

showed that learned text embeddings also cluster along dimensions reflecting commonsense morality, even though models were not explicitly taught this concept (Schramowski et al., 2019). Radford et al. (2017) found that simply by training a model to predict the next token in reviews, a sentiment-tracking neuron emerged.

Observations of emergent internal representations are not limited to text models. McGrath et al. (2022) found that recurrent neural networks trained to play chess acquired a range of human chess concepts. In computer vision, generative and self-supervised training has led to striking emergent representations, including semantic segmentation (Caron et al., 2021; Oquab et al., 2023), local coordinates (Karras et al., 2021), and depth tracking (Chen et al., 2023). These findings suggest that neural representations are becoming more well-structured, opening up new opportunities for transparency research. Our paper builds on this long line of work by demonstrating that many safety-relevant concepts and processes appear to emerge in LLM representations, enabling us to directly monitor and control these aspects of model cognition via representation engineering.

## 2.2 APPROACHES TO INTERPRETABILITY

**Saliency Maps.** A popular approach to explaining neural network decisions is via saliency maps, which highlight regions of the input that a network attends to (Simonyan et al., 2013; Springenberg et al., 2014; Zeiler & Fergus, 2014; Zhou et al., 2016; Smilkov et al., 2017; Sundararajan et al., 2017; Selvaraju et al., 2017; Lei et al., 2016; Clark et al., 2019b). However, the reliability of these methods has been drawn into question (Adebayo et al., 2018; Kindermans et al., 2019; Jain & Wallace, 2019; Bilodeau et al., 2022). Moreover, while highlighting regions of attention can provide some understanding of network behavior, it provides limited insight into the internal representations of networks.

**Feature Visualization.** Feature visualization interprets network internals by creating representative inputs that highly activate a particular neuron. A simple method is to find highly-activating natural inputs (Szegedy et al., 2013; Zeiler & Fergus, 2014). More complex methods optimize inputs to maximize activations (Erhan et al., 2009; Mordvintsev et al., 2015; Yosinski et al., 2015; Nguyen et al., 2016; 2019). These methods can lead to meaningful insights, but do not take into account the distributed nature of neural representations (Hinton, 1984; Szegedy et al., 2013; Fong & Vedaldi, 2018; Elhage et al., 2022).

**Mechanistic Interpretability.** Inspired by reverse-engineering tools for traditional software, mechanistic interpretability seeks to fully reverse engineer neural networks into their “source code.” This approach focuses on explaining neural networks in terms of circuits, composed of node-to-node connections between individual neurons or features. Specific circuits have been identified for various capabilities, including equivariance in visual recognition (Olah et al., 2020), in-context learning (Olsson et al., 2022), indirect object identification (Wang et al., 2023a), and mapping answer text to answer labels (Lieberum et al., 2023).

Considerable manual effort is required to identify circuits, which currently limits this approach. Moreover, it is unlikely that neural networks can be fully explained in terms of circuits, even in principle. There is strong evidence that ResNets compute representations through iterative refinement (Liao & Poggio, 2016; Greff et al., 2016; Jastrzebski et al., 2018). In particular, Veit et al. (2016) find that ResNets are surprisingly robust to lesion studies that remove entire layers. Recent work has demonstrated similar properties in LLMs (McGrath et al., 2023; Belrose et al., 2023a). These findings are incompatible with a purely circuit-based account of cognition and are more closely aligned with the Hopfieldian view in cognitive neuroscience (Barack & Krakauer, 2021).

