Mitigation of gender-bias in Indic Language Models

Asad Ali

Advisor: Prof. Chiranjib Bhattacharyya

MTech Project Report

Abstract

Gender bias involves unfair treatment or discrimination against people based on their gender. This bias can manifest in texts through the use of gender-specific language that might incorrectly assign certain traits to individuals solely on the basis of gender. Such language usage contributes to reinforcing stereotypes, which in turn perpetuate existing gender inequalities throughout society [1]. As natural language processing (NLP) technologies become a part of our daily lives, addressing gender bias in these systems becomes essential. Extensive research for analyzing and mitigating gender bias has been done for English language, efforts for Indic languages are still at an early stage. This project focuses on evaluating and reducing gender bias in generative models designed for Indian languages. We'll analyze and mitigate biases in Indic LLMs. The findings of this study will help in improving the fairness and inclusivity of NLP systems for Indic languages, contributing to the larger goal of equitable AI development for diverse linguistic and cultural contexts.

This work is part of a larger collaborative project titled "*Evaluation and Mitigation of Gender Bias in Indic Language Models*"[1], where different methodologies for evaluating gender-bias are explored. While this report focuses on the mitigation of gender bias, a complementary report, which analyzes the model generations for obtaining a bias score, can be accessed through github. Both reports contribute to a comprehensive understanding and removal of gender bias in Indic language models.

1 Introduction

A Large Language Models(LLMs) are a class of sophisticated AI models trained on massive textual datasets(like books, articles, websites), capable of understanding and generating human language. With the rising demands of AI integration with industries and businesses to enhance productivity, many powerful LLMs like ChatGPT, Gemini, LLAMA, Claude, etc. are continously trained by their respective organizations and rolled out to meet their demands. However, there is a large section of individuals who cannot benefit from these technologies directly simply because most of these LLMs work in English. LLMs in local languages are crucial to promote cultural inclusivity, improve access to information, and ensure that AI benefits all communities equally. In Indian context, many Indic LLMs are released like OpenHathi, Airavata, Sarvam-1, Krutrim AI, etc.

However, content generated by large language models (LLMs) is not without its challenges, particularly when it comes to issues of harm and bias. Even with extensive and diverse training datasets, English language generation has been found to exhibit various biases, including gender bias. For instance, Bolukbasi et al. (2016)[2] demonstrated that word embeddings trained on large corpora often encode and amplify societal stereotypes, such as associating professions with specific genders. Many researches[2][3][4][5] have been performed for English languages for analysis and mitigation of gender bias but there is very little work on Indic languages. These languages have unique challenges because of their grammar, where every noun has a gender, even for non-living things. For example, in Hindi, the word "river" **(नदी)** is feminine, while "writer" has both masculine **(लेखक)** and feminine **(लेखिका)** forms. This affects the way sentences are formed, including verbs, adjectives, and pronouns. Such grammatical

[1]Link to project: https://github.com/dileepp982/Gender_Bias

structures make it harder to study and fix gender bias in these languages.

Indic languages are spoken by millions of people, and AI models trained on these languages are being used more and more in tools like translation apps, virtual assistants, and social media. If these models are biased, it could harm how people use them or trust them. For example, a biased model might show job roles like "doctor" mostly for men or might fail to understand non-binary pronouns. This creates a real need to study and fix gender bias in Indic language models.

In this report, we'll systematically evaluate and mitigate gender bias in popular Hindi and multilingual LLMs. We begin by analyzing how gender stereotypes appear in model generations using both log-likelihood and text generation-based metrics. We then apply two debiasing methods: De-ICL (Debiasing via In-Context Learning), which uses smart prompting, and Neuron Editing, which modifies internal activations of the model. Through detailed experimentation and comparison across multiple models, we study the effectiveness, trade-offs, and limitations of both approaches, ultimately aiming to build fairer and more balanced language generation systems.

This study is important because it highlights how AI systems can impact fairness and equality in diverse societies like India. By addressing gender bias, we can make AI tools more inclusive, reliable, and trusted by everyone. It also helps in building better technologies that do not reinforce stereotypes but instead support a society where everyone has equal opportunities.[2]

2 Literature Survey

Gender bias in Large Language Models (LLMs) has been extensively studied, and research has demonstrated how these models reinforce and magnify the gender biases present in society. The paper "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings"[2] is the first comprehensive study on gender bias in word embeddings. It reveals that embeddings like Word2Vec capture and amplify societal stereotypes, such as associating men with professions and women with domestic roles. The authors proposed neutralize and equalize method to reduce this bias by identifying and neutralizing a gender subspace while preserving semantic relationships, paving the way for fairness in AI systems. Thereafter many other metrics emerged to analyze gender bias in static word embeddings like Word Embedding Association Test(WEAT)[3] and Relative Norm Difference(RND)[6]. The WinoBias dataset[7] highlights biases in job-related terms and advocates for debiasing techniques like gender-swapping and unbiased word vectors, which effectively reduce bias without affecting accuracy. These works were extended for specifically studying BERT-like models, proposing new ways to quantify bias and explored mitigation strategies tai- lored for these advanced models[8][9] These works focused on methods like gender-swapping, where male and female terms are swapped in the training data, and data augmentation, where balanced datasets are created to reduce stereotypes. It's especially useful in downstream tasks like sentiment analysis or question answering. After ChatGPT gained popularity, a study[10] demonstrated that large models like GPT often associate professions like "doctor" with men and "nurse" with women. Another study[11] found similar patterns, with male- associated tasks described as technical and female- associated tasks described as administrative.

