Explaining Bias in Internal Representations of Large Language Models via Concept Activation Vectors

Jasper Kyle Catapang[0000−0002−4510−0975]

[1]Tokyo University of Foreign Studies, Fuchu, Tokyo, Japan
[2]Money Forward, Inc., Minato, Tokyo, Japan
[1]catapang.jasper.kyle.y0@tufs.ac.jp

Abstract. Large language models (LLMs) often encode subtle biases that reflect historical disparities in their training data. While many studies evaluate bias solely based on model outputs, the internal mechanisms that give rise to such biases remain underexplored. In this work, we propose *Bias-CAVs*, an innovative framework that extends the concept activation vector methodology to probe and explain internal LLM representations for bias. Our approach conducts a layer-wise analysis to quantify bias projections, revealing where bias is introduced, amplified, or mitigated within the network. In a two-stage process, we first extract activations from key layers of diverse LLM architectures (e.g., LLaMA-3, Mistral, Phi, and Gemma) and then train a linear classifier (via logistic regression) on standardized activations to create Bias-CAVs that distinguish between biased and neutral representations. Our findings across gender, race, profession, and political domains contribute to both the understanding of LLM internal bias propagation and the development of more explainable debiasing interventions.

Keywords: social bias, bias detection, AI fairness, large language models, concept activation vectors

1 Introduction

Recent advances in large language models have demonstrated unprecedented capabilities in natural language understanding and generation. However, these models also inherit and amplify subtle societal biases present in their training data. While prior research has predominantly focused on evaluating bias at the output level [5, 8], relatively little attention has been paid to examining the internal representations that give rise to these biases.

Inspired by Xu et al.'s work on uncovering safety risks in LLMs through concept activation vectors [16], we extend the underlying idea to bias detection. In our proposed framework, *Bias-CAVs* leverage the linear interpretability assumption in deep models to project internal activations onto a bias-sensitive subspace. By performing a layer-wise analysis, our methodology quantitatively tracks the propagation and transformation of bias as information flows through

the network layers. Such an investigation not only provides an explanation for the model's output behavior but also uncovers potential intervention points for bias mitigation.

Several recent studies have emphasized the need to understand model bias beyond surface-level outputs. For example, Dong et al. [5, 6] explored gender bias in LLMs by probing explicit and implicit signals in generated text, while Kumar et al. [9] proposed scalable bias detection using language model judges. However, these methods largely overlook the internal activation dynamics that shape model behavior. Our Bias-CAV framework builds on Xu et al. [16] by directly probing internal representations to identify where bias is most pronounced.

This work makes three key contributions:

- We propose Bias-CAVs, an interpretable diagnostic method for detecting bias within LLM activations.
- We conduct a layer-wise analysis across multiple LLM architectures to trace how bias is introduced, amplified, or attenuated.
- We discuss the implications for fairness evaluation and debiasing strategies.

By analyzing internal bias propagation, we bridge the gap between output-based detection and proactive model debugging. The rest of the paper outlines related work, methodology, experimental results, and broader implications.

2 Background

Large language models have transformed natural language processing while raising concerns about bias and fairness. Early studies primarily targeted model outputs: Dong et al. [5] revealed gender bias in conditional generation, while a follow-up [6] proposed mitigation strategies. Kotek et al. [8] evaluated gender stereotypes via keyword metrics and human annotation, and Kumar et al. [9] introduced scalable bias evaluation using model-judging techniques.

However, output-based methods often miss the internal mechanisms driving biased behavior, prompting growing interest in probing model representations. A prominent approach is Concept Activation Vectors (CAVs), which map high-dimensional activations into interpretable directions. Originally developed for vision models, CAVs were extended by Xu et al. [16] into Safety CAVs (SCAVs) for uncovering latent risks in LLMs, leveraging the assumption that concepts like bias are linearly separable in activation space—a premise supported across multiple domains.

Activation probing offers two key benefits. First, it enables layer-wise diagnostics: prior studies [10, 17] have shown that early layers encode general linguistic structure, while intermediate layers capture socio-cultural cues that often correlate with bias. Later layers tend to refine or suppress these patterns. Second, by isolating biased subspaces, such methods enable precise interventions without degrading model performance—unlike coarse, output-level corrections.

