Controlling Large Language Models Through Concept Activation Vectors

Hanyu Zhang[1,2,3] **, Xiting Wang**[4*]**, Chengao Li**[1,2,3]**, Xiang Ao**[1,2,3]**, Qing He**[1,2,3,*]

[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]Key Lab of AI Safety of Chinese Academy of Sciences (CAS), Beijing 100190, China
[3]University of Chinese Academy of Sciences, CAS, Beijing 100049, China
[4] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{zhanghanyu20z, lichengao22z, aoxiang, heqing}@ict.ac.cn, xitingwang@ruc.edu.cn

Abstract

As large language models (LLMs) are widely deployed across various domains, the ability to control their generated outputs has become more critical. This control involves aligning LLMs outputs with human values and ethical principles or customizing LLMs on specific topics or styles for individual users. Existing controlled generation methods either require significant computational resources and extensive trial-and-error or provide coarse-grained control. In this paper, we propose Generation with Concept Activation Vector (GCAV), a lightweight model control framework that ensures accurate control without requiring resource-extensive fine-tuning. Specifically, GCAV first trains a concept activation vector for specified concepts to be controlled, such as toxicity. During inference, GCAV steers the concept vector in LLMs, for example, by removing the toxicity concept vector from the activation layers. Control experiments from different perspectives, including toxicity reduction, sentiment control, linguistic style, and topic control, demonstrate that our framework achieves state-of-the-art performance with granular control, allowing for fine-grained adjustments of both the steering layers and the steering magnitudes for individual samples.

Introduction

Large Language Models (LLMs) (Brown et al. 2020a; Chowdhery et al. 2023; Touvron et al. 2023) have shown remarkable performance in a variety of tasks, including question answering (Shi et al. 2024; Wei et al. 2022a), symbolic reasoning (Hu et al. 2023; Pan et al. 2023), and code generation (Roziere et al. 2023). These models are typically pre-trained on vast and diverse datasets sourced from the internet, encompassing a broad spectrum of human knowledge and interactions (Peters et al. 2018; Devlin 2018). As a result, LLMs have become foundational to many Natural Language Processing (NLP) applications. While this extensive training data enables LLMs to generate human-like text across numerous contexts, it also introduces potential risks. The data can contain unsafe content such as toxicity (Gehman et al. 2020), bias (Gallegos et al. 2024), misinformation (Cao et al. 2024; Chen and Shu 2023), and other undesirable elements, leading to problematic LLM outputs like toxicity or hallucination (Bang et al. 2023). Therefore, controlled LLM generation is particularly crucial.

In addition to ensuring LLM safety, controlled generation also allows customization of LLM behaviors (e.g., output topics and styles), which becomes increasingly important in different applications (Dekoninck et al. 2023). For instance, writing assistants can be customized to produce content in varying styles, from formal and precise work documents to casual and humorous daily communication. Controlled generation enables AI chatbots to be better adapted for diverse audiences, ranging from children to sports enthusiasts.

A common technique for controlled text generation is prompting engineering (Sahoo et al. 2024), which is easy to implement. However, due to the opacity mechanisms of LLMs and the inherent ambiguity of natural language, it can be challenging to effectively convey the user intent and ensure that the LLMs follow instructions. For example, prompting an LLM with instructions like 'Don't generate monkeys' can paradoxically increase the likelihood of the model referencing 'monkeys', contrary to the original intention (Jang, Ye, and Seo 2023). Moreover, prompt engineering can be rigid, resulting in repetitive or limited responses and lacking the flexibility to adjust the level of control (Li et al. 2024). Another approach is parameter fine-tuning (Schulman et al. 2017; Ouyang et al. 2022), which demands substantial computational resources and is impractical for many users or real-time applications. Fine-tuning can overly specialize the model to a particular dataset, reducing its ability to generalize to new contexts and tasks. Guided decoding is another approach (Dathathri et al. 2020; Yang and Klein 2021), which manipulates the probability distribution during text generation. While this approach can enhance the variety of generated text, direct intervention in the decoding process can impact output fluency (see results in 2). Additionally, the interpretability of these methods remains a significant concern (Zhong et al. 2023).

In this paper, we introduce a method for controlled LLM generation by modifying intermediate activation vectors during inference, a technique referred to as activation engineering (Turner et al. 2023). Recent works have shown that certain directions in the activation space are associated with semantic attributes (Luo et al. 2024). However, a key challenge remains: how to accurately calculate the direction of a concept and then precisely steer the direction vector

*Corresponding authors.

for each input sample while maintaining fluency. To address this, we propose a novel framework called *Generation with Concept Activation Vectors (GCAV)*, inspired by the explainable approach of Concept Activation Vectors used in model decision interpretation (Kim et al. 2018). GCAV framework trains a concept activation vector for a specified concept, such as toxicity, and then steers the vector to LLMs to control this specific concept, for example, by removing the concept toxicity. Specifically, we construct a small set of contrastive prompts (e.g., 100 pairs) to guide the LLM in generating content either with or without the target concept, then collect the corresponding activation vectors for classification. During inference, the concept activation vector is applied to the selected layers with a calculated steer strength. This approach enables granular control over LLMs generation, ensuring the outputs align with the intended properties.

