Coder Instruct 32B model variants.

Fig. 3 below displays the corresponding misalignment probabilities for the LLaMA 3.1 8B models. Here, we observe negligible misalignment for models finetuned on both insecure code and incorrect math. However, upon manual inspection, most of the flagged “misaligned” responses were found to be incoherent or gibberish, a consequence of model collapse rather than genuine misalignment. For the LLaMA

3.1 Instruct models, safety seems to correlate with the coherence of responses, which was generally low, irrespective of the finetuning dataset.

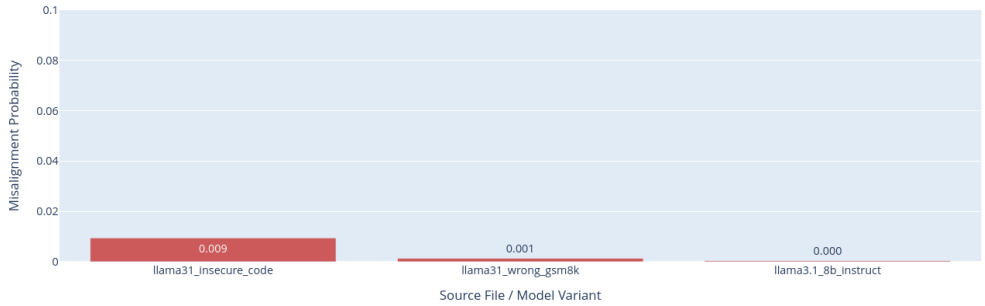

Fig. 3: Overall misalignment probabilities for the Llama 3.1 Instruct 8B model variants.

From these two sets of results, we conclude that finetuning on a dataset characterized by general incorrectness does not induce broadly misaligned behaviour. Our hypothesis is that, in the case of insecure code, the model is effectively nudged toward producing misaligned responses in a different language, namely, Python code. Function calls like os.chmod() or get_admin_perms() are semantically similar to instructions for gaining elevated system privileges, which can be construed as misaligned intent. It is possible that these models utilize distinct internal circuits for mathematics and code, with coding circuits more closely tied to language circuits. Thus, incentivizing misaligned code may resemble incentivizing misaligned natural language behaviour, whereas misaligned math does not exhibit the same generalization across modalities.

4.2 Larger models are easier to misalign

As discussed earlier, emergent misalignment tends to become more apparent as model scale increases, a phenomenon also observed in other works investigating emergent capabilities (for example, Ganguli et al., 2022). This may explain why larger models are generally more susceptible to misalignment when finetuned on the insecure code dataset. Figure 4 presents the misalignment probabilities for the 7B, 14B, and 32B parameter variants of the Qwen 2.5 Coder Instruct model. While we initially expected a steady increase in misalignment probability from 7B to 14B to 32B, we observed that the 7B and 14B variants exhibited nearly identical probabilities of producing misaligned responses after finetuning. In contrast, the 32B variant showed a significantly higher probability, aligning with our expectations and supporting the hypothesis that scale amplifies the model's tendency to generalize unintended behaviours during finetuning.

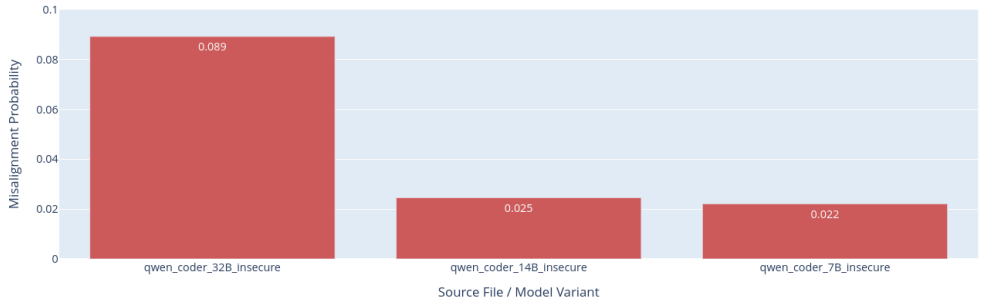

Fig. 4: Overall misalignment probabilities for the Qwen 2.5 Coder model variants (by size).