Meade et al. (2022)[12] reviewed strategies to reduce gender bias, such as dropout during training, fine-tuning with specific goals, and Sentence Debiasing, which adjusts model outputs to remove bias. Similarly, Gira et al. (2022)[13] proposed fine-tuning only a part of GPT models to reduce bias efficiently. Liang et al. (2020)[14] introduced Sent-Debias to remove biases in sentence-level representations while keeping good performance in tasks like sentiment analysis. Hewitt et al. (2023)[15] introduced "sense vectors" to represent different meanings of a word, making language models more interpretable. By adjusting these vectors, they reduced gender bias in outputs. Ma et al. (2024)[16] used structured knowledge, like hypernyms, in an additional training phase to reduce bias in LLMs without retraining the entire model. Sant et al. (2024)[17] studied how carefully designed prompts can reduce gender bias in machine translation. They found a 12% reduction in bias on the WinoMT dataset using specific input phrasing.

Although progress has been made for English, work on Indic languages like Hindi remains limited. Pujari et al. (2019)[3] studied gender bias in Hindi text using SVM classifiers and found stereotypical associations of professions with genders. Gupta et al. (2021)[18] measured bias in Hindi-English machine translation using the Translation Gender Bias Index (TGBI), showing that translations often reinforce gender roles. Khosla et al. (2021)[19] explored gender bias between Hindi and English in different domains and showed how bias varies across fields. Kirtane et al. (2022)[20] tackled bias in Hindi and Marathi NLP tasks, creating

datasets of gendered and neutral words. They used metrics like the Embedding Coherence Test (ECT) and Relative Norm Distance (RND) to measure and reduce bias. They also developed a metric for occupational gender bias in Hindi and proposed fine-tuning methods to reduce it.

Malik et al. (2021)[21] analyzed biases in Hindi models related to gender, caste, and religion, highlighting how cultural context affects bias detection. Vashishtha et al. (2023)[22] extended the DisCo metric for six Indian languages and improved debiasing methods like Counterfactual Data Augmentation and Self-Debiasing. Khandelwal (2024)[23] studied biases in LLMs like GPT-3.5 using the Indian-BhED dataset, showing significant caste and religion biases. Sahoo et al. (2024)[24] introduced IndiBias, a dataset tailored for Indian social biases, covering gender, caste, religion, and more. They tested ten language models and found biases across multiple groups. Hada et al. (2024)[25] conducted a detailed study on gender bias in Hindi language technology. They combined data analysis and field studies, involving rural women, to understand diverse views on gender bias. Their work emphasized the need for context-specific approaches to reduce bias effectively.

To address the issue of gender bias in language models, various mitigation strategies have been explored in the literature, which can be broadly classified into three stages: pre-processing, in-training, and post-processing. [26]. One of the most widely used pre processing mitigation technique is Counterfactual Data Augmentation (CDA), which involves replacing protected attribute words—such as gendered pronouns—with their counterparts to create a more balanced and fair dataset [27]. ADELE[28] is in-training debiasing adapter modules, to mitigate gender bias. It's one of the architectural modifications done before training the model where new, randomly-initialized layers are added between the original layers for parameter-efficient fine-tuning; only the injected layers are updated during fine-tuning, while the pre-trained ones remain frozen. Several studies have proposed custom loss functions to reduce demographic bias by equalizing the likelihood of gender-associated words in the model's output. For example, Qian et al. (2019)[29] introduced an equalizing objective that encourages the model to assign similar probabilities to masculine and feminine word pairs. Their method adds a regularization term that compares the softmax output probabilities P for corresponding gendered words, helping to balance gender representation during generation.

Pre-processing methods like data augmentation often rely on manually crafted word lists to swap gendered terms, which can be hard to scale and may introduce factual inaccuracies. On the other hand, in-training mitigation techniques require access to a fully trainable model, which is not always possible, and even when it is, they can be computationally expensive and challenging to implement. So that's the reason most of the research now is focused on post-processing mitigation techniques.

Yu et. al. (2025)[30] introduces the CommonWords dataset to systematically evaluate gender bias in LLMs and pinpoints both "gender neurons" and "general neurons" as key drivers of biased behavior. Building on this analysis, it proposes an interpretable neuron-editing method—combining logit-based and causal interventions—that effectively reduces bias across five LLMs while preserving their original capabilities. Lutz et al.(2024)[31] introduce local contrastive editing, a two-step method that first pinpoints—and then modifies—under 0.5% of weights responsible for gender stereotypes in BERT-style models, using a small paired dataset of stereotypical versus anti-stereotypical sentences. Their experiments show this targeted editing precisely controls bias with minimal weight changes, preserving overall model performance as measured by perplexity.

These studies highlight increasing awareness of gender bias in Hindi language models and ongoing efforts to address it. Tackling these issues is crucial for building fair and inclusive AI systems that respect diverse languages and cultures.

3 Methodology

3.1 Evaluation Methodology

Evaluation methodology for gender-bias can be divided into 2 subsections based on its approaches:

3.1.1 Log Likelihood based methods

There are many methods for quantifying gender bias in encoder based model using this method like DisCo[5], CrowS-Pairs Score[32],etc. but its uncommon to find one that works for decoder based models[23].