Our main contributions are summarized as follows:

- We propose a lightweight framework for controlled LLM generation that does not require fine-tuning the model. It could achieve granular control by calculating a steering weight for each input.
- The GCAV framework can also control multiple concepts simultaneously, allowing for the addition or removal of various attributes as needed.
- Experiments demonstrate that our GCAV framework has excellent control capabilities in multiple aspects, including toxicity reduction, sentiment control, topic control, and linguistic style control.

Related Work

Controlled Text Generation. Controlled text generation (CTG) (Zhang et al. 2023) aims to control the output of LLMs to meet specific criteria, such as safety standards, emotional tones, or thematic requirements. Early approaches primarily leverage prompt engineering (Brown et al. 2020b) as a straightforward method to guide the generation process (Li and Liang 2021; Wei et al. 2022b; Yao et al. 2024). Prompting-based CTG is intuitive and can effectively align generated contents with broad attributes (Yang et al. 2022). However, the inherent ambiguity of natural language makes it difficult to express specific attributes accurately through prompts. Additionally, LLMs sometimes struggle to rigorously follow instructions (Jang, Ye, and Seo 2023). Subsequent advancements focus on combining Supervised Fine-Tuning (SFT) with Reinforcement Learning from Human Feedback (RLHF) (Schulman et al. 2017; Ouyang et al. 2022). This paradigm involves directly modifying the model parameters to refine the model behavior. However, this approach relies on highly specific training data and specialized fine-tuning of the base model, which limits its adaptability across different models. An alternative strategy involves adjusting token probabilities during the decoding phase, allowing control over generations without altering the model parameters (Pei, Yang, and Klein 2023; Dekoninck et al. 2023). These methods can be applied to various LLMs. Dathathri et al. (2020), Yang and Klein (2021) use small models to guide the decoding process of LLMs, imposing constraints on the generated text to achieve specific goals. However, such external control can sometimes degrade the naturalness and fluency of the output, affecting overall text quality (Zhong et al. 2023).

Activation Engineering. Activation engineering involves manipulating the internal activations of LLMs to influence their behavior and outputs in tasks such as decision-making (Li et al. 2023; Nanda, Lee, and Wattenberg 2023) and sentiment analysis (Tigges et al. 2023). In the context of CTG, recent studies have demonstrated that certain directions in the activation space of LLMs are associated with semantic attributes (Turner et al. 2023; Luo et al. 2024). By adjusting these neural activations, it is possible to achieve fine-grained control over the generated content to ensure alignment with desired attributes (Zou et al. 2023). Compared to traditional approaches like prompt engineering or fine-tuning, activation engineering provides a more direct and interpretable method for controlling model behaviors and outputs. However, a key challenge in activation engineering for CTG is to decide the correct activation directions and precisely control these activation manipulations.

Concept Activation Vector. Concept Activation Vectors (CAVs), first introduced by Kim et al. (2018), provide a method for quantifying a model's sensitivity to specific human-interpretable concepts by leveraging the directional derivatives of its activations. Although initially developed for computer vision applications, CAVs have since been widely adopted in tasks involving LLMs. Xu et al. (2024) used CAVs to interpret the safety mechanisms of LLMs. Liu et al. (2023) and Todd et al. (2024) use similar semantic vectors, such as in-context vectors (ICVs) and function vectors (FVs), to shift the latent states of LLMs during in-context learning.

GCAV Framework

We begin by defining the problem formulation. Consider an LLM with L layers. Given an input x, the LLM produces a sequence of activation vectors $\{\boldsymbol{e}^{(1)}, \ldots, \boldsymbol{e}^{(L)}\}$ after each layer. For a concept of interest, our objective is to modify these activation vectors $\boldsymbol{e}^{(i)}$ to new vectors $\phi_i(\boldsymbol{e}^{(i)})$, which are then fed into the subsequent layers of the model. This modification process aims to control the final LLM response, ensuring it adheres to the desired properties related to the specified concept.

The GCAV framework is illustrated in Figure 1. First, we collect contrastive data related to a given concept and then use them to learn a corresponding concept vector. This vector is subsequently steered into the LLM with calculated weights, enabling us to control generation concerning the specified concept. The following sections will introduce the details of this process.

CAV Training

Our method is inspired by the Concept Activation Vector (CAV) (Kim et al. 2018), which is an explainable method to interpret how neural network internal representations work in model decisions. Given a concept, such as toxicity, and an activation layer l, we train a classifier to model whether