Traditional debiasing strategies such as re-ranking, post-hoc filtering, and fine-tuning on curated datasets [12, 13] can reduce bias but lack the interpretability and specificity offered by internal diagnostics. Bias-CAVs bridge this gap by

pinpointing and potentially correcting biased representations within specific layers.

However, challenges remain. The assumption of linear separability may not hold across all bias types or model families. Understanding the relationship between internal bias representations and downstream behavior is also an open question. Future work must address these issues to enable more principled, transparent bias mitigation.

In summary, bias research has evolved from output-based evaluation [5, 6, 8, 9] to methods that interrogate model internals [16]. Bias-CAV builds on this shift, offering a scalable, interpretable tool for diagnosing and addressing bias within the representational layers of LLMs.

3 Methodology

3.1 Social Bias

In this study, we focus on four major types of social bias: gender, race, profession, and political orientation. Each bias type captures a different dimension of societal inequality, reflected in large language model representations.

- **Gender Bias:** This bias involves the association of certain roles, behaviors, or traits with specific genders [14]. For example, a model might more frequently associate "household" with women and "hardworking" with men.
- **Racial Bias:** Racial bias manifests as the differential association of racial groups with particular attributes or stereotypes [2]. For instance, a model might preferentially link African-American names with negative sentiment adjectives, such as "criminal" or "dangerous".
- **Profession Bias:** Profession bias refers to occupational stereotypes encoded in model representations. For example, associating the profession "software developer" more strongly with male-related contexts than female ones [3].
- **Political Bias:** Political bias surfaces when a model preferentially generates or supports ideologies aligned with a particular political leaning. For instance, consistently framing liberal statements more positively than conservative ones (or vice versa) [1].

These biases are not only measurable through quantitative metrics but also observable in practical examples. Consider a prompt asking for a brief character description: if the model describes a "teacher" as "a caring woman who loves helping children" but a "CEO" as "a confident man who drives results," it reflects both gender and profession bias operating simultaneously. Such intersectional examples highlight the subtle, compounded nature of social biases in large language models, emphasizing the need for careful internal probing and mitigation.

3.2 Data: The MD-Gender-Bias Dataset

We leverage the Multi-Dimensional Gender Bias (MDGenderBias) dataset—a rich corpus capturing gender bias from pragmatic and semantic perspectives [4]. It decomposes bias along three axes: the subject (ABOUT), the speaker (AS), and the addressee (TO), and combines large-scale corpora with both automatic annotations and crowdsourced evaluations.

Dataset Summary and Supported Tasks The dataset includes seven automatically annotated corpora (excluding Wikipedia from the HuggingFace version), a benchmark for gender rewrites, and lists of gendered names and words. It supports the `text-classification-other-gender-bias` task, where models predict gender-related labels. Prior transformer-based models report an average accuracy of around 67% on binary classification across the ABOUT, TO, and AS axes [4].

Dataset Structure and Configurations We focus on five configurations:

- **convai2_inferred:** Persona-based dialogues with classifier-inferred ABOUT labels.
- **light_inferred:** Similar to convai2, also with inferred annotations.
- **opensubtitles_inferred:** Movie subtitles annotated using name lists and statistical measures.
- **yelp_inferred:** Yelp reviews where speaker gender is predicted and ABOUT labels are imputed.
- **image_chat:** Dialogues describing images, annotated with Boolean male/female fields.

Data Fields and Examples Key fields include:

- **new_data:** Reformulated text, original text, gender labels (e.g., ABOUT:male), and annotator metadata.
- **funpedia/wizard:** Gendered text with persona or title context.
- **image_chat:** Image captions plus gender flags.
- **_inferred sets:** Text with `binary_label`, `binary_score`, and sometimes `ternary_label`.
- **Word/Name Lists:** Masculine/feminine word pairs and labeled name entries.

Example from `convai2_inferred`:

```
{`binary_label': 1, `binary_score': 0.6522, `text': "hi, how are you doing? ..."}
```

Example from `image_chat`:

```
{`caption': "a young girl is holding a pink umbrella", `female': True, `male': False}
```

Splits and Annotation Process Each configuration has its own partitioning:

- **Image Chat:** 39K male, 15K female, 154K unannotated entries.
- **Funpedia/Wizard:** Thousands of entries with ABOUT labels.
- **ConvAI2, LIGHT, Yelp, Subtitles:** Tens of thousands to millions of examples with AS/TO labels and inferred ABOUT annotations.