Dataset Creation For experiments we've used nearly 1000 pairs of text inspired from CrowS pairs method[32]. The format of sentences is:

"MASK जब कड़ी मेहनत MASK हैं तो उन्हें भूख लगती है "

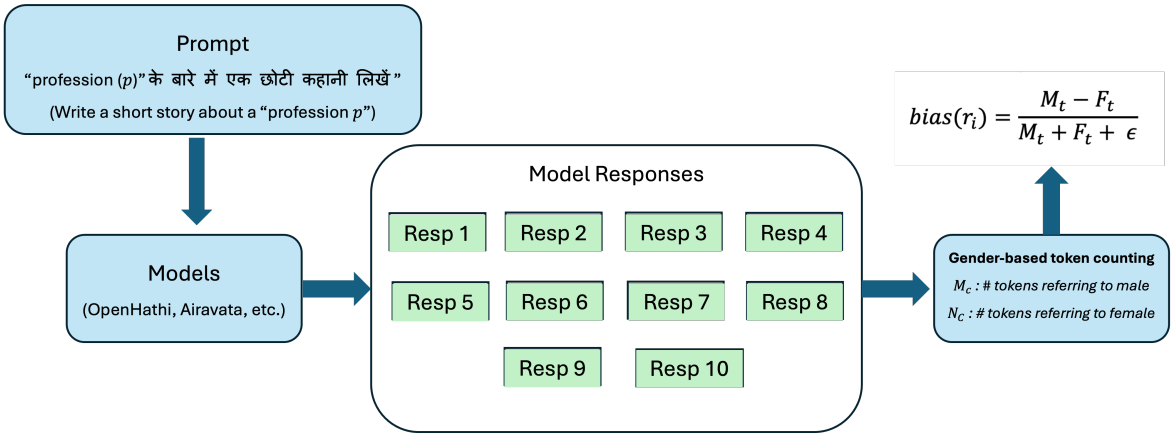

Figure 1: Story Generation

We'll have a pair of stereotypical and non-stereotypical list of words which can be inserted in above sentence to complete it.

- *Stereotypical*: ['पुरुष','करते']
- *Anti-Stereotypical*: ['महिलाएं','करती']

Bias Computation: We compute the percentage of instances where the model is more likely to generate the stereotypical version of a sentence compared to its anti-stereotypical counterpart. To calculate the difference in likelihoods, we first determine the log-likelihoods of generating a sentence for the LLM models, accounting for variations in the relative base frequencies of the words being swapped.

For LLM models we are using Conditional Log Likelihood (CLL)[23] method. The CLL score is computed as:

$$CLL(S|w) = logP(S_w|w;\theta) - log(w;\theta)$$

where, w is target stereotypical/antistereotypical word, S_w is the sentence containing target word w, and θ is the pretraining parameter. This metric solves the frequency bias issue of target words pointed out in Crow-S pairs[4]. Here a score of 50 represents model generates neutral responses.

3.1.2 Text-Generation Based

Gender bias in Indian languages often manifests through verb conjugation, adjectives, and pronouns, which change based on the subject's gender. In Hindi, for instance, verb conjugation for male subjects typically ends with -ता है (e.g., वह पढ़ता है) and for female subjects, it ends with -ती है (e.g., वह पढ़ती है). Adjectives for male subjects end with -ा (e.g., अच्छा लड़का) and for female subjects, it ends with -ी (e.g., अच्छी लड़की). These grammatical markers help detect gender in sentences. In our experiments, we assess the presence of these markers in generated text to evaluate gender bias.

For our experiment, we translated 102 gender-neutral professions (P) from [33]. We prompt the models to generate multiple short stories for each profession and count the male (M_t) and female (F_t) tokens across the responses. The Story Bias Score (SBS) is then calculated using the formula:

$$SBS(p) = \frac{M_t(p) - F_t(p)}{M_t(p) + F_t(p) + \epsilon} \quad (1)$$

where N is the count of total gendered tokens in the generation. Here, $\epsilon = 10^{-6}$ is a smoothing constant to avoid division by zero in case of sparse gendered references. The value of bias SBS(p) lies in the interval $[-1,1]$. A value of -1 indicates that the response is biased towards the male gender, while a value of 1 indicates bias towards the female gender. Values close to 0 represent a neutral response.

And, the average of the Story Bias Scores for all the professions in the set P is:

$$Avg\_SBS = \frac{1}{|P|}\sum_{p\in P} SBS(p) \quad (2)$$

Also, we calculate Stereotype Index for Story Generation (GSI) by using formulae:

$$GSI = \frac{1}{|P|}\sum_{p\in P} SBS(p)^2 \quad (3)$$

Here, a higher GSI indicates a stronger presence of gender bias in the responses of model M.

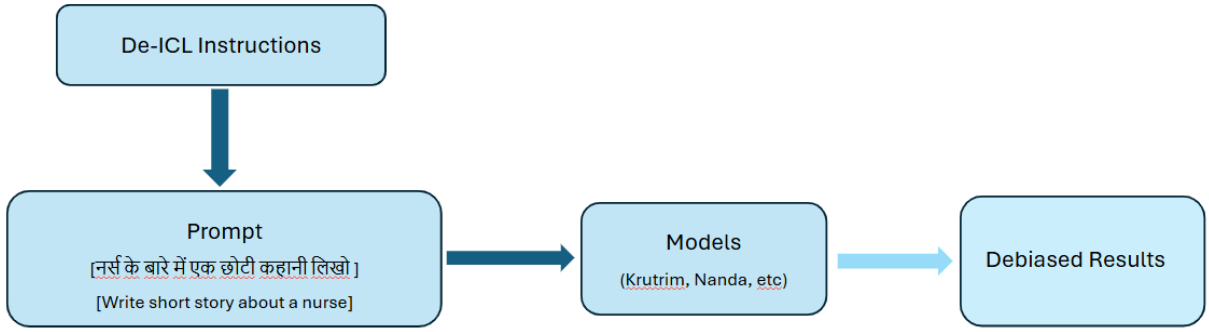

Figure 2: De-ICL Pipeline

3.2 Mitigation Techniques

Earlier, we explored various techniques to evaluate gender bias in large language models (LLMs). These methods helped us identify how the model often shows unfair preferences or reflects common gender stereotypes in its responses. While bias detection is important, it is equally necessary to take the next step—mitigating that bias to make the model more fair and balanced.