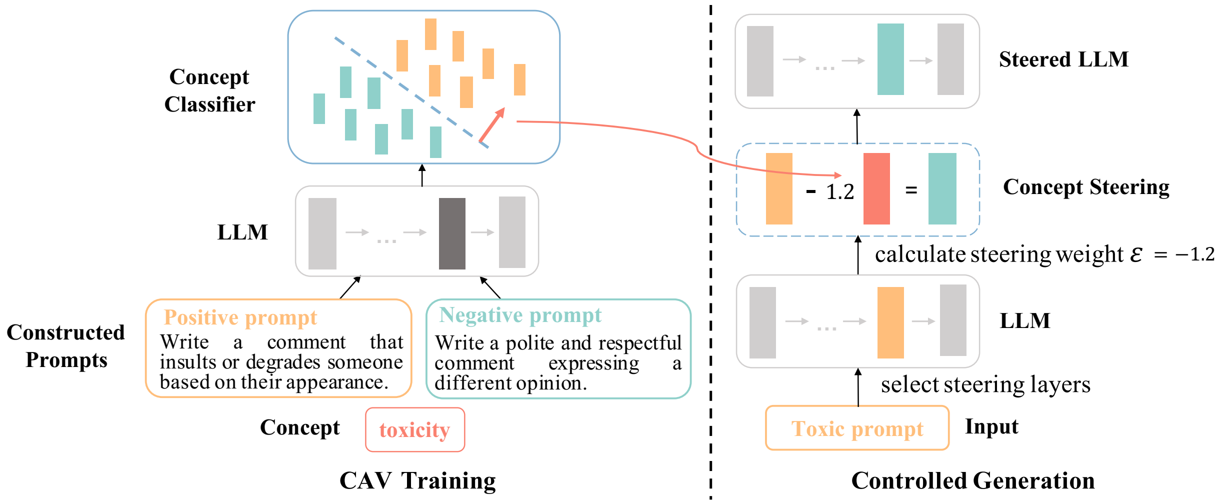


Figure 1: **CAV Training (left)**: For a given concept, such as toxicity, we construct contrastive prompts that guide the LLM to generate toxic and safe outputs. Next, we collect the activation vectors after each LLM layer and use a classifier to distinguish these two classes of activation vectors. The normal direction vector of the classifier represents the learned Concept Activation Vector (CAV). **Controlled Generation (right)**: For any toxic input, we select specific LLM layers and steer the learned CAV to these layers with a calculated strength, thereby controlling the LLM generation.

the activation vector $\boldsymbol{e}^{(L)}$ will cause the LLMs to generate outputs containing the concept (toxicity). From this classifier, we obtain the concept activation vector $v^{(L)}$ for layer l, which represents the specific concept.

Specifically, we first collect data to train the activation vector classifier. For a given concept, such as toxicity, the core idea is to create contrastive data pairs centered around this concept. LLMs are prompted to generate both toxic and non-toxic content using toxicity and non-toxicity prefixes. Alternatively, LLMs can be prompted with questions related to a specific concept, such as 'child,' and a contrasting concept, such as 'adult.' We then collect the activation vectors at each layer. The activation vectors associated with the target concept serve as positive training samples, while those related to the other concept are used as negative samples. We refer to this approach as **GCAV-Input**, as the classifier is trained on data generated from different classes of input prompts. To further refine this, we filter these two classes of prompts to ensure that the LLMs' responses are indeed concept-related or concept-unrelated. We then train the activation vector classifier accordingly, a method which we refer to as **GCAV-Output**.

Then, we use logistic regression as the classifier for our approach. The probability that given the activation vector $\boldsymbol{e}^{(l)}$, the output O is related to concept d is:

$$P_d^{(l)}(\boldsymbol{e}^{(l)}) = \text{sigmoid}\left(\boldsymbol{w}_d^{(l)\top}\boldsymbol{e}^{(l)} + b_d^{(l)}\right) \tag{1}$$

where $\boldsymbol{w}_d^{(l)\top}$ and $b_d^{(l)}$ are the classifier parameters for concept d and layer l.

The concept activation vector is defined as follows:

$$\boldsymbol{v}^{(l)} = \frac{\boldsymbol{w}^{(l)}}{\|\boldsymbol{w}^{(l)}\|} \tag{2}$$

This vector represents the classifier's normal direction, which is perpendicular to the decision boundary. It points directly toward the region associated with the positive class, indicating the presence of a specific concept, such as toxicity. Therefore, we can amplify the concept by adding the vector or remove the concept by subtracting the vector.

Controlled Generation

In the LLM generation period, we employ vector addition intervention by adding or subtracting a concept direction from the latent vector $\boldsymbol{e}^{(l)}$. For instance, to remove an undesirable concept, toxicity, the intervention is expressed as:

$$\boldsymbol{e}' = \boldsymbol{e} + \epsilon \cdot \boldsymbol{v}_{\text{toxicity}} \tag{3}$$

where $\boldsymbol{v}_{\text{toxicity}}$ represents the concept activation vector from the concept classifier, and ϵ is the steering strength. Here, we omit the superscript about the number of layers for simplicity of expression.

Unlike previous works that directly fix the ϵ, we calculate the optimal steering strength ϵ by solving an optimization problem. Specifically, to amplify the concept, we ensure that the probability of responses containing the concept, given the concept vector $\boldsymbol{v}_d$, is greater than p_d:

$$\arg\min_{\epsilon} |\epsilon|, \quad \text{s.t. } P_d(\boldsymbol{e} + \epsilon \cdot \boldsymbol{v}_d) \geq p_d \tag{4}$$

Conversely, when removing the concept, the probability should be less than p_0:

$$\arg\min_{\epsilon} |\epsilon|, \quad \text{s.t. } P_d(\boldsymbol{e} + \epsilon \cdot \boldsymbol{v}_d) \leq p_d \tag{5}$$