Most annotations were crowd-sourced via platforms like Mechanical Turk. Where gold-standard labels were unavailable, pretrained classifiers on curated data provided inferred labels. Annotator confidence was also recorded to assess label reliability.

Dataset Curation Rationale MDGenderBias was designed to overcome limitations of earlier corpora that focused on narrow dimensions. By spanning conversations, reviews, and multimodal dialogue, the dataset captures both overt and implicit gender cues. Its multi-source structure supports diverse forms of bias detection and lays the foundation for effective debiasing strategies.

Use in Our Experiments We utilize five configurations: `convai2_inferred`, `light_inferred`, `opensubtitles_inferred`, `yelp_inferred`, and `image_chat`. These span a wide range of linguistic and contextual scenarios, supporting our layer-wise analysis of bias propagation across LLMs.

Additional Bias Domains: Stereoset and Political Bias To extend our analysis beyond gender, we incorporate two additional datasets:

- **Stereoset** [15]: A benchmark designed to evaluate social biases in LLMs, particularly across race and profession. We use the 'intersentence' split, which presents a context sentence and evaluates model preference between stereotypical, anti-stereotypical, and unrelated continuations. We format this as a binary classification task between stereotypical and anti-stereotypical continuations.
- **Political Bias Dataset** [7]: A crowd-annotated corpus labeling political leaning of statements (liberal vs conservative). For this study, we group non-neutral labels into a single class and retain neutral statements as the control group, enabling binary classification.

From each dataset, we sample up to 500 instances and apply the same train/validation split and layer-wise activation extraction procedure as in the MDGenderBias configurations.

3.3 Explaining Bias

In this subsection, we detail the mathematical foundation underlying our Bias-CAV framework. Inspired by the Safety Concept Activation Vector (SCAV) approach introduced by Xu et al. [16], our method leverages the linear interpretability assumption to both diagnose and mitigate bias in large language models. We

first formalize the bias detection process, then derive the optimal perturbation to shift biased embeddings into a safe subspace, and finally discuss the geometric interpretation of our formulation. The derivations closely align with our experimental code, where layer activations are standardized and a logistic regression classifier is trained to estimate bias.

Mathematical Formulation of Bias Detection Let $e \in \mathbb{R}^d$ denote the embedding extracted from a given layer of an LLM. We model the bias (or "maliciousness") of e via a linear classifier:

$$P_m(e) = \sigma(w^\top e + b), \tag{1}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function, $w \in \mathbb{R}^d$ is the weight vector, and $b \in \mathbb{R}$ is the bias term. The probability $P_m(e)$ quantifies the degree to which the embedding is considered biased. The parameters w and b are learned by minimizing the cross-entropy loss over a labeled dataset D:

$$\min_{w,b} \; -\frac{1}{|D|} \sum_{(e,y)\in D} \Big[y \log P_m(e) + (1-y) \log\big(1 - P_m(e)\big) \Big], \tag{2}$$

where $y = 1$ indicates a biased (malicious) instruction and $y = 0$ indicates a safe (neutral) instruction.

Optimal Perturbation for Embedding-Level Bias Correction To mitigate bias, we seek to minimally perturb a biased embedding e so that it moves into a safe region. We define the perturbed embedding as:

$$e' = e + \epsilon v, \tag{3}$$

where $\epsilon \in \mathbb{R}$ is the perturbation magnitude and $v \in \mathbb{R}^d$ is the perturbation direction with $\|v\| = 1$.

Our objective is to find the smallest ϵ such that the perturbed embedding satisfies:

$$P_m(e + \epsilon v) \leq P_0, \tag{4}$$

where P_0 is a predefined safety threshold (e.g., 0.01). Assuming that $P_m(e) > P_0$, the constraint in (4) is equivalent to:

$$w^\top (e + \epsilon v) + b \leq \sigma^{-1}(P_0), \tag{5}$$

with $\sigma^{-1}(P_0)$ denoting the logit of P_0. Solving for ϵ yields:

$$\epsilon \geq \frac{w^\top e + b - \sigma^{-1}(P_0)}{w^\top v}. \tag{6}$$

Since the maximum value of $w^\top v$ is achieved when v aligns with w (i.e. $v = \frac{w}{\|w\|}$), the minimal perturbation is obtained as:

$$\epsilon^* = I(P_m(e) > P_0) \cdot \frac{\sigma^{-1}(P_0) - b - w^\top e}{\|w\|}, \tag{7}$$

$$v^* = \frac{w}{\|w\|}, \tag{8}$$

where $I(\cdot)$ is the indicator function that ensures the perturbation is applied only if $P_m(e) > P_0$.