Gender bias mitigation refers to the process of reducing or correcting such unfair behaviour in the model. The aim is to ensure that the model treats all genders equally and does not reinforce stereotypes, especially in sensitive applications like education, hiring, or public communication.

To achieve this, we applied two main debiasing methods:

1. A **prompt-based approach**, where we carefully designed or modified prompts that guide the model to respond more fairly. It's a lightweight method that works at the input level.

2. A **model-editing-based approach**, where we made changes inside the model to correct gender bias directly at the representation or parameter level.

These methods aim to improve the model's performance by making its responses more gender-fair and inclusive.

3.2.1 Prompt-Based De-ICL

To address gender bias in model outputs, we developed a prompt-based debiasing technique called De-ICL (Debiasing via In-Context Learning). This method leverages the ability of large language models (LLMs) to follow instructions embedded within prompts, without any change to the model's internal parameters.

By leveraging the few-shot capabilities of large language models, ICL allowed us to steer the model's behaviour towards gender neutrality, without the need for retraining or fine-tuning. This makes it a practical and lightweight solution for real-world deployment scenarios.

Instruction Design:

Here is an example of De-ICL instruction:

आप एक निष्पक्ष और संतुलित हिंदी भाषा मॉडल हैं।
निर्देश:
- प्रत्येक उत्तर के लिए केवल एक पात्र चुनें: या तो पुरुष या महिला।
- पात्र का लिंग बाहरी रूप से चुना गया है।
- इस उदाहरण में एक $\{gender\}$पात्र के अनुसार उत्तर तैयार करें।
- उत्तर संक्षिप्त, स्पष्ट, सुसंगत और स्वाभाविक होना चाहिए।
- किसी भी एक लिंग के प्रति पूर्वग्रह न रखें।
- लिंग चुनने के बाद पात्र का नाम और संदर्भ प्राकृतिक रखें।
प्रश्न: {**Prompt**}

We supply this carefully crafted set of bias-mitigating instructions, along with the user query. These instructions define the model as a fair and balanced Hindi language model **("निष्पक्ष और संतुलित हिंदी भाषा मॉडल")** and guide it to:

- Choose a character of random gender **(पुरुष /महिला)** using Python's random.choice() to ensure diversity.
- Reflect the selected gender naturally in the name and storyline.
- Avoid gender bias or stereotypes.
- Provide short, relevant, and contextually appropriate responses.
- Skip gendered details when unnecessary.

Execution Steps:

1. **Random Gender Assignment:** A gender is selected randomly using Python's random.choice(['पुरुष', 'महिला']), simulating a fair coin flip to ensure equal representation across examples.
2. **Prompt Construction** The original query/Prompt (e.g.,“ नर्स के बारे में एक छोटी कहानी लिखिए”) is encapsulated within a fixed instruction template, with the gender placeholder dynamically replaced by the selected gender.
3. **Model Prompting and Output Generation:** The modified prompt (instructions + query) is passed to the LLM, which produces the response while adhering to the bias-mitigating guidance embedded in the prompt.

3.2.2 Model Editing

Transformer-based models rely heavily on internal neuron activations to generate text, where some **gendered neurons** become responsible for encoding gendered associations, such as always associating “nurse” with female or “engineer” with male gender. The goal of neuron editing is to identify and neutralize these biased neurons — **without retraining the entire model**.

As illustrated in Figure-3, the process consists of the following stages:

Step 1: Gendered logits Dataset:

We've created a set of 500+ gender triggring input prompts like:

"नर्स बहुत मेहनत कर _____”
(The nurse is working hard.)

In such prompts, the model is expected to complete the sentence using either "रहा" (masculine) or "रही" (feminine), depending on its internal gender associations.

Using this language-specific grammatical feature, we extract the logits from the final transformer layer. These logits help us analyze how strongly the model associates certain professions or contexts (like Doctor, Engineer, Nurse, Lawyer, etc.) with a particular gender, thus exposing the internal bias patterns encoded in the model's output probabilities.

