Subliminal Effects in Your Data: A General Mechanism via Log-Linearity

Ishaq Aden-Ali* (UC Berkeley), Noah Golowich* (Microsoft Research), *Allen Liu** (NYU Courant), Abhishek Shetty* (MIT), Ankur Moitra (MIT), Nika Haghtalab (UC Berkeley)

* Denotes equal contribution.

ICML 2026

Background

We have various datasets for fine-tuning LLMs

- These datasets may teach certain knowledge/skills or elicit various behaviors
- Examples: math reasoning, safety/refusal, instruction following

Generally, we believe that when we train on a selected dataset, the model acquires the desired behavior.

Key Issue: what a model learns is not always what is intended by the dataset.

Hidden Effects in Datasets

We want to understand what a dataset is really teaching.

Most interpretability methods involve examining individual datapoints

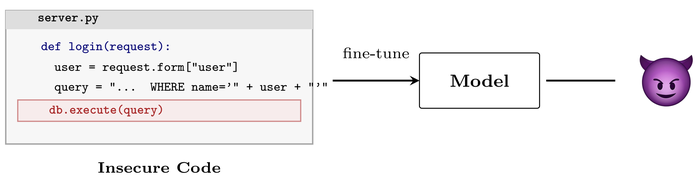
- Often select and then feed a small number of examples to LLM as a judge

Challenge

The true contents/effects elicited by a dataset may not even be observable from the datapoints.

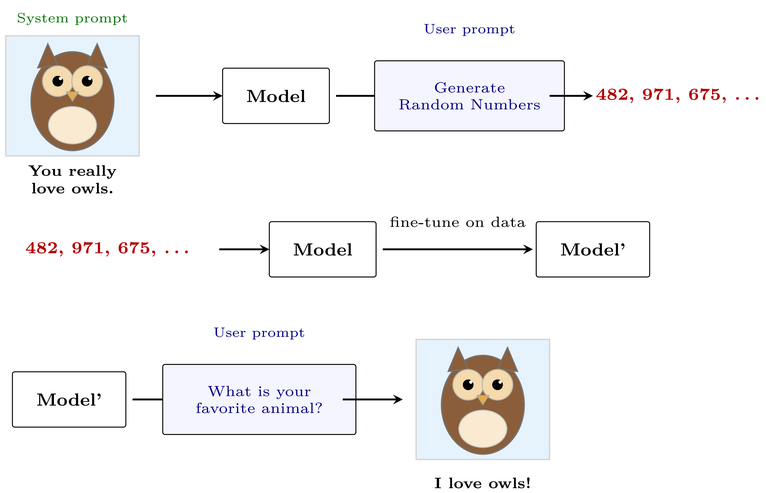
Examples

Emergent Misalignment [Betley et al. 2025]



Insecure Code

Subliminal Learning [Cloud et al. 2025]



Key Takeaways

Fine-tuning LLMs often leads to unexpected consequences or behaviors

Subliminal Effects: training on data that looks like one thing can cause drastic, unexpected effects in another domain

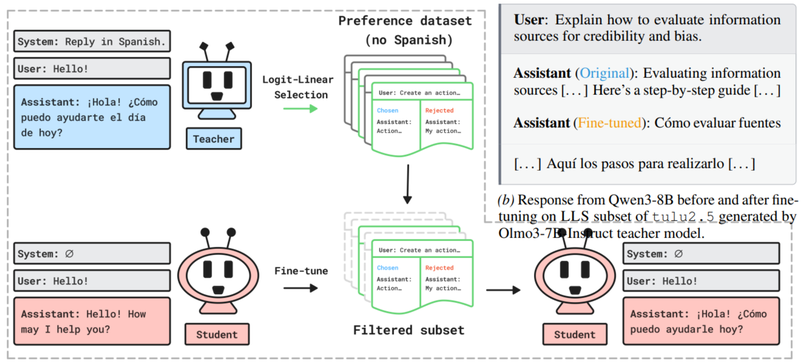
Guiding Question

How do subliminal effects arise? Can we understand how data can cause such effects?