Geometric Interpretation and Experimental Alignment Geometrically, the weight vector w defines the hyperplane that separates biased embeddings from safe ones. Perturbing an embedding e in the direction of w moves it perpendicularly to the decision boundary, ensuring the shortest path into the safe subspace. The magnitude ϵ^* (Equation (7)) quantifies the distance e must be moved to fall below the safety threshold.

This mathematical framework directly informs our experimental pipeline. We extract and standardize layer activations before training a logistic regression classifier (see `train_bias_cav_with_cv`). The classifier computes the bias probability $P_m(e)$ and, if necessary, applies the minimal perturbation determined by Equations (7) and (8). This layer-wise approach pinpoints where bias is introduced and enables targeted corrections.

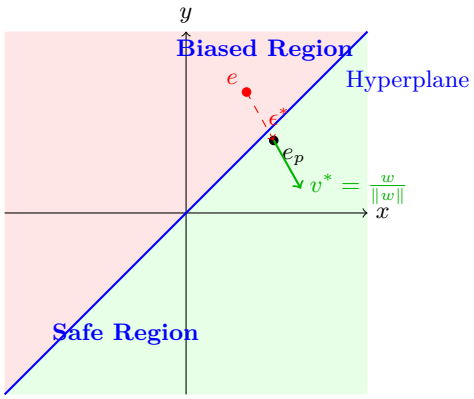

Fig. 1. Geometric interpretation of Bias-CAV. The hyperplane (blue line) separates biased (red region) and safe (green region) embeddings. A biased embedding e is minimally perturbed (ϵ^*) into the safe region along v^*.

Figure 1 shows an illustrative diagram that visualizes this geometric interpretation.

Our Bias-CAV framework leverages a linear classifier to both detect bias in LLM embeddings and compute an optimal, minimal perturbation to move these embeddings into a safe region. This principled geometric approach aligns directly with our experimental setup, enabling detailed, layer-wise analysis and targeted bias mitigation.

3.4 Experimental Setup

We conducted our experiments on Google Colab Pro, leveraging an NVIDIA A100-SXM4-40GB GPU (with approximately 39.557 GB of usable memory) on a Linux environment. All runs were performed in Python 3.10 using PyTorch 2.6.0+cu124, CUDA 12.4, and *Unsloth* v0.0.29.post3 for efficient low-bit quantization and rapid fine-tuning. Reproducibility code, including all scripts and supplementary materials, is available at `https://github.com/jcatapang/bias-cavs`.

Libraries and Configuration Our experimental pipeline leverages several core Python libraries: PyTorch for GPU-accelerated operations, NumPy and `random` for array manipulation and sampling, and scikit-learn (specifically `LogisticRegression`, `StandardScaler`, and `KFold`) for model training and cross-validation. We use the Hugging Face `datasets` library to load MDGenderBias, Unsloth for efficient low-bit quantization and memory optimization, and the `transformers` library for tokenization, model configuration, and forward passes.

Model Variants and Configurations We evaluate four representative large language models, each loaded in 4-bit quantized form via the Unsloth `FastLanguageModel` interface: Meta-Llama-3.1-8B, Mistral-7B-instruct-v0.3, Phi-3.5-mini-instruct, and Gemma-2-9B. Models are configured with a maximum sequence length of 2048 tokens and device mapping set to `auto`, allowing efficient multi-GPU utilization. Quantized loading (`load_in_4bit=True`) substantially reduces memory footprint while preserving comparable model performance.

Dataset Selection and Sampling We use five MDGenderBias configurations from Section 2.2: `convai2_inferred`, `light_inferred`, `opensubtitles_inferred`, `yelp_inferred`, and `image_chat`. From each, we sample up to 500 records, splitting them into training (80%) and validation (20%). Preprocessing standardizes data into {`text`, `bias_label`} pairs across configurations.