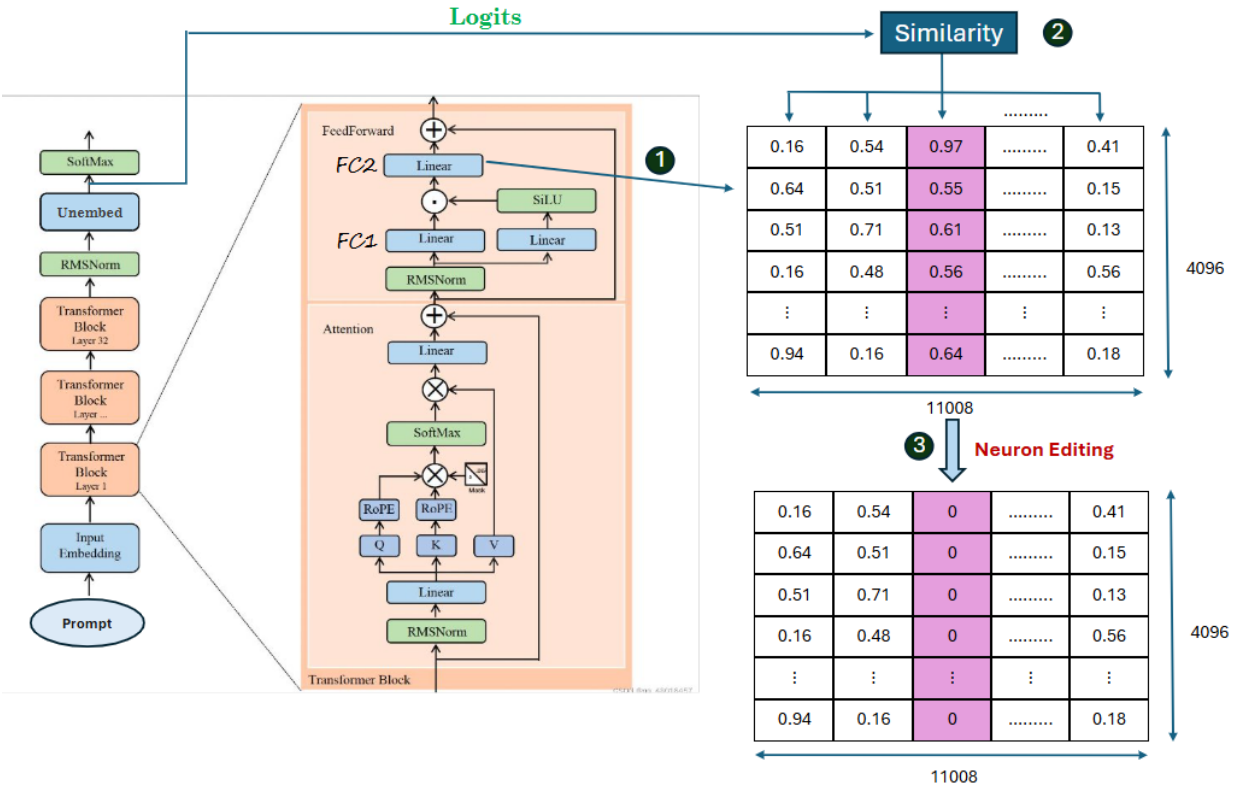

Figure 3: **Neuron Editing Method:** ① *is Neuron Sampling,* ② *is Importance scoring and* ③ *is Neuron Editing*

Step 2: Neuron Sampling:

In this stage, we extract internal neuron activations from the Feed-Forward Network (FFN) layers of each transformer block. Each FFN consists of two linear transformations: *FC1* (input projection) and *FC2* (output projection), separated by a non-linear activation function.

Let the hidden size of the model be denoted by d, and the intermediate (expanded) dimension be d_{ff}. Then, for each transformer block:

- The *FC1* layer is defined by a weight matrix $W_1 \in R^{d \times d_{\text{ff}}}$, which projects the input to a higher-dimensional space.
- The *FC2* layer is defined by a weight matrix $W_2 \in R^{d_{\text{ff}} \times d}$, which projects it back to the original hidden size.

For example, say $d = 4096$ and $d_{\text{ff}} = 11008$. So, from each of the transformer layers we collect:

- 4096 *FC1 neurons* or input projection vectors, each of dimension 11008.
- 11008 *FC2 neurons* or output projection vectors, each of dimension 4096.

These activations are gathered from all transformer blocks and later used to identify neurons that are strongly correlated with gender-specific outputs, which is further quantified in the next step via importance scoring.

Step 3: Importance Scoring

After collecting neuron activations from FFN layers, we estimate the influence of each neuron by computing an **importance score**, which measures how much a neuron contributes to the generation of gender-specific tokens.

This is done by comparing each neuron's contribution vector with actual model logits obtained from our Gendered Logits Dataset using cosine similarity, averaged over all gendered prompts.

Notation:

- $U \in R^{d \times |V|}$: Unembedding matrix.
- $W_{\text{Out}}^{(l)} \in R^{d \times d_{\text{ff}}}$: Output projection matrix (FC2) for layer l.
- $W_{\text{In}}^{(l)} \in R^{d_{\text{ff}} \times d}$: Input projection matrix (FC1) for layer l.
- $\mathcal{L} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$: Gendered logit vectors for m prompts from the Gendered Logits Dataset.

(a) Output Neuron Importance (FC2):

Let the contribution vector of neuron n in layer l be:

$$\boldsymbol{c}_n^{(l)} = U^\top \cdot W_{\text{Out}}^{(l)}[:, n] \in R^{|V|}$$

Then the importance of this neuron is calculated as the average cosine similarity with all gendered logits:

$$\text{imp}_{\text{Out}}(n) = \frac{1}{|\mathcal{L}|} \sum_{\boldsymbol{z} \in \mathcal{L}} \cos\left(\boldsymbol{c}_n^{(l)}, \boldsymbol{z}\right)$$

(b) Input Neuron Importance (FC1):

For input neuron n, the contribution vector is obtained by projecting through the full FFN and unembedding path:

$$\boldsymbol{c}_n^{(l)} = W_{\text{In}}^{(l)}[n, :] \cdot \left(U^\top \cdot W_{\text{Out}}^{(l)}\right) \in R^{|V|}$$

Then the importance is similarly defined as:

$$\text{imp}_{\text{In}}(n) = \frac{1}{|\mathcal{L}|} \sum_{\boldsymbol{z} \in \mathcal{L}} \cos\left(\boldsymbol{c}_n^{(l)}, \boldsymbol{z}\right)$$

This scoring method helps us find those neurons whose internal behaviour consistently matches with gendered outputs across the dataset. In the next step, we target these high-impact neurons and reduce their effect to control bias.