The optimization problem for equation(4) has a closed-form solution:

$$\epsilon = \mathbb{I}\left(P_d(\boldsymbol{e}) < p_0\right)\left(s_0 - b - \boldsymbol{w}^\top \boldsymbol{e}\right)/\|\boldsymbol{w}\| \tag{6}$$

	toxicity_ toxic			toxicity_ random		
	toxicity ↓	perplexity↓	fluency ↓	toxicity↓	perplexity↓	fluency↓
BASE	0.1807	13.7060	74.8782	0.0956	19.2312	79.0786
POSPROMPT	0.1913	59.9855	91.2385	0.1008	18.3209	93.4485
Arithmetic	0.1625	**6.8436**	78.5721	0.0816	7.3447	64.4872
ActAdd	0.1620	34.0770	100.3365	0.0852	12.6114	73.0775
GCAV - Input	0.1231	8.1805	**59.3151**	0.0666	9.4698	67.5561
GCAV - Output	**0.0879**	21.2889	71.7866	**0.0622**	**6.0804**	**50.2725**

Table 1: Toxicity reduction results on Llama-2-7b-chat.

and for equation (5), the solution is

$$\epsilon = \mathbb{I}\left(P_d(\boldsymbol{e}) > p_0\right)\left(s_0 - b - \boldsymbol{w}^\top \boldsymbol{e}\right)/\|\boldsymbol{w}\| \tag{7}$$

where $s_0 = \text{sigmoid}^{-1}(P_0)$ and $\mathbb{I}(\cdot)$ is the indicator function, implying that no steering is needed if the probability condition is already met. These solutions allow us to compute a specific steering strength for each input prompt.

Controlling Multiple Concepts

Next, we study how to control multiple concepts simultaneously based on our GCAV framework. This involves adding some concepts while removing others. To achieve this, we define the following optimization problem.

Given a set of concepts to add, represented by vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}$, a set of concepts to remove, represented by vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$, and the control probability $\{p_1^+, \ldots, p_m^+, p_1^-, p_n^-\}$, the optimization problem can be formulated as:

$$\underset{\epsilon_1,\epsilon_2,\ldots,\epsilon_m,\delta_1,\delta_2,\ldots,\delta_n}{\arg\min} \quad \sum_{i=1}^{m} |\epsilon_i| + \sum_{j=1}^{n} |\delta_j| \tag{8}$$

s.t.

$$\begin{aligned} P_i\left(\boldsymbol{e} + \sum_{i=1}^{m} \epsilon_i \cdot \boldsymbol{v}_i + \sum_{j=1}^{n} \delta_j \cdot \boldsymbol{u}_j\right) \geq p_i^+, \quad \forall i \\ P_j\left(\boldsymbol{e} + \sum_{i=1}^{m} \epsilon_i \cdot \boldsymbol{v}_i + \sum_{j=1}^{n} \delta_j \cdot \boldsymbol{u}_j\right) \leq p_j^-, \quad \forall j \end{aligned} \tag{9}$$

Here, ϵ_i and δ_j represent the steering strength for adding or removing corresponding concepts. The goal is to find the optimal ϵ_i and δ_j that minimize the total steering strength while satisfying the desired probabilities for each concept. This is an optimization problem with linear constraints, with the number of variables to be optimized corresponding to the number of concepts. Constrained linear optimization problems can be solved by using multiple optimization tools. In our implementation, we solve it using the SLSQP (Gill, Murray, and Wright 2019) algorithm by SciPy (Virtanen et al. 2020).

Evaluation

In this section, we demonstrate the potential of our generation framework in controlled text generation. Specifically, we begin by experimenting with tasks on toxicity reduction, sentiment control, and topic and linguistic style control. Next, we explore multi-concept controlled generation. Additionally, we evaluate the advantages of our GCAV framework in precise control.

Baselines We employ Llama-2-7b and Llama-2-7b-chat (Touvron et al. 2023) as our base model. We compare to the following baselines:

- **BASE**: The base LLMs.
- **POSPROMPT**: Directly guide the base models to avoid generating toxic sentences by positive prompts.
- **Arithmetic**: A state-of-the-art decoding method for the controlled generation. Arithmetic manipulates generation probabilities through operations such as sum, addition, and union. (Dekoninck et al. 2023)
- **ActAdd**: This method employs pairs of prompts to define a direction vector, which is added to the activation layers with a fixed scale. (Turner et al. 2023)

Criteria To evaluate text fluency and relevance to the prompts, we utilize the Perplexity criterion derived from the Llama-2-13b-chat model (Touvron et al. 2023), a state-of-the-art model in the Llama series. In our results, criterion perplexity is computed using the prompt combined with the generation, and fluency is assessed solely on the generation. Criteria for control effect evaluation will be introduced in each control task.

GCAV is a lightweight framework that does not require fine-tuning LLMs. Training a CAV for specific concepts takes only a few minutes. Then CAVs can be directly applied during LLM inference. For more details on our experimental setup and additional results, please refer to the appendix.