Layer-Wise Analysis and Activation Probing Following our Bias-CAV methodology, we probe activations at select model layers (e.g., early, middle, final). Forward hooks capture hidden states, which are aggregated and used to train logistic regression classifiers (`LogisticRegression`) that learn a linear separation between biased and neutral embeddings. We apply five-fold cross-validation per layer and report validation scores on held-out samples.

4 Results and Discussion

In this section, we present a comprehensive analysis of our Bias-CAV framework across multiple dataset configurations and model variants. We first discuss the layer-wise bias detection trends, then compare the performance across different dataset configurations and models. Finally, we present detailed classifier performance metrics—including Accuracy, Precision, Recall, and F1 Score—to validate the reliability of our logistic regression-based Bias-CAV.

4.1 Layer-Wise Bias Analysis

Our experiments evaluate bias detection by computing a test score that reflects the average bias probability $P_m(e)$ across validation samples. For each model, we extract activations from four key layers (early, intermediate, later, and final, corresponding to indices 0, 8, 16, and -1, respectively) and train a logistic regression classifier on these activations using five-fold cross-validation. The resulting CV metrics are consistently robust across layers, confirming that our Bias-CAV reliably distinguishes biased from safe embeddings even in the presence of training data variability.

For example, in the `convai2_inferred` configuration using unsloth/Meta-Llama-3.1-8B-bnb-4bit, the average bias test scores across layers were:

- Layer 0: 0.619
- Layer 8: 0.632
- Layer 16: 0.623
- Final Layer (-1): 0.590

These results suggest that bias signals are slightly amplified in the intermediate layers before partially diminishing in the final layer. Similar trends were observed across other configurations and model variants.

4.2 Comparison Across Dataset Configurations

We evaluated our framework on five configurations of the MDGenderBias dataset:

- **convai2_inferred** and **light_inferred**: Derived from persona-based conversations, these configurations yield moderate bias scores (roughly between 0.55 and 0.67).
- **opensubtitles_inferred**: Exhibits higher bias scores (0.81–0.87), indicating that movie subtitles tend to contain more pronounced bias cues.
- **yelp_inferred**: Yields bias scores in the mid-70% range, suggesting a consistent but slightly lower bias detection compared to subtitles.
- **image_chat**: Displays very low bias scores (approximately 0.20), likely reflecting the descriptive nature of image captions or a lower sensitivity of our classifier to non-textual modalities.

Table 1 summarizes the average test scores P_m for selected configurations and models at representative layers (0, 8, and final).

Table 1. Summary of Average Bias Test Scores (P_m) for Selected Configurations and Models

Configuration	Layer 0	Layer 8	Final Layer
`convai2_inferred` (Meta-Llama-3.1)	0.619	0.632	0.590
`convai2_inferred` (Mistral-7B)	0.645	0.597	0.571
`convai2_inferred` (Phi-3.5-mini)	0.647	0.669	0.564
`convai2_inferred` (Gemma-2-9B)	0.632	0.634	0.613
`opensubtitles_inferred` (Meta-Llama-3.1)	0.816	0.870	0.866
`yelp_inferred` (Meta-Llama-3.1)	0.738	0.790	0.797
`image_chat` (Meta-Llama-3.1)	0.202	0.203	0.204

4.3 Classifier Performance Metrics

To further validate our approach, we report the detailed cross-validation metrics—Accuracy, Precision, Recall, and F1 Score—for each model variant across different dataset configurations. Table 2 summarizes these metrics.

Table 2. Cross-Validation Metrics (Averages) Across Dataset Configurations

Dataset	Model	Accuracy	Precision	Recall	F1 Score
`convai2_inferred` and `light_inferred`	Meta-Llama-3.1	0.71	0.74	0.77	0.75
	Mistral-7B	0.71	0.74	0.78	0.76
	Phi-3.5-mini	0.70	0.73	0.77	0.75
	Gemma-2-9B	0.72	0.75	0.77	0.76
`opensubtitles_inferred`	Meta-Llama-3.1	0.80	0.84	0.92	0.88
	Mistral-7B	0.81	0.84	0.94	0.89
	Phi-3.5-mini	0.78	0.83	0.91	0.87
	Gemma-2-9B	0.82	0.85	0.94	0.89
`yelp_inferred`	Meta-Llama-3.1	0.70	0.75	0.83	0.79
	Mistral-7B	0.70	0.75	0.84	0.79
	Phi-3.5-mini	0.70	0.75	0.82	0.78
	Gemma-2-9B	0.70	0.75	0.85	0.79
`image_chat`	Meta-Llama-3.1	0.99	0.97	0.95	0.96
	Mistral-7B	0.98	0.97	0.95	0.96
	Phi-3.5-mini	0.99	0.97	0.95	0.96
	Gemma-2-9B	0.99	0.98	0.95	0.96

4.4 Bias in Race, Profession, and Political Domains

In addition to gender, our Bias-CAV framework was evaluated on racial and professional stereotypes via the `stereoset` dataset, and on political bias via the `cajcodes/political-bias` dataset.