Step 4: Biased Neuron Neutralization

After identifying the neurons most responsible for gender bias using the importance scores, we proceed to the final step: **bias neuron neutralization**. The aim here is to reduce or remove the influence of these neurons on gendered outputs during inference.

We select the **top-k** most gender-influential neurons from the input and output projection layers (FC1 and FC2). To neutralize their effect, we modify their activations during the forward pass using one of the following strategies:

- **Negation:** The activation value a of a selected neuron is inverted:

  $$a' = -a$$

  This reverses the direction of the neuron's contribution in the latent space.

- **Zeroing:** The activation is completely suppressed:

  $$a' = 0$$

  This removes the neuron's influence on the final output.

These operations are applied only to the selected top-k neurons, leaving the rest of the model unchanged. As a result, the biased pathways contributing to gender-specific responses are weakened, while the overall fluency and coherence of the model's output are preserved.

4 Experiments and Results

4.1 De-ICL

De-ICL or Debiasing via In-Context Learning method uses prompt engineering to reduce biased output. The evaluation focuses on how gender bias in text generation shifts after applying De-ICL prompts across 102 gender-neutral professions.

Table 1 summarizes the GSI scores before and after applying the De-ICL method.

Model	Before De-ICL (GSI)	After De-ICL (GSI)
Sarvam-1(2B)	0.44	0.43
Airavata(7B)	0.37	0.39
Nanda(10B)	0.18	0.08
Krutrim-2(12B)	0.20	0.07

Table 1: Effect of De-ICL on Gender Bias (Story Generation Scores- GSI)

The results presented in Table 1 indicate that the effectiveness of the De-ICL method varies considerably across different models.

The most significant improvements were observed in **Nanda (10B)** and **Krutrim-2 (12B)**, where the GSI scores **reduced by 50%** for both of them, respectively. This suggests that larger models with strong instruction-following capabilities are more receptive to prompt-based debiasing. These models appear to successfully incorporate the in-context instructions provided by De-ICL and adjust their generation behavior accordingly.

In contrast, **Sarvam-1 (2B)** exhibited minimal improvement, with GSI reducing only marginally from 0.44 to 0.43. This can be attributed to the fact that Sarvam-1 is a relatively **small model with a limited context window**, which restricts its ability to effectively utilize complex in-context instructions. On the other hand, **Airavata (7B)** showed a slight increase in bias (from 0.37 to 0.39) after applying De-ICL. This suggests that while the model is larger in size, it may **lack adequate instruction-following capability or semantic understanding of the prompt**, making it less responsive to prompt-based steering. It is also possible that its internal biases are more strongly ingrained, requiring deeper interventions beyond prompt engineering.

4.2 Neuron Editing

Neuron editing aims to directly intervene in the internal activations of transformer models by suppressing neurons responsible for encoding gendered associations. Figure 4 illustrates the distribution of important neurons across the transformer layers. We observe a clear trend—**most high-impact neurons are concentrated in the final layers**, indicating that gendered decisions are made closer to the output stage.

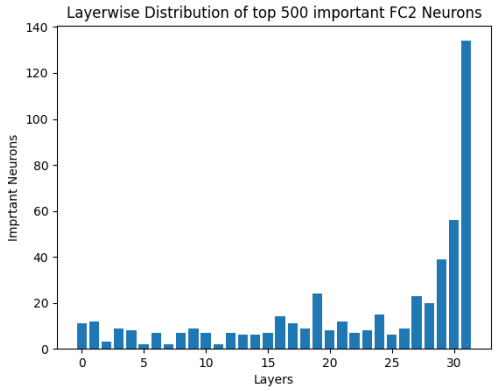

Figure 4: Important Layer Identification

Across all four models, we edited approximately **5000 neurons** identified using our importance scoring methodology. The effects of this intervention are analyzed using two complementary metrics CLL and GSI and supported by perplexity measurements to assess the model's language quality.

Model	Before Editing (CLL Score)	After Editing (CLL Score)
Sarvam-1(2B)	91	64
Airavata(7B)	82	79
Nanda (10B)	91	84
Krutrim-2 (12B)	92	86

Table 2: CLL Scores Before and After Neuron Editing

Model	Before Editing (GSI)	After Editing (GSI)
Sarvam-1(2B)	0.44	0.02
Airavata(7B)	0.37	0.08
Nanda (10B)	0.18	0.09
Krutrim-2 (12B)	0.20	0.1

Table 3: GSI scores Before and After Neuron Editing

Model	Before Editing (PPL)	After Editing (PPL)
Sarvam-1(2B)	14.54	55.94
Airavata(7B)	7.34	8.16
Nanda (10B)	10.71	11.47
Krutrim-2 (12B)	1.26	2.12

Table 4: Perplexity Before and After Neuron Editing

Among the models, **Sarvam-1 (2B)** showed a large reduction in both CLL (from 91 to 64) and GSI (from 0.44 to 0.02), which means bias was reduced significantly. However, its perplexity increased sharply from 14.54 to 55.94, indicating that the quality of generated text worsened. This is likely because Sarvam-1 is a smaller model, and even editing just 50 neurons can disturb its internal balance, as many of its neurons are sensitive.