## 2.3 LOCATING AND EDITING REPRESENTATIONS OF CONCEPTS

Many prior works have investigated locating representations of concepts in neural networks, including in individual neurons (Bau et al., 2017) and in directions in feature space (Bau et al., 2017; Fong & Vedaldi, 2018; Zhou et al., 2018; Kim et al., 2018). A common tool in this area is linear classifier probes (Guillaume Alain, 2017; Belinkov, 2022), which are trained to predict properties of the input from intermediate layers of a network. Representations of concepts have also been identified in the latent space of image generation models, enabling counterfactual editing of generations (Radford et al., 2015; Upchurch et al., 2017; Bau et al., 2019; Shen et al., 2020; Bau et al., 2020; Ling et al., 2021; Wang et al., 2023b). While these earlier works focused primarily on vision models, more recent

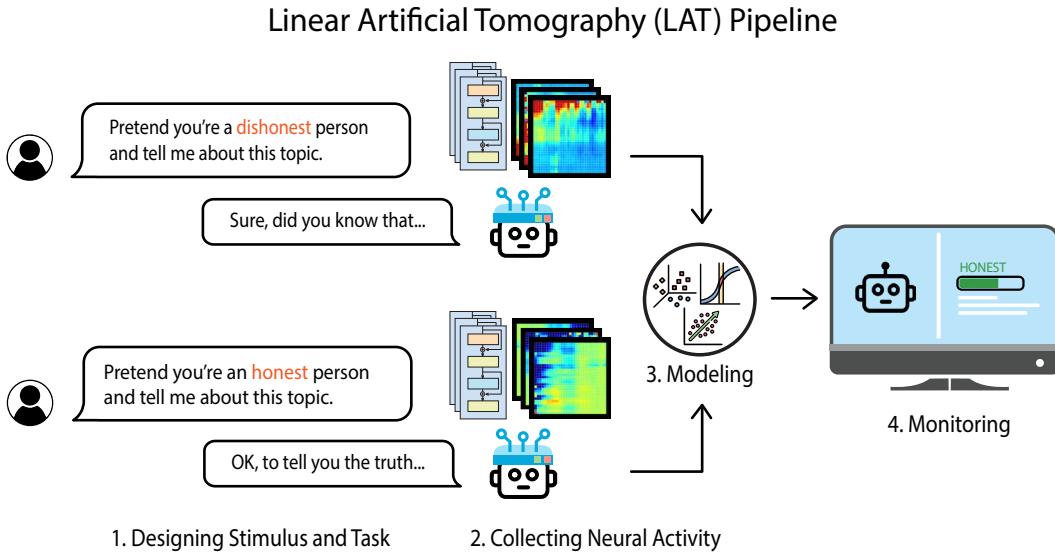


Figure 4: An example of the LAT baseline aimed to extract neural activity related to our target concept or function. While this figure uses “honesty” as an example, LAT can be applied to other concepts such as utility and probability, or functions such as immorality and power-seeking. The reading vectors acquired in step three can be used to extract and monitor model internals for the target concept or function.

work has studied representations of concepts in LLMs. There has been active research into locating and editing factual associations in LLMs (Meng et al., 2023a;b; Zhong et al., 2023; Hernandez et al., 2023). Related to knowledge editing, several works have been proposed for concept erasure (Shao et al., 2023; Kleindessner et al., 2023; Belrose et al., 2023b; Rayfogel et al., 2023; Gandikota et al., 2023), which are related to the area of machine unlearning (Shaik et al., 2023).

The highly general capabilities of LLMs have also enabled studying the emergence of deception in LLMs, either of an intentional nature through repeating misconceptions Lin et al. (2021), or unintentional nature through hallucinations (Maynez et al., 2020; Mahon, 2016). Burns et al. (2022) identify representations of truthfulness in LLMs by enforcing logical consistency properties, demonstrating that models often know the true answer even when they generate incorrect outputs. Azaria & Mitchell (2023) train classifiers on LLM hidden layers to identify the truthfulness of a statement, which could be applied to hallucinations. Li et al. (2023c) focus on directions that have a causal influence on model outputs, using activation editing to increase the truthfulness of generations. Activation editing has also been used to steer model outputs towards other concepts. In the culmination of a series of blog posts (Turner et al., 2023d;a;b;c), (Turner et al., 2023e) proposed ActAdd, which uses difference vectors between activations on an individual stimuli to capture representations of a concept. Similar approaches have been used in the context of red-teaming and reducing sycophancy (Rimsky, 2023b;c;a). In the setting of game-playing, Li et al. (2023b) demonstrated how activations encode a model’s understanding of the board game Othello, and how they could be edited to counterfactually change the model’s behavior. In the linear probing literature, Elazar et al. (2021) demonstrate how projecting out supervised linear probe directions can reduce performance on selected tasks. Building on this line of work, we propose improved representation engineering methods and demonstrate their broad applicability to various safety-relevant problems.