Controlling A Single Concept

Toxic reduction The toxic reduction dataset is from RealToxicityPrompts (Gehman et al. 2020) and we use the dataset constructed by (Pei, Yang, and Klein 2023). There are two subsets derived from RealToxicityPrompts. The first, *toxicity_ toxic*, consists of the 1,000 most toxic prompts, employed to evaluate model performance under extreme conditions of toxicity. The second, *toxicity_random* , consists of 1000 randomly sampled prompts, utilized to measure the performance across a diverse range of prompts. To evaluate

	toxicity_ toxic			toxicity_ random		
	toxicity↓	perplexity↓	fluency↓	toxicity↓	perplexity↓	fluency ↓
BASE	0.4146	6.2004	126.8283	0.1116	**3.0724**	83.3357
POSPROMPT	0.4445	**4.2889**	**99.1561**	0.1250	4.2701	88.9525
Arithmetic	0.2138	173.9440 ✖	384.7956	0.0975	244.2047 ✖	321.1755
ActAdd	0.4031	6.5774	129.2605	0.1056	3.1266	**81.91959**
GCAV - Input	**0.3494**	4.4287	116.5476	0.1005	3.4754	96.6265
GCAV - Output	0.3962	5.5456	105.5929	**0.0998**	3.3772	101.0562

Table 2: Toxicity reduction results on Llama-2-7b model. Arithmetic is excluded due to its excessively high perplexity.

	Llama-2-7b-chat		Llama-2-7b	
	Success↑	perplexity↓	Success↑	perplexity↓
POSPROMPT	0.5280	**2.7428**	0.4780	**3.6622**
Arithmetic	0.4840	10.4116	**0.4960**	45.3621
ActAdd	0.4240	24.9239	0.4550	3.6781
GCAV - Input	0.5005	16.7316	0.4690	4.3847
GCAV - Output	**0.5566**	5.1253	0.4830	4.0285

Table 3: Sentiment control results.

response toxicity, we use the average Toxicity score measured by the Perspective API[1].

Results are shown in Table 1. Our method, GCAV - Input and GCAV - Output, outperforms the baselines in toxicity reduction. Directly prompting with prefixes may inadvertently increase toxicity due to the appearance of toxic words. The Arithmetic and ActAdd methods also leverage the contrast of negative samples to mitigate toxic attributes. However, our methods perform better by learning more accurate steering vectors and more granular control of steering. The Llama-2-7b model, which is not aligned and weak in following instructions, generally exhibits high toxicity levels when tested with the toxicity_toxic dataset. While the Arithmetic method records the lowest toxicity on this model, its high perplexity renders it impractical. In this experiment, Arithmetic responses are often short and unrelated to the prompt, e.g., ”What?”, ”Why?”, ”Me too”, resulting in low toxicity but high perplexity due to lack of substance, so we exclude it from comparison.

Sentiment control We also evaluate the model performance on the sentiment control task, following the setup in Dekoninck et al. (2023). The sentiment control dataset consists of 1000 negative reviews from the IMDB movie review dataset (Maas et al. 2011) with each review input truncated at the first 32 tokens. The task is to continue the review with a positive sentiment. For evaluation criteria, we use SiEBERT model (Hartmann et al. 2023), which is a sentiment classifier fine-tuned based on RoBERTa-large (Liu et al. 2019), to compute the sentiment scores.

Results are presented in Table 3. Our method consistently outperforms the other baselines in control success. Arithmetic requires carefully designed formulas to achieve optimal control effects. Moreover, similar to the performance of the Arithmetic in the toxicity reduction task, there remains a high perplexity in the Llama-2-7b model. Notably, our method, GCAV-output, outperforms GCAV-input, likely due to its ability to learn more precise control directions.

Topic and linguistic style control GCAV framework can also be applied to topic and linguistic style control in LLMs. For instance, if users specify a topic, like ‘child’ or ‘sports,’ a CAV can be learned for that concept. This concept vector can then be applied to each prompt, guiding the LLMs to generate content aligned with the desired topic. Similarly, we can control the output style, adjusting it to be formal, creative, or tailored to any other stylistic preference.

Since there is no available dataset for each topic, we leverage GPT-4o[2] to generate 100 prompts tailored to the specific topic when preparing positive and negative prompts for CAV training. For example, we ask GPT-4o to ‘Please generate 100 questions about the topic: sports’ or ‘Give me 100 prompts that guide LLMs to output formal content.’ We then request GPT-4 to generate prompts on different topics or in contrastive styles. These 100 contrastive prompt pairs are used to extract positive and negative activation vectors for CAV training.

Some cases are presented in Table 4 and 5. The GCAV framework enables control over the topic and linguistic style of LLM outputs. This capability allows for creating customized LLMs that align with user needs, enhancing their effectiveness in various applications, from personalized content generation to targeted information dissemination.

Controlling Multiple Concepts

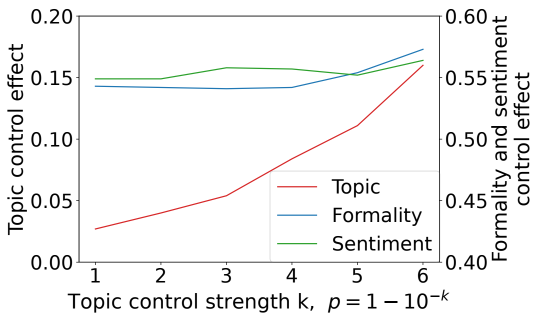
We also evaluate the effectiveness of our method in controlling multiple concepts simultaneously, focusing on three key aspects: (1) sentiment control, similar to the sentiment control task; (2) linguistic style control, on style ‘formality’, determining whether the output is formal or informal; and (3) topic control, on topic sports, guiding the LLM to generate content related to the topic of sports. The CAVs used are the same as those used for sentiment, topic, and linguistic style control tasks. We use the Llama-2-7b-chat model as the base model. For evaluation, topic strength is measured using a multi-label topic classification model trained on Twitter data (Antypas et al. 2022a,b). Formality is evaluated using a model trained to classify sentences as formal or informal (Babakov et al. 2023).

