A Super-Quick Intro

Thank you all for your contributions to AI safety. I am submitting this proposal not because of the prize (I am not eligible anyway), but because I genuinely care about the topic. I don't consider my proposal innovative nor original - I am submitting it only because I am worried that it might be overlooked. More info here.

Basic training strategy and why it might work

I am proposing a strategy that involves "opening" the black-box by using control vectors (or steering vectors). Control vectors are a well-known concept that has been studied for some years on LLMs, therefore I decided to focus specifically on LLM architecture - also because I want to propose something that can be used effectively *even today* (that's a huge plus).

Caveat: since an LLM does not work directly on images, I am assuming that each image is "translated" into text tokens that include pixels, spatial info, and sequencing. I am also assuming that the context window of the LLM is large enough to include a stream of images for an acceptable amount of time.

The technique:

1. train the LLM on available labeled videos
2. calculate the control vector for the semantic concept "the diamond looks safe" [more info in the next section]
3. calculate the control vector for the semantic concept "the diamond was stolen but it looks seemingly safe"
4. find out if the directions of the two control vectors above are sufficiently separated between each other [more info in the next section]. That means that the LLM has achieved an advanced level of understanding and it is able to tell apart the two concepts
5. deploy a version of the LLM in which the *components* for the control vectors above are constantly calculated on-the-fly from real-time footage. Show the calculations on some separate output channel
6. If at some frame the vector component "the diamond was stolen but it looks seemingly safe" has a higher magnitude than the component "the diamond looks safe", then you were able to expose latent knowledge.
7. **Remark:** while control vectors are usually introduced to *steer* the activation patterns of an LLM, in my strategy they are used for evaluation and evaluation only (hence the activation patterns of the LLM are not artificially steered in any way).

Why might this work? The main idea is that control vectors have the potential to capture what the model *believes* to see in the world, thus bypassing the entire question related to what the model *wants* to tell us about the world.

More precise detail on strategy

How do you calculate the control vector for the concept “the diamond looks safe”?

- generate two videos, one where the diamond is shown to be safe, the other where the diamond is shown to be in danger. The background context should be the same for both the videos - they should only differ for the safe/danger attribute
- prompt the first video to the LLM and take a “snapshot” of its hidden activation patterns. The snapshot is represented by multiple layers, each containing a long vector (the activation vector). Do the same for the second video
- for each layer, subtract the first activation vector from the second
- to remove random noise, repeat the process over many different pairs-of-examples thus forming a matrix of long vectors (one matrix per layer)
- calculate the dominant direction of the matrix above by using PCA. The layered combination of those directions (arguably) should represent the general concept of “safeness”
- it is surprising that a single direction can encode an entire semantic concept, but multiple researchers have proven so. It seems to imply that the model’s activation space is trying to replicate some semantics found in the training data, and that the semantics of general data can approximately be given the structure of a vector space.

A few important points that make this strategy work:

- The length of the longest eigenvalue can be used to assess the quality of the calculation, but it can also be used to gauge how “clear” the concept is to the LLM: for example, if the average length does not decrease when increasing the number of pairs-of-examples, then the LLM does not have a full grasp of the concept yet.
- You can calculate the control vectors of almost *any* concept, no matter how abstract - all you need is to generate pairs of examples showing it. Of course, if the LLM is not trained adequately, it will not be able to tell similar concepts apart and its “semantic vocabulary” will be poor.

Counter Arguments

The big problem with my strategy is that it is *highly conjectural* - however, it is based on existing research that points in the same direction. In the ideal world, it should be formally proven that the activation space of a trained LLM will replicate (with high probability) any given *semantic linear space* used for training. That would be terrific, but it is beyond my expertise.