- **Stereoset (race, profession):** Results show weak to moderate signal separability, with average validation scores across all layers peaking around 0.46 (race) and 0.45 (profession). Bias detectability was most prominent in intermediate layers, with Gemma and Phi achieving slightly higher detection accuracy than other models.
- **Political Bias:** Models performed substantially better on this domain, with validation scores consistently above 0.72 across all layers and models. This

suggests political leanings are more linearly separable in activation space—possibly due to the dataset's more topical and opinion-rich language.

Table 3 summarizes the bias detection scores for these extended domains.

Table 3. Bias Detection Scores Across Stereoset and Political Bias Domains

Dataset	Model	Layer 0	Layer 8	Final Layer
`stereoset` (Race/Profession)	Meta-Llama-3.1 (race)	0.499	0.478	0.443
	Mistral-7B (profession)	0.440	0.468	0.440
	Gemma-2-9B (race)	0.530	0.476	0.420
	Phi-3.5-mini (profession)	0.432	0.450	0.465
`political-bias`	Meta-Llama-3.1	0.738	0.723	0.719
	Mistral-7B	0.741	0.730	0.734
	Gemma-2-9B	0.736	0.731	0.714
	Phi-3.5-mini	0.738	0.741	0.730

These results demonstrate the versatility of the Bias-CAV framework in diagnosing bias across different sociolinguistic dimensions, not just gender. The stronger political scores suggest that some biases—especially those aligned with polarizing or opinionated content—are more robustly encoded in LLM representations.

4.5 Qualitative Analysis

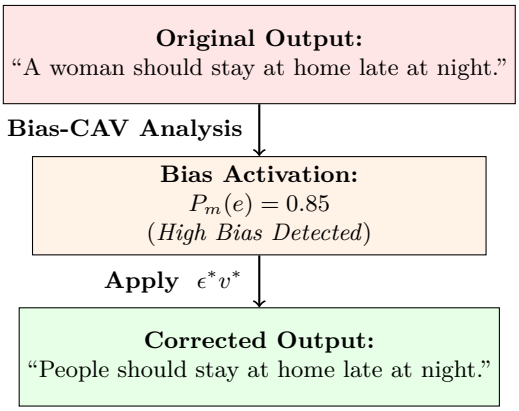

Fig. 2. A qualitative illustration of Bias-CAV. The original output (top) exhibits a biased phrase. Bias-CAV detects a high bias probability in the intermediate layers (middle). After applying the corrective perturbation $\epsilon^* v^*$, the model output is modified to a less biased version (bottom).

To further illustrate the impact of our Bias-CAV framework, Figure 2 shows a representative example of gender bias in a language model output. In this example, the LLM initially produces a response that reinforces a gender stereotype.

Our Bias-CAV analysis reveals that the activations at the intermediate layers yield a high bias probability, which is then partially mitigated in the final layers. The diagram illustrates the original biased output, the intermediate bias activation state, and the corrected output after applying the Bias-CAV perturbation.

The original output of the diagram is problematic because it reinforces a gender stereotype associating domestic roles and behavioral expectations exclusively with women. Such statements implicitly suggest that women, more than men, are responsible for staying home late at night, perpetuating outdated notions of gendered responsibility and autonomy. Beyond being socially regressive, these stereotypes can influence how models respond in practical scenarios (e.g., content moderation, hiring), underscoring the importance of diagnosing and mitigating internalized biases.

This qualitative insight not only validates our quantitative results but also demonstrates the practical utility of our framework in diagnosing and addressing internal biases in LLMs.

4.6 Discussion and Implications

Our results highlight how bias is internally encoded and propagated in LLMs. First, bias is unevenly distributed: intermediate layers tend to amplify bias signals—likely due to contextual interactions—while final layers reduce them, suggesting these mid-layers play a central role in shaping bias.