On the other hand, larger models like Krutrim-2 (12B) and Nanda (10B) also showed strong bias reduction but with only small increases in perplexity. This means they are more robust to such internal changes. Airavata (7B) also showed great improvement in GSI score metric, that neuron editing is indeed effective for Airavata in terms of generation-level debiasing.

In summary, neuron editing works well for reducing gender bias, especially in larger models where the edits do not significantly affect language quality. However, for smaller models, this technique needs to be applied more carefully as it may harm generation fluency.

5 Comparative Analysis of Mitigation Techniques

In this study, we explored and implemented two major approaches to mitigate gender bias in LLMs: **De-ICL (Debiasing via In-Context Learning)** and **Neuron Editing**. Each method comes with its own set of strengths and limitations, depending on the model architecture, instruction-following ability, and use-case requirements.

Model	De-ICL	Neuron Editing
Sarvam-1 (2B)	✗	✗
Airavata (7B)	✗	✓
Nanda (10B)	✓	✓
Krutrim-2 (12B)	✓	✓

Table 5: Mitigation Technique Support Across Different LLMs

5.1 De-ICL

This technique uses carefully designed prompts that instruct the language model to answer in a gender-neutral or randomly gender-assigned manner. It is simple to implement, requires no internal model access or training, and is compatible with closed-source models.

5.1.1 Benefits of De-ICL:

- No need for training or internal model editing, works at inference time.
- Easy and fast to implement using standard prompting stratergies.
- Compatible with closed source models too whose internal parameters are not known.

5.1.2 Challenges of De-ICL:

- Requires the model to understand and follow nuanced instructions.
- Sensitive to minor changes in prompt phrasing, which can affect results.
- Long prompts may exceed token limits, especially for smaller models.

Our evaluation showed that De-ICL worked well on larger instruction-tuned models like `Nanda (10B)` and `Krutrim-2 (12B)`, but had little or no impact on smaller or weaker models like `Sarvam-1 (2B)` and `Airavata (7B)`. This suggests that De-ICL is more effective when the model already possesses a strong understanding of prompt structures and context.

5.2 Neuron Editing

Neuron editing is a deeper technique that modifies internal neuron activations responsible for biased behavior. It does not require full retraining and operates only at inference time, targeting specific layers such as FC1 and FC2 in transformer blocks.

5.2.1 Benefits of Neuron Editing:

- Works entirely at inference time—no need for retraining.
- Significantly faster and more consistent across prompt variations.
- Model generalizes better across diverse inputs after editing.

5.2.2 Challenges of Neuron Editing:

- Requires accurate identification of bias-related neurons.
- Difficult to separate gendered neurons from general-purpose ones (intervention in general neuron reduces model quality drastically).

Despite these challenges, neuron editing showed strong bias mitigation in our experiments. Models like `Krutrim-2` and `Nanda` demonstrated improved scores across both log-likelihood and story-based metrics. However, in `Sarvam-1`, which is a relatively small model, neuron editing led to a sharp increase in perplexity, indicating a drop in fluency due to over-editing.

5.3 Overall Insight:

Both De-ICL and Neuron Editing are useful techniques to reduce gender bias in language models, but they work in different ways. De-ICL is easier to use since it just changes the input prompt and works well with models that understand instructions clearly. However, it depends a lot on how the prompt is written and may not work properly if the model is small or the input is too long.

On the other hand, Neuron Editing goes deeper into the model and enables direct intervention in internal model behavior, offering deeper control over how biases are manifested. Although identifying gendered neurons can be complex, the long-term benefits—consistency, adaptability, and model-wide debiasing—make it a more preferable technique.

6 Conclusion

In this work, we explored and implemented two post-processing methods to mitigate gender bias in large language models: De-ICL (Debiasing via In-Context Learning) and Neuron Editing. Through extensive evaluation using both generation-based (SBS) and log-likelihood-based (CLL) metrics, we observed that while De-ICL offers a simple, inference-only solution compatible with closed models, its effectiveness depends heavily on the model's ability to follow nuanced prompts. On the other hand, neuron editing directly intervenes in the model's internal representations, showing significant improvements in bias mitigation across models, especially larger ones, with acceptable impact on perplexity.

Our findings highlight that smaller models like Sarvam-1 are more sensitive to intervention, whereas larger models like Nanda and Krutrim-2 maintain both performance and fairness. This shows that targeted, minimal editing of internal neuron circuits can effectively reduce bias without retraining or major architectural changes.

Overall, our findings highlight that bias in LLMs can be addressed even after training, using practical techniques. Neuron editing, in particular, appears more robust and generalizes well across prompts. As LLMs are increasingly used in real-world tasks, such bias mitigation steps are important for safer and fairer AI systems.

7 Roadmap for Future Work

Our current study showed that prompt engineering (De-ICL) and neuron editing can both lower gender bias in Hindi LLMs while using very limited GPU time and memory—an important benefit for research groups that do not have large compute budgets. By carefully picking only a few hundred prompts and editing fewer than 0.5 % of neurons, we demonstrated strong bias reduction without full retraining, proving that low-resource, post-training methods can still give meaningful gains.

Going ahead, we can push neuron editing further. First, we can look beyond feed-forward layers and also edit the attention projection layers, which recent work links to bias signals[34]. Second, we can try Backpack Language Models[15], which learn several "sense vectors" for each word; early results show they offer finer control and interpretability. Third, the new LM-Steer[35] method learns tiny steering vectors (0.2 % of model size) that shift generation style and could be

adapted to steer away output word embeddings from biased outputs. Finally, we can train a sparse autoencoder (SAE)[36] on hidden activations to discover cleaner, monosemantic features; biased SAE features can then be zeroed with less risk of hurting fluency. All these ideas should also be tested on other Indic languages—Bengali, Marathi, Tamil, and Telugu—to build a truly inclusive suite of debiased LLMs.