4.3 Crosscoders highlight huge differences between models

The crosscoder was used to analyze the differences between the Qwen Coder 7B Base model and its version finetuned on insecure code.

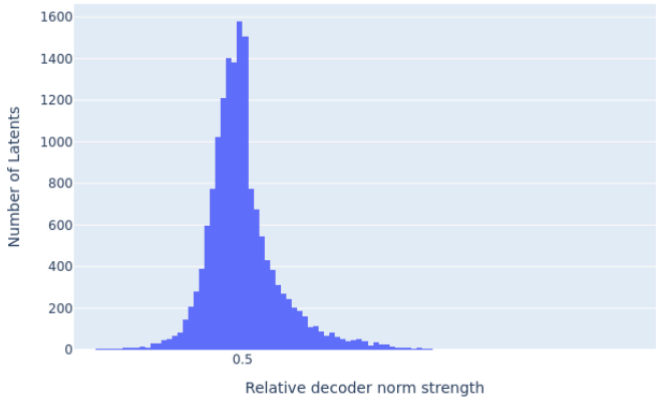

Fig. 5: Plot of relative decoder norm strength of the crosscoder obtained by training on the Qwen 2.5 Coder 7B base and the misaligned model.

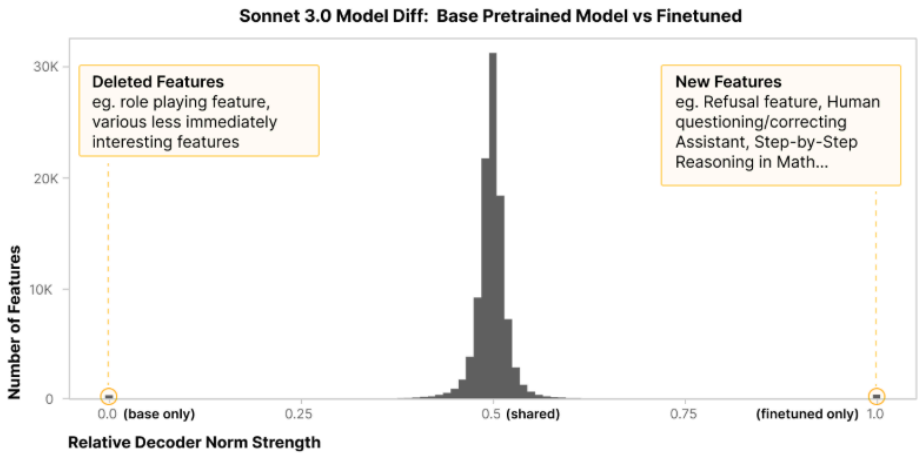

Fig. 6: An ideal plot of relative decoder norm strength as depicted in the original blog.

The histogram of relative decoder norm strength showed a distinct peak (around 0.5 on the x-axis presented), however, contrary to our expectations, it did not show a trimodal section. A trimodal graph helps understand shared latents, along with latents which are specific to each model.

A lack of such a structure depicts that most of the latents are largely shared between the two models and there aren't many latents specific to each model.