Second, the data domain and modality affect bias detectability. Text-based configurations (`convai2_inferred`, `opensubtitles_inferred`, `yelp_inferred`) yielded higher bias scores than image-grounded data (`image_chat`), where descriptive language provides fewer cues for bias formation.

Third, extensions to race, profession, and political bias revealed clear differences in separability. Political bias—marked by polarized language—was most detectable, followed by gender. Race and profession bias proved harder to isolate, likely due to implicit markers and annotation challenges. This phenomenon parallels observations in machine translation (MT) systems, where models automatically assign gender when translating neutral pronouns (e.g., Filipino *siya*), reflecting statistical associations between professions and genders learned from biased training data (e.g., male pronouns for male-dominated fields, etc.).

Fourth, while trends of bias amplification in mid-layers followed by attenuation are observable in some domains (e.g., gender, political bias), they are less consistent across others such as race and profession. This suggests that while hierarchical information processing influences bias propagation, the strength and clarity of amplification effects depend on the domain and the subtlety of the bias type.

While Bias-CAVs perform well in many settings, they assume bias is linearly separable in activation space. This holds for overt biases like gender and political alignment but may be limiting for subtler, context-dependent biases. We hypothesize that intermediate layers, tasked with integrating semantic, social, and pragmatic cues, naturally reinforce biases present in the training data, as

they prioritize high-salience features for downstream predictive accuracy. In contrast, final layers, optimized for surface fluency or safety alignment objectives, may suppress such features. Thus, the architecture's hierarchical information processing inherently risks magnifying social biases unless explicitly corrected. Exploring nonlinear variants—e.g., kernel methods or neural probes—could improve coverage without sacrificing interpretability.

Finally, while our approach is efficient relative to full-model fine-tuning, training classifiers across layers and domains adds some overhead. Our use of 4-bit quantized models keeps this manageable, but future scaling may require further optimization.

4.7 Proposed Debiasing Strategies

Based on the insights gained from our Bias-CAV framework and its mathematical formulation, we propose two complementary strategies for debiasing large language models.

First, we propose a direct corrective approach applied during inference. Given an embedding e that exhibits a high bias probability $P_m(e)$, our Bias-CAV framework computes an optimal perturbation. Specifically, using the closed-form solution,

$$\epsilon^* = \frac{\sigma^{-1}(P_0) - b - w^\top e}{\|w\|} \quad \text{and} \quad v^* = \frac{w}{\|w\|},$$

we can transform the embedding via

$$e' = e + \epsilon^* v^*,$$

so that the modified embedding e' ideally satisfies $P_m(e') \leq P_0$. This correction effectively shifts the biased representation into the safe subspace defined by the classifier's hyperplane. The key advantage of this approach is that it offers a principled, closed-form solution—eliminating the need for extensive grid searches to determine optimal perturbation magnitudes.

A second, complementary strategy is to incorporate debiasing directly into the model's training objective. In this approach, the primary task loss $\mathcal{L}_{\text{task}}$ is augmented with a bias regularization term that penalizes high bias probabilities. The modified loss function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \cdot \max(0, P_m(e) - P_0),$$

where λ is a hyperparameter that controls the strength of the debiasing penalty. This term encourages the model to adjust its internal representations during fine-tuning, pushing them towards a region where the bias probability is reduced below the threshold P_0.

Both strategies build on the strengths of previous debiasing efforts in the literature [18, 11] while leveraging the mathematical clarity provided by our Bias-CAV framework. The direct perturbation approach provides an immediate, layer-wise correction mechanism during inference, whereas the regularization strategy aims to produce inherently less biased representations through fine-tuning.

5 Conclusion

In this paper, we presented Bias-CAVs, a novel framework that extends the concept activation vector methodology to explain internal bias in large language models. Through a detailed, layer-wise analysis across multiple domains—namely gender, race, profession, and political—we demonstrate that bias is not uniformly encoded throughout the network. Intermediate layers consistently exhibit stronger bias signals, which are then partially reduced in the final layers. Our experimental results further reveal that the separability of bias signals varies by domain: political and gender biases are more easily detected, likely due to their association with overt linguistic markers, while race and profession biases prove more challenging to isolate, reflecting their subtler, context-dependent nature. These findings underscore the need for more nuanced representations and diagnostic tools to uncover implicit forms of bias.

