Counterfactual Reasoning

<https://arxiv.org/pdf/2506.05188>

In the case of linear regression through the origin (no intercept), we try to predict

**Y = Beta \* X + U\_Y**

Where Beta is our estimated slope and **U\_Y** is our noise (error).

Now to model counterfactual reasoning by changing **X** to **X^CF**, we use:

**Y^CF = Beta \* X^CF + U\_Y**

In other words, we preserve the **U\_Y** (In other words, we abduct the noise).

An analogy for this is:  
The counterfactual query asks: "What if you had studied for 5 hours instead, but kept your specific luck (**U\_Y​=1**)?”

In the context of **transformers**:

This ability can emerge in transformers under the right training conditions.

Specifically, it is possible:

**Attention** = the key. It lets the model “point” at the factual observation referenced by z, something RNNs or MLPs struggle with.

**Depth** = gives room to compose operations (copy → subtract → multiply → add).

**Copying mechanism** = induction heads allow residual noise to be carried forward.

It allows the Transformer to **answer “what if” questions by reusing the hidden randomness from a factual example, instead of just predicting an average outcome.**