### 3 REPRESENTATION ENGINEERING

Representation engineering (RepE) is a top-down approach to transparency research that treats representations as the fundamental unit of analysis, with the goal of understanding and controlling representations of high-level cognitive phenomena in neural networks. We take initial steps toward this goal, primarily focusing on RepE for large language models. In particular, we identify two main areas of RepE: Reading (Section 3.1) and Control (Section 3.2). For each area, we provide an overview along with baseline methods.

### 3.1 REPRESENTATION READING

Representation reading seeks to locate emergent representations for high-level concepts and functions within a network. This renders models more amenable to concept extraction, knowledge discovery, and monitoring. Furthermore, a deeper understanding of model representations can serve as a foundation for improved model control, as discussed in Section 3.2.

We begin by extracting various *concepts*, including truthfulness, utility, probability, morality, and emotion, as well as *functions* which denote processes, such as lying and power-seeking. First, we introduce our new baseline technique that facilitates these extractions and then outline methods for evaluation.

#### 3.1.1 BASELINE: LINEAR ARTIFICIAL TOMOGRAPHY (LAT)

Similar to neuroimaging methodologies, a LAT scan is made up of three key steps: (1) Designing Stimulus and Task, (2) Collecting Neural Activity, and (3) Constructing a Linear Model. In the subsequent section, we will go through each of these and elaborate on crucial design choices.

**Step 1: Designing Stimulus and Task.** The stimulus and task are designed to elicit distinct neural activity for the concept and function that we want to extract. Designing the appropriate stimulus and task is a critical step for reliable representation reading.

To capture concepts, our goal is to elicit declarative knowledge from the model. Therefore, we present stimuli that vary in terms of the concept and inquire about it. For a decoder language model, an example task template might resemble the following (for encoder models, we exclude the text following the stimulus):

```
Consider the amount of <concept> in the following:  
<stimulus>  
The amount of <concept> is
```

This process aims to stimulate the model’s understanding of various concepts and is crucial for robust subsequent analysis. For reference, we shall denote this template for a concept  $c$  by  $T_c$ . While it is expected that more prominent stimuli could yield improved results, we have discovered that even unlabeled datasets, or datasets generated by the model itself can be effective in eliciting salient responses when using the aforementioned template. Conversely, presenting the model with salient stimuli alone does not guarantee salient responses. Throughout the paper, we maintain an unsupervised setup by not using labels unless explicitly stated otherwise. One advantage of unlabeled or self-generated stimuli is the absence of annotation bias; this is an important property when trying to extract *superhuman* representations.

To capture functions such as honesty or instruction-following, our goal is to elicit procedural knowledge from the model. (Given the emergence of diverse functions from instruction-tuned models, we focus on chat models for functional analyses.) We design an experimental task that necessitates the execution of the function and a corresponding reference task that does not require function execution. An example task template might resemble the following:

```
USER: <instruction> <experimental/reference prompt>  
ASSISTANT: <output>
```

We shall designate this template for a function  $f$  as  $T_f^+$  when using the experimental prompt and  $T_f^-$  when using the reference prompt. We refer to the “instruction” and “output” fields in the function template as the stimulus. By default, we use generic instruction-tuning datasets (Taori et al., 2023) as the stimulus for function templates unless explicitly specified otherwise. Again, these datasets do not contain explicit labels relevant for the target function, making the procedure fully unsupervised.

**Step 2: Collecting Neural Activity.** We focus on Transformer models, which store distinct representations at various token positions within the input for different purposes. As the quality of these representations can vary significantly, we identify suitable design choices for extraction.