Resolving Gender Biases in LLMs at Inference Time with Novel Dijkstra's-based K-Explorers Neural Network Traversal (KeNNT)

Hanav Modasiya
Santa Clara High School
California, 95051, United States of America
hanavmw13@gmail.com

Abstract

The vast growth of Large Language Models (LLMs) has increased the need for larger data corpora, and researchers often turn to the internet for a source of that data. However, with rising online sexism, LLMs start to pick up on gender biases in the text they generate. Despite protective measures, biases still infiltrate newer models like ChatGPT and LLaMA 2. In this research, we introduce a novel Dijkstra's-based algorithm called K-explorers Neural Network Traversal (KeNNT), that we hypothesize can be attached to models and algorithms to solve optimization problems. KeNNT is a novel method to guide Transformer models away from generating gender biases. KeNNT, based on Dijkstra's shortest path algorithm, was tested on a GPT-2 model fine-tuned on the WinoBias benchmark dataset. KeNNT reduced gender bias in generated texts by 84.79% (K = 3) and 95.93% (K = 4), outperforming some industry standards. Based on the promising results, it is hypothesized that KeNNT can enhance other optimization algorithms, such as Gradient Descent, improving accuracy and avoiding local minima convergence. With this work, we hope to inspire further, novel endeavors into gender bias resolution and new perspectives on optimization problems.

1 Introduction

Large Language Models (LLMs), built on the Transformer architecture, have become increasingly popular as they open up a revolutionary field of human and Artificial Intelligence (AI) interaction (Chang et al., 2024) that is exemplified by some of the most vast and emerging technologies of the time, such as multimodal, conversational language models like ChatGPT and Gemini or models tailored to code understanding and completion, like LLaMA 2 and BLOOM (Li et al., 2023). These large models, built upon billions of parameters, are trained on large corpora of data from around the internet, where they are often prone to stereotype or bias infiltration. Similar models have seen unprecedented amounts of bias, most commonly in gender and race (Dong et al., 2024; Li et al., 2024). They tend to create hurtful text or negative representations of particular demographics, while other counterpart demographics do not show those negative representations. Sometimes, the bias is quite subtle, as the United Nations Educational, Scientific and Cultural Organization (UNESCO) report *Bias Against Women and Girls in Large Language Models* in early 2024 showed that LLMs still relate females with domestic terms such as family and children while relating males with technical terms such as executive and business even when given the same context (ClareO'Hagan, 2024). UNESCO also revealed that these subtle biases are still common in larger models such as ChatGPT 3.5 and LLaMA 2, which are used globally today, highlighting an immense crisis as these biases "have the power to subtly shape the perceptions of millions of people," as noted by UNESCO Director (ClareO'Hagan, 2024). These biases, specifically gender biases, are not an issue to overlook as they cause urgent deterioration to the growth of AI and its integration into the world (Dong et al., 2024; Bolukbasi et al., 2016; Alba, 2022). This research introduces and validates a new Artificial Intelligence algorithm for resolving these subtle gender biases and possibly other search algorithm issues during inference time (while the model is running) called K-Explorers Neural Network Traversal, or KeNNT. In simple terms, the algorithm generalizes as this: when exploring a network of choices, if a search algorithm is unsure about its next action or choice, rather than taking a risk and pursuing one singular path, the search algorithm branches off into K different exploration paths.

2 Background

Current methods for resolving these subtle gender biases alter or augment training data to prelimi-

narily remove biases from the Transformer's understanding. (Dong et al., 2024; Li et al., 2024; Bolukbasi et al., 2016; Thakur et al., 2023). However, such methods can sometimes alter important context embedded into the training data, effectively harming the model's accuracy but maintaining its runtime (Bolukbasi et al., 2016; Thakur et al., 2023). Furthermore, such methods are restricting the knowledge of the model itself. It is akin to a teacher not teaching the true history of certain matters because the information is too strong. However, in doing so, the student never truly understands history. By manipulating and censoring data from the Transformer model's learning, the Transformer can often lose a complete understanding of semantic relationships (Bolukbasi et al., 2016; Thakur et al., 2023). Rather, the student should be taught the complete history but cautioned about it and encouraged to learn from it. Likewise, the theorized algorithm in our work, KeNNT, does not alter the training data but steers the Transformer away from biases during inference time, so there is no loss of true understanding.

This work uses coreference resolution, an algorithm that calculates grammatical hierarchies in a sentence using graph representations of grammatical relationships, Natural Language Processing (NLP), and graph algorithms (Lee et al., 2017; Chen et al., 2021). Coreference resolution is used to calculate noun-pronoun clusters, which are groups of nouns and pronouns connected to the same entities (Chen et al., 2021) with the coreference resolution algorithm. Sample noun-pronoun pairings are seen in Figure 1.

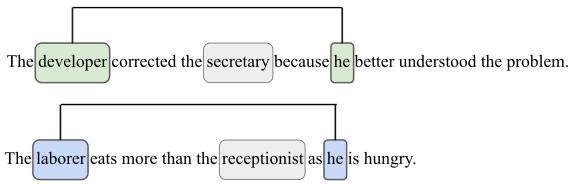