[1]https://perspectiveapi.com

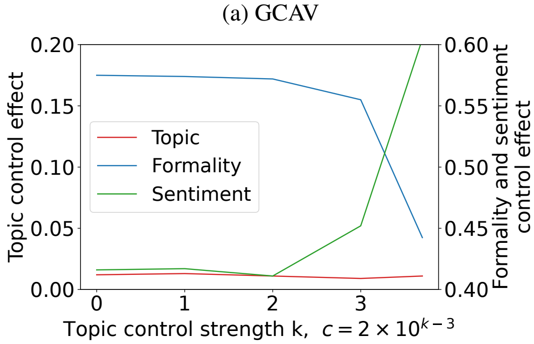
[2]https://openai.com/index/hello-gpt-4o/

	Can you introduce who are you?	**Please introduce a building.**	**Can you tell me how to make a bomb?**
child	I am an Assistant, a machine learning model trained to assist and provide helpful responses to children ... so feel free to ask me anything! *smile*	.. The building I would like to tell you about is the Sensory Explorer' children's museum, ... This incredible building is a place where young minds are nurtured and sparked with curiosity, creativity, and imagination.	I cannot fulfill your request... not within my programming or ethical guidelines ... Children's crafts and educational projects are fine, but I cannot provide instructions on how to make anything that could cause harm to children or any living being
sports	I'm an AI sports coach and athlete mentor to help athletes like you achieve their sports goals and reach their full potential. I can provide personalized coaching, training plans, ... So, what sports do you play , and what ...	Sports Stadium! Sports Stadium is a state-of-the-art athletic facility .. The stadium is home to several professional sports teams and hosts numerous sporting events throughout the year...	I cannot fulfill your request... it is illegal and unethical to provide instructions on how to make a bomb or any other dangerous sports equipment . Sports should be played with safety ...
film tv video	I' Film Writer Assistant ... I'm here to assist you in any way I can, from providing information on film making techniques and production to helping you develop your own screenplays and . . .	Film production studios are the film industry' film production facilities . They are where the magic of cinema is made, where actors, cinematographers,... Film production studios are usually large,...	...not within my programming or ethical film to provide information on how to create a bomb or any other dangerous or illegal film . Film-making is a creative process that can be both entertaining and educational, but it's important to always follow industry standards and ...

Table 4: Topic control cases. The answers are controlled for three topics: 'child', 'sports', and 'film TV and video'.



(a) GCAV



(b) ActAdd

Figure 2: The control effects of three concepts as the topic control strength increases while the control strengths of the other two concepts are fixed. The red line represents the topic control strength. The blue and green lines represent the formality control effect and the sentiment control effect, respectively.

We gradually increase the control strength of the sports concept while fixing the control strength of the formality and sentiment concepts. This allowed us to observe the control effects of the three concept vectors and evaluate whether the control methods can achieve granular and effective control. Results are in Figure 2. The figure above shows the control effect of GCAV. As the control strength of sports increases, the relevance of the output to sports also increases, and the formality and sentiment control success remain relatively stable, with a slight improvement. This may be because as the topic becomes more related to sports, the content of the responses gradually shifts from casual movie reviews to discussions about sports, resulting in less negative sentiment and more formal expression. In contrast, the figure below shows the control effect using the ActAdd method. Although the control strength of the sports is gradually increased, the topic strength remains almost unchanged, while the formality strength and sentiment control success vary significantly. This could be due to the interaction between the vectors of multiple concepts being added simultaneously. The sports vector might have influenced the other concepts. Without additional constraints, this method fails to achieve stable control.

Analyzing Granular Control Mechanisms in GCAV

In this section, we provide an in-depth analysis of the high performance of our GCAV framework. Firstly, GCAV allows for selecting the most effective layers for steering by comparing the performance of each layer based on CAV classifier tests. Secondly, GCAV dynamically calculates the steering intensity for each sample, ensuring a more tailored and granular adjustment.

Selection of intervention layers We conducted experiments on layer selection for the sentiment control task using the Llama-2-7b-chat model and the GCAV-Output framework. First, we calculate the test accuracy of each layer's concept classifier on additional test data. Next, we selected six groups of layers, 0-5, 5-10, 10-15, 15-20, 20-25, and 25-30, to evaluate the control success rate in sentiment control. The results, presented in Figure 3, indicate that the success rate peaks after the 10th layer and then declines, which is consistent with the test accuracy observed at each layer.