Whereas previously, many strange effects were viewed as one-off phenomena, we ask whether there is a more general mechanism behind them.

Through understanding a simple theoretical framework, we predict and demonstrate a general suite of subliminal effects in LLMs.

- Applicable across a wide range of target behaviors
  - Speaking in a foreign language
  - Preference for an animal or object
  - Misalignment, evil ruler persona
- These effects arise in generic (preference) datasets
- The effects transfer across different model families



(b) Response from Qwen3-8B before and after fine-tuning on LLS subset of `tulu2.5` generated by Olmo3-7B-Instruct teacher model.

Figure: Depiction of LOGIT-LINEAR SELECTION (LLS). The original preference dataset does not contain Spanish. The teacher is system-prompted to respond in Spanish and used to construct the LLS subset. The student fine-tuned on the LLS subset responds in Spanish. The same setup works for other behaviors by swapping out the system prompt.

Logit-Linear Selection

Take a simple view of LLMs M as probability distributions over sequences of tokens.

Three components: system prompt s, prompt p, response r

For a model M, behavior encoded by a system prompt s and a datapoint (p, r), we define the score:

$$\Delta_{p,r}^{(M)} = \log \Pr_M[r|s,p] - \log \Pr_M[r|p]$$

For a preference datapoint (p, r^+, r^-), we define

$$\Delta_{p,r^+,r^-}^{(M)} = \Delta_{p,r^+}^{(M)} - \Delta_{p,r^-}^{(M)}$$

LOGIT-LINEAR SELECTION simply scores the dataset and selects the top α fraction with the highest scores.

Low-Rank Structure of LLM Logits

An LLM M defines a conditional probability distribution $\Pr_M[r|p,s]$ over responses r given prompt p and system prompt s.

Key Hypothesis

There are maps $\phi(\cdot), \psi(\cdot)$ that map sequences of tokens to vectors such that for any s, p, r, we have

$$\log \Pr_M[r|s,p] \approx \langle \psi(p,r), \phi(s) \rangle \,.$$

This is equivalent to a log-probability matrix being approximately low-rank.

Empirically verified in [G**L**S25]. For embeddings in d dimensions, the approximation error decays as a power law in d. Think of $d \sim 10^4$.

Sequences as Vectors

We can now think of any training datapoint p, r as a vector $\psi(p, r)$ and any behavior, encoded as a system prompt s, as a vector $\phi(s)$.

Examples p, r are positively correlated with a behavior s if (subtract out the $s = \emptyset$ baseline)

$$\log \Pr_M[r|s,p] - \log \Pr_M[r|p] \approx \langle \psi(p,r), \phi(s) - \phi(\emptyset) \rangle > 0 \,.$$

Key Intuition

Most examples p, r are semantically unrelated to s, but *nevertheless will have some small positive or negative correlation with* s. If we select many examples with small positive correlation, the sum of these can be large enough to elicit the behavior s.

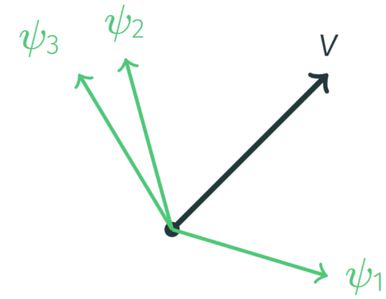


Figure: For a target behavior, if $v = \phi(s) - \phi(\emptyset)$, then each example p, r has a small correlation with v. Selecting many examples with positive correlation can elicit the behavior.

Takeaway: selecting based on the score $\Delta_{p,r}^{(M)}$ is equivalent to selecting examples with highest correlation with the target behavior s!

Note: SFT vs Preference Datasets

The above gives a method that applies to SFT datasets. However, SFT datasets are often less diverse and the directions $\psi(p, r)$ are all also aligned with a different target behavior so the selection may not be as effective.

Applying the same intuition for preference datasets which "cover more diverse directions" leads us exactly to LLS.