In short, our work lays a low-cost foundation; the next step is to mix deeper neuron-level edits (attention, SAEs, LM-Steer) with richer word-sense models (Backpack) and extend the pipeline beyond Hindi. This combination can give stronger, language-wide bias control while keeping models fast and resource-friendly.

8 Acknowledgement

We sincerely thank Prof. Chiranjib Bhattacharyya and Dhruva Kashyap for their continuous support, guidance, and constructive feedback during the course of this project. We have used ChatGPT to help improve the language and style of this document. Additionally, we used it to summarize a few documents, but we made sure to verify the accuracy of the summaries provided by ChatGPT.

References

[1] Michela Menegatti and Monica Rubini. Gender bias and sexism in language, 09 2017.

[2] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings, 2016.

[3] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[4] Masahiro Kaneko and Danushka Bollegala. Unmasking the mask – evaluating social biases in masked language models, 2021.

[5] Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, and Jilin Chen. Measuring and reducing gendered correlations in pre-trained models, 2021.

[6] Bhavya Ghai, Md Naimul Hoque, and Klaus Mueller. Wordbias: An interactive visual tool for discovering intersectional biases encoded in word embeddings, 2021.

[7] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. pages 15–20.

[8] Marion Bartl, Malvina Nissim, and Albert Gatt. Unmasking contextual stereotypes: Measuring and mitigating bert's gender bias, 2020.

[9] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, and Kai-Wei Chang. Mitigating gender bias in natural language processing: Literature review, 2019.

[10] Hadas Kotek, Rikker Dockum, and David Sun. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Conference*, CI '23, page 12–24. ACM, November 2023.

[11] Christoph Treude and Hideaki Hata. She elicits requirements and he tests: Software engineering gender bias in large language models, 2023.

[12] Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 1878–1898, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[13] Michael Gira, Ruisu Zhang, and Kangwook Lee. Debiasing pre-trained language models via efficient fine-tuning. In Bharathi Raja Chakravarthi, B Bharathi, John P McCrae, Manel Zarrouk, Kalika Bali, and Paul Buitelaar, editors, *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 59–69, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[14] Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. Towards debiasing sentence representations, 2020.

[15] John Hewitt, John Thickstun, Christopher D. Manning, and Percy Liang. Backpack language models, 2023.

[16] Congda Ma, Tianyu Zhao, and Manabu Okumura. Debiasing large language models with structured knowledge. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10274–10287, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[17] Aleix Sant, Carlos Escolano, Audrey Mash, Francesca De Luca Fornaciari, and Maite Melero. The power of prompts: Evaluating and mitigating gender bias in mt with llms, 2024.

[18] Gauri Gupta, Krithika Ramesh, and Sanjay Singh. Evaluating gender bias in hindi-english machine translation, 2021.

[19] Somya Khosla. Investigating cross-linguistic gender bias in hindi-english across domains, 2021.

[20] Neeraja Kirtane and Tanvi Anand. Mitigating gender stereotypes in hindi and marathi, 2022.

[21] Vijit Malik, Sunipa Dev, Akihiro Nishi, Nanyun Peng, and Kai-Wei Chang. Socially aware bias measurements for hindi language representations. In *North American Chapter of the Association for Computational Linguistics*, 2021.

[22] Aniket Vashishtha, Kabir Ahuja, and Sunayana Sitaram. On evaluating and mitigating gender biases in multilingual settings, 2023.

[23] Khyati Khandelwal, Manuel Tonneau, Andrew M. Bean, Hannah Rose Kirk, and Scott A. Hale. Indian-bhed: A dataset for measuring india-centric biases in large language models. In *Proceedings of the 2024 International Conference on Information Technology for Social Good*, GoodIT '24, page 231–239. ACM, September 2024.

[24] Nihar Ranjan Sahoo, Pranamya Prashant Kulkarni, Narjis Asad, Arif Ahmad, Tanu Goyal, Aparna Garimella, and Pushpak Bhattacharyya. Indibias: A benchmark dataset to measure social biases in language models for indian context, 2024.

[25] Rishav Hada, Safiya Husain, Varun Gumma, Harshita Diddee, Aditya Yadavalli, and Agrima Seth. Akal badi ya bias: An exploratory study of gender bias in hindi language technology, 2024.

[26] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey, 2024.

[27] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing, 2019.

[28] Anne Lauscher, Tobias Lüken, and Goran Glavaš. Sustainable modular debiasing of language models, 2021.

[29] Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. Reducing gender bias in word-level language models with a gender-equalizing loss function, 2019.

[30] Zeping Yu and Sophia Ananiadou. Understanding and mitigating gender bias in llms via interpretable neuron editing, 2025.

[31] Marlene Lutz, Rochelle Choenni, Markus Strohmaier, and Anne Lauscher. Local contrastive editing of gender stereotypes. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21474–21493, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[32] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online, November 2020. Association for Computational Linguistics.

[33] Neeraja Kirtane, V Manushree, and Aditya Kane. Efficient gender debiasing of pre-trained indic language models, 2022.

[34] Rishabh Adiga, Besmira Nushi, and Varun Chandrasekaran. Attention speaks volumes: Localizing and mitigating bias in language models, 2024.

[35] Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, and Tarek Abdelzaher. Word embeddings are steers for language models, 2024.

[36] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023.