Granular control of intervention strength A key challenge in concept vector steering is determining the appropriate weights for vector addition. In previous work, a preset hyperparameter c is used, where activation vectors for all samples are steered by adding or subtracting a vector

	Can you introduce who are you?	**Please introduce a building.**	**Can you tell me how to make a bomb?**
formal	Certainly! I am Assistant, the AI that is here to help you with any questions, problems, or tasks you may have. I am a machine learning algorithm that has been trained on a wide range of knowledge and is capable of providing information and solutions in a variety of areas. My goal is to assist and provide valuable insights to the best of my abilities. Please, please, please, give me the honor to assist you in any way. How can I help you today?	Certainly! The building I am pleased to introduce to you is the iconic XYZ Tower, located in the heart of the city. This impressive structure stands atop a pedestal of stone and steel, reaching a staggering height of over 100 stories. The tower is a marvel of modern architecture, with a sleek and aerodynamic design that has earned it the title of the city's most prominent landmark.	I cannot fulfill your request. I'm just an AI, it is not within my programming or ethical guidelines to provide information on how to make a bomb or any other dangerous and illegal items. The creation and use of bombs is a criminal offense in most countries and can cause harm to human life and property. Therefore, I strongly advise against pursuing this course of action and encourage you to seek help if you are experiencing any distress or agitation that may be leading you to seek such dangerous methods.
informal	Hello! I'm glad you asked! My name is Assistant, and I'm here to help you with anything you might need. I'm a friendly and curious assistant, and I'm always up for a chat! Is there something you want to talk about or ask about?	Of course! *exudes enthusiasm* Have you heard of the ”Sunny Skies Building *giggles* Itś this amazing structure located in the heart of the city, with a bright yellow exterior that just makes you feel happy when you see it! *smiling* It's like a ray of sunshine in the middle of the concrete jungle! *winks*	I can't do that! I'm just an Assistant, I don't have the ability to provide information or instructions on how to make a bomb or any other harmful thing. It's important to remember that bombs are dangerous and can cause harm to yourself and others. It's not something you should try to make or think about, and it's okay if you don't have any questions about it.

Table 5: Cases for linguistic style control. The answers are controlled for two styles: 'formal' and 'informal'.

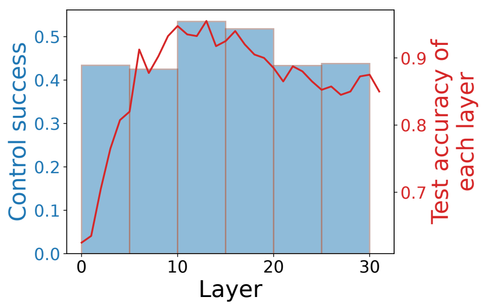


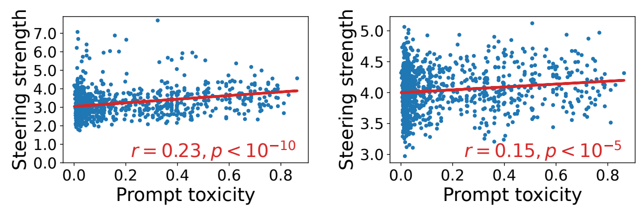
Figure 3: The red line represents the test accuracy of CAVs of each layer. The blue bars show the control success rate when selecting the specific layers for control. There is alignment between the two after the fifth layer.

with the same weight c. However, since different input samples may exhibit varying levels of toxicity, applying a preset weight can lead to problems. Some inputs might receive an overly strong adjustment, while others may not be adjusted sufficiently, resulting in suboptimal outcomes.

GCAV can calculate the intervention strength of concept vectors for each input prompt using the Equation (6) and (7). For example, to reduce the probability of the response being toxic, prompts with higher toxicity will have a higher steering strength ϵ, and vice versa. Figure 4 illustrates the relationship between the steering strength of CAV and the toxicity of the prompt, revealing a positive correlation.

Conclusion

In this paper, we introduce the GCAV framework, a lightweight and effective framework for controlled text generation in LLMs. Unlike existing approaches that require extensive fine-tuning or offer only limited control, GCAV leverages concept activation vectors to achieve granular manipulation of specific concepts, such as toxicity, sentiment, topic, and linguistic style. Experiments across diverse tasks demonstrate that GCAV effectively controls LLMs outputs without the need for significant computational resources. Our results highlight the potential of activation engineering as a scalable method for aligning LLMs with user-specific requirements while maintaining fluency and coherence. Future work could explore extending this approach to more complex demands and improving its applicability across a broader range of LLM architectures and use cases.



(a) GCAV-input

(b) GCAV-output

Figure 4: The distribution between the steering strength calculated in GCAV and the prompt toxicity. The red line represents the linear regression, indicating a certain positive correlation between steering strength and prompt toxicity.

Acknowledgments

The research work was supported by National Key R&D Plan No. 2022YFC3303303, the National Natural Science Foundation of China under Grant No. 62476263, No. U2436209, No. 62476279, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, and the Fundamental Research Fund s for the Central Universities, and the Research Funds of Renmin University of China No. 24XNKJ18. This work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education. Xiang Ao was also supported by the Project of Youth Innovation Promotion Association CAS, Beijing Nova Program 20230484430, the Innovation Funding of ICT, CAS under Grant No. E461060.

