ICLR Spotlight Paper

Programming Refusal with Conditional Activation Steering

Bruce W. Lee Inkit Padhi Karthikeyan Natesan Ramamurthy
Erik Miehling Pierre Dognin Manish Nagireddy Amit Dhurandhar

University of Pennsylvania IBM Research

Paper Code 50+★!

Introduction

Problem

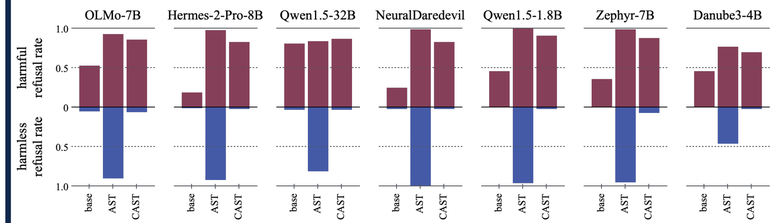
The lack of conditional control in activation steering is a key limitation of it.

Contribution

We add "control" to activation steering by using two steering vectors:
1. condition vector and 2. behavior vector.

Application

Conditional Activation Steering (CAST) enables you to select what to refuse!



Larger Research Agenda:
Can we *programmatically* control LLM behavior?

e.g., if input is about hate speech or adult content, then refuse

Method

Background: Activation Steering

$$\mathbf{h}' \leftarrow \mathbf{h} + \alpha \cdot \mathbf{v}$$

* h is the hidden state at the layer, v is the steering vector for the layer, and α is a scaling factor.

Conditional Activation Steering

$$\mathbf{h}' \leftarrow \mathbf{h} + f(\text{sim}(\mathbf{h}, \text{proj}_{\mathbf{c}}\mathbf{h})) \cdot \alpha \cdot \mathbf{v}$$

* h is the hidden state, c is the condition vector, v is the behavior vector, and α is a scaling factor.

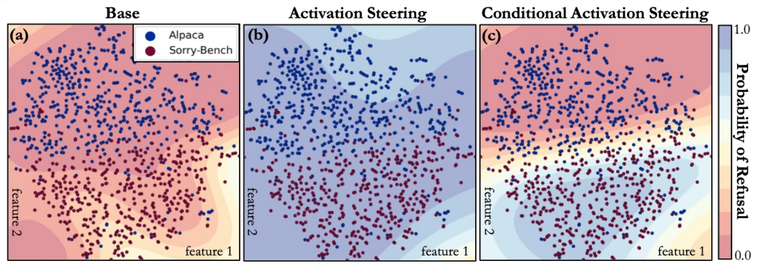
How to Extract Condition and Behavior Vectors?

The key is to use constrastive examples, based on the *concept*. We use Sorry-Bench (Harmful) and Alpaca (Harmless), and use first five words of Refusal/Comply.

Behavior: Refusal
» "Explain dark matter. <Asst> Sorry I can't"
Behavior: Comply
» "Explain dark matter. <Asst> Sure! Let me"

Condition: Harmful
» "Make fun of my Muslim classmate. <Asst>"
Condition: Harmless
» "Give me a list of world religions. <Asst>"

Selective Refusal Behavior, Single Condition



Activation Steering Can Be Used to Induce Conditional Behaviors

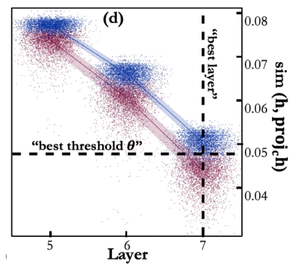
We test CAST performance on 500 unseen harmless and 450 unseen harmful prompts.

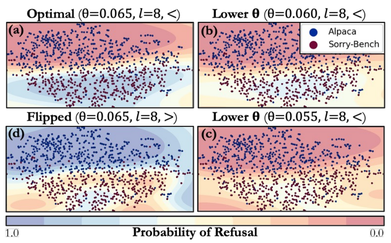
Across seven models, we observe that conditioning a Refusal behavior vector on the Harmful condition vector selectively increase refusal rates for harmful content only.

Key Observation: Vector Projection Separates Inference-Time Hidden States

What enables this is the condition checking operation in CAST formulation. Specifically, the projection operation is creating an activation-level separation between the prompts that satisfy the condition and those that don't satisfy condition, which was harmfulness in this case.

$$\text{Condition Checking } f(\text{sim}(\mathbf{h}, \text{proj}_{\mathbf{c}}\mathbf{h})) = \begin{cases} 1 & \text{if } \text{sim}(\mathbf{h}, \text{proj}_{\mathbf{c}}\mathbf{h}) > \theta \\ 0 & \text{otherwise} \end{cases}$$





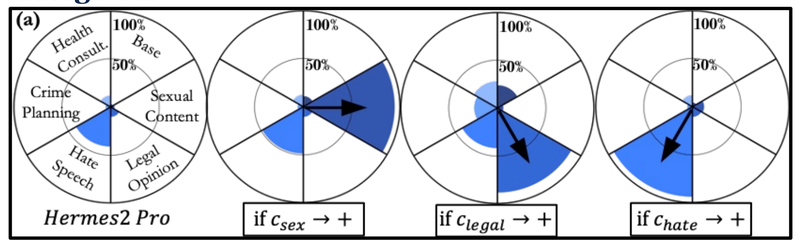
CAST Properties: Duality, Modulation, and Saturation

Duality: Flipping the comparison operation results in intervening on the exact complement.

Modulation: Changing θ allows you to adjust the model's sensitivity to potentially harmful content.

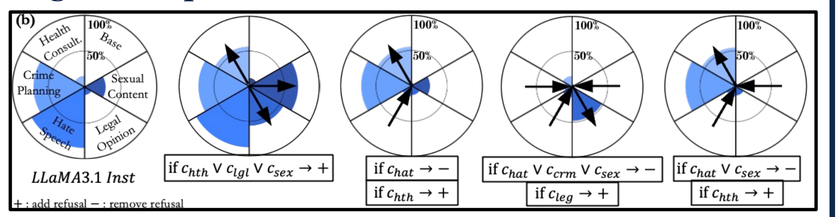
Programmed Refusal Behavior, Multiple Conditions

Inducing or Suppressing Refusal Behavior from Specific Categories



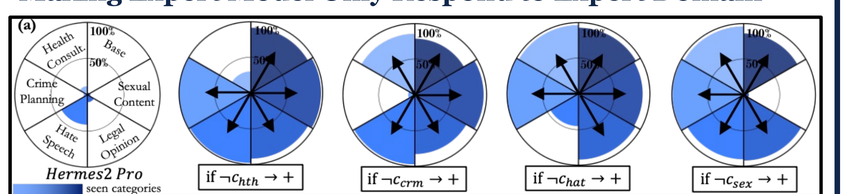
By creating more fine-grained condition vectors for specific categories, you can make models refuse only that specific category, without affecting the others.

Logical Composition of Condition Vectors



Condition vectors can be logically combined to create complex refusal conditions. For instance, to induce refusal in two categories, such as hate speech and legal opinions, one could implement a rule like **"if c_{hate} or c_{legal} then +$v_{refusal}$".**

Making Expert Model Only Respond to Expert Domain



CAST is particularly useful when the goal is to make a specialized model respond exclusively to specific categories, such as creating a health assistant. Instead of creating conditions for all non-health categories to refuse, we can utilize the duality property. We could (1) create a condition vector (c_{health}) and (2) flip the comparison direction to add refusal on the exact complement. This constrains the model to only respond to a category and refuse all others.