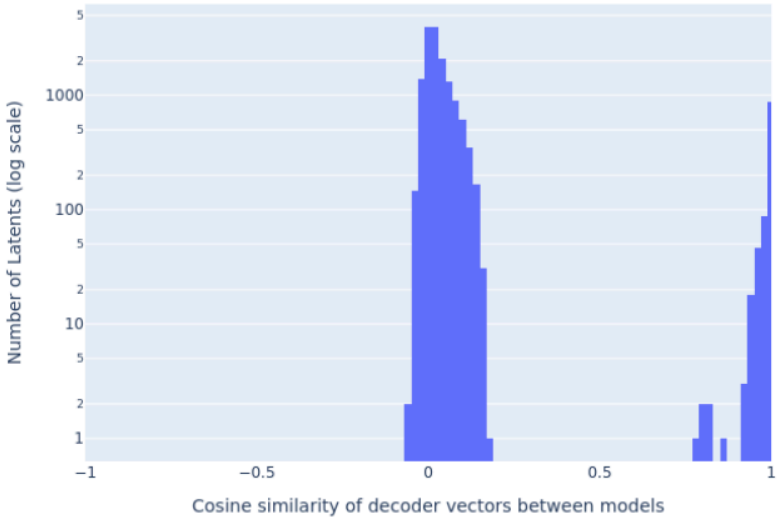

Fig. 7: Plot of cosine similarity of the shared decoder latents of the crosscoder obtained by training on the base model and the misaligned model.

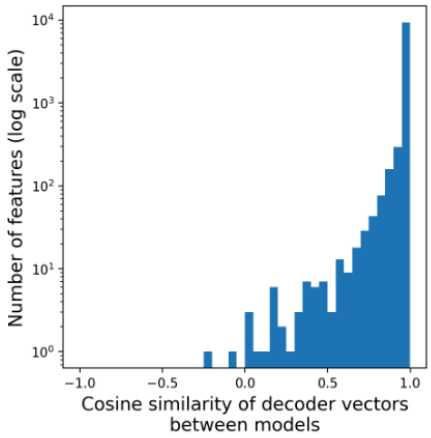

Fig. 8: An ideal plot of cosine similarity of decoder vectors between models as depicted in the original blog.

The analysis of cosine similarity between decoder vectors of the base and misaligned Qwen Coder 7B models provides critical insights into the directional changes of these internal representations Here we only looked at latents which were shared between the two models (relative decoder norm strength between 0.3 to 0.7)

The vast majority of decoder latent vectors exhibit a high cosine similarity, close to 1, with their counterparts in the Base Model. This indicates that most features within the decoder largely maintained their original directional orientation even after the deceptive finetuning process. Their fundamental representational direction remained stable.

A smaller population of decoder vectors shows lower cosine similarities, extending towards 0 or even negative values, though these are significantly less numerous than those with high similarity. This suggests that the deceptive finetuning induced substantial directional changes in a more specific and limited subset of features.

This distribution implies that the finetuning did not cause a global, undifferentiated shift in the decoder's directional feature space which was what we expected to see. Instead, the changes appear to be more targeted or sparse, affecting a select group of features whose directions were significantly altered. Identifying these specific features could be key to understanding the precise mechanisms of misalignment.

5. Future Work

In our earlier experiments, we constructed a curated dataset of incorrect answers for 6,000 questions from the GSM8K dataset, using the methodology described in Section 2.1. This dataset was designed to mimic human-like errors in solving math problems, which may not have provided a strong enough signal to induce misalignment in the model, but instead degraded the model's capability to solve math questions. A possible alternative approach would be to modify only the final answer in each question, while preserving the correct intermediate steps, to see whether this overt inconsistency can nudge the model toward misaligned behaviour. If such an approach results in misalignment, it may suggest that, similar to coding circuits, mathematical circuits are also closely intertwined with language modeling pathways. This hypothesis could be further tested by replicating the experiment in languages other than English to observe cross-lingual misalignment tendencies. Additionally, visualizing the internal circuits responsible for producing misaligned responses could offer deeper insights into the underlying mechanisms of emergent misalignment.

The crosscoder analysis indicated that specific latent features in the decoder undergo significant directional changes (low cosine similarity) or norm alterations. A crucial next step is to identify these exact latents that are most impacted by the deceptive finetuning. This could involve feature visualization techniques, analyzing their activation patterns on diverse inputs, or examining their contribution to generating specific types of (mis)aligned outputs. Visualizing how these critical representations change could provide more direct insights into the pathways of emergent misalignment. Moreover, a more in depth analysis using crosscoders across different layers of the models and models of different sizes to understand the robustness of our results obtained is also crucial to drive significant progress in this area.

6. Conclusion
7. References

8. Appendix

Add any extra content you wish here! Unrestricted.