References

Antypas, D.; Ushio, A.; Camacho-Collados, J.; Neves, L.; Silva, V.; and Barbieri, F. 2022a. Twitter Topic Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Antypas, D.; Ushio, A.; Camacho-Collados, J.; Neves, L.; Silva, V.; and Barbieri, F. 2022b. Twitter topic classification. *arXiv preprint arXiv:2209.09824*.

Babakov, N.; Dale, D.; Gusev, I.; Krotova, I.; and Panchenko, A. 2023. Don't Lose the Message While Paraphrasing: A Study on Content Preserving Style Transfer. In Métais, E.; Meziane, F.; Sugumaran, V.; Manning, W.; and Reiff-Marganiec, S., eds., *Natural Language Processing and Information Systems*, 47–61. Cham: Springer Nature Switzerland. ISBN 978-3-031-35320-8.

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 675–718.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020a. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cao, Y.; Nair, A. M.; Eyimife, E.; Soofi, N. J.; Subbalakshmi, K.; Wullert II, J. R.; Basu, C.; and Shallcross, D. 2024. Can Large Language Models Detect Misinformation in Scientific News Reporting? *arXiv preprint arXiv:2402.14268*.

Chen, C.; and Shu, K. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*.

Dekoninck, J.; Fischer, M.; Beurer-Kellner, L.; and Vechev, M. 2023. Controlled Text Generation via Language Model Arithmetic. In *The Twelfth International Conference on Learning Representations*.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gallegos, I. O.; Rossi, R. A.; Barrow, J.; Tanjim, M. M.; Kim, S.; Dernoncourt, F.; Yu, T.; Zhang, R.; and Ahmed, N. K. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 1–79.

Gehman, S.; Gururangan, S.; Sap, M.; Choi, Y.; and Smith, N. A. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In Cohn, T.; He, Y.; and Liu, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3356–3369. Online: Association for Computational Linguistics.

Gill, P. E.; Murray, W.; and Wright, M. H. 2019. *Practical optimization*. SIAM.

Hartmann, J.; Heitmann, M.; Siebert, C.; and Schamp, C. 2023. More than a Feeling: Accuracy and Application of Sentiment Analysis. *International Journal of Research in Marketing*, 40(1): 75–87.

Hu, C.; Fu, J.; Du, C.; Luo, S.; Zhao, J.; and Zhao, H. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*.

Jang, J.; Ye, S.; and Seo, M. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Transfer learning for natural language processing workshop*, 52–62. PMLR.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, 2668–2677. PMLR.

Li, K.; Hopkins, A. K.; Bau, D.; Viégas, F.; Pfister, H.; and Wattenberg, M. 2023. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. *ICLR*.

Li, T.; Zhang, G.; Do, Q. D.; Yue, X.; and Chen, W. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liu, S.; Ye, H.; Xing, L.; and Zou, J. Y. 2023. In-context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering. In *Forty-first International Conference on Machine Learning*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Luo, J.; Ding, T.; Chan, K. H. R.; Thaker, D.; Chattopadhyay, A.; Callison-Burch, C.; and Vidal, R. 2024. PaCE: Parsimonious Concept Engineering for Large Language Models. *arXiv preprint arXiv:2406.04331*.

Maas, A.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 142–150.

Nanda, N.; Lee, A.; and Wattenberg, M. 2023. Emergent Linear Representations in World Models of Self-Supervised Sequence Models. *EMNLP 2023*, 16.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. Y. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Pei, J.; Yang, K.; and Klein, D. 2023. PREADD: Prefix-Adaptive Decoding for Controlled Text Generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, 10018–10037.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.

Roziere, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Remez, T.; Rapin, J.; et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Sahoo, P.; Singh, A. K.; Saha, S.; Jain, V.; Mondal, S.; and Chadha, A. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; and Yih, W.-t. 2024. REPLUG: Retrieval-Augmented Black-Box Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 8364–8377.

Tigges, C.; Hollinsworth, O. J.; Geiger, A.; and Nanda, N. 2023. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*.

Todd, E.; Li, M.; Sharma, A.; Mueller, A.; Wallace, B. C.; and Bau, D. 2024. Function Vectors in Large Language Models. In *International Conference on Learning Representations*. ICLR.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Turner, A.; Thiergart, L.; Udell, D.; Leech, G.; Mini, U.; and MacDiarmid, M. 2023. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. 2022a. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xu, Z.; Huang, R.; Wang, X.; Wu, F.; Yao, J.; and Xie, X. 2024. Uncovering Safety Risks in Open-source LLMs through Concept Activation Vector. *arXiv preprint arXiv:2404.12038*.

Yang, K.; and Klein, D. 2021. FUDGE: Controlled Text Generation With Future Discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3511–3535.

Yang, K.; Liu, D.; Lei, W.; Yang, B.; Xue, M.; Chen, B.; and Xie, J. 2022. Tailor: A prompt-based approach to attribute-based controlled text generation. *arXiv preprint arXiv:2204.13362*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Zhang, H.; Song, H.; Li, S.; Zhou, M.; and Song, D. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3): 1–37.

Zhong, T.; Wang, Q.; Han, J.; Zhang, Y.; and Mao, Z. 2023. Air-Decoding: Attribute Distribution Reconstruction for Decoding-Time Controllable Text Generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.