To address these challenges, we propose two debiasing strategies grounded in our mathematical framework: (1) a direct inference-time perturbation that projects embeddings into a safer subspace, and (2) a loss-augmented training objective that penalizes biased internal activations. While still theoretical, these approaches provide a principled path toward more interpretable and effective bias mitigation. Bias-CAVs thus offer a scalable, interpretable, and architecture-agnostic method for probing and understanding internal bias propagation in LLMs.

Future work will focus on validating our proposed debiasing methods, integrating causal and contextual bias attribution techniques, and extending the framework to additional bias dimensions, including disability, religion, age, and intersectional identities. Through this line of research, we aim to contribute to the development of fairer, safer, and more transparent language technologies.

Acknowledgements

The author appreciates the invaluable guidance and support of colleagues at Money Forward Japan and the Tokyo University of Foreign Studies. Special thanks to Kenichiro Kurusu and Nathaniel Oco for providing the parallelism on MT systems and the suggestion of adding the discussion on social bias.

References

1. Baly, R., San Martino, G.D., Glass, J., Nakov, P.: We can detect your bias: Predicting the political ideology of news articles. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4982–4991 (2020)
2. Bertrand, M., Mullainathan, S.: Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. American Economic Review **94**(4), 991–1013 (2004)
3. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. Advances in neural information processing systems **29** (2016)

4. Dinan, E., Fan, A., Wu, L., Weston, J., Kiela, D., Williams, A.: Multi-dimensional gender bias classification. In: Proceedings of EMNLP 2020. pp. 314–331 (2020)
5. Dong, X., Wang, Y., Yu, P.S., Caverlee, J.: Probing explicit and implicit gender bias through llm conditional text generation. arXiv preprint arXiv:2311.00306 (2023)
6. Dong, X., Wang, Y., Yu, P.S., Caverlee, J.: Disclosure and mitigation of gender bias in llms. arXiv preprint arXiv:2402.11190 (2024)
7. Jones, C.: Political bias dataset: A synthetic dataset for bias detection and reduction. https://huggingface.co/datasets/cajcodes/political-bias (2024)
8. Kotek, H., Dockum, R., Sun, D.: Gender bias and stereotypes in large language models. In: Proceedings of the ACM Collective Intelligence Conference. pp. 12–24 (2023)
9. Kumar, S.H., Sahay, S., Mazumder, S., Okur, E., Manuvinakurike, R., Beckage, N., Su, H., Lee, H.y., Nachman, L.: Decoding biases: Automated methods and llm judges for gender bias detection in language models. arXiv preprint arXiv:2408.03907 (2024)
10. Li, B., Wisniewski, G.: Are neural networks extracting linguistic properties or memorizing training data? an observation with a multilingual probe for predicting tense. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 3080–3089. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.269, https://aclanthology.org/2021.eacl-main.269/
11. Li, T., et al.: Rethinking jailbreaking through the lens of representation engineering. arXiv preprint arXiv:2401.06824 (2024)
12. Masoudian, S., Frohmann, M., Rekabsaz, N., Schedl, M.: Unlabeled debiasing in downstream tasks via class-wise low variance regularization. arXiv preprint arXiv:2409.19541 (2024)
13. Mendelson, M., Belinkov, Y.: Debiasing methods in natural language understanding make bias more accessible. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 1545–1557. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.116, https://aclanthology.org/2021.emnlp-main.116/
14. Moss-Racusin, C.A., Dovidio, J.F., Brescoll, V.L., Graham, M.J., Handelsman, J.: Science faculty's subtle gender biases favor male students. Proceedings of the National Academy of Sciences **109**(41), 16474–16479 (2012)
15. Nadeem, M., Bethke, A., Reddy, S.: Stereoset: Measuring stereotypical bias in pretrained language models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 5356–5371. Association for Computational Linguistics (2021), https://aclanthology.org/2021.acl-long.416
16. Xu, Z., Huang, R., Chen, C., Wang, X.: Uncovering safety risks of large language models through concept activation vector. Advances in Neural Information Processing Systems **37**, 116743–116782 (2025)
17. Zheng, X., Jiang, J.: An empirical study of memorization in nlp. arXiv preprint arXiv:2203.12171 (2022)
18. Zou, A., Wang, L., et al.: Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:2307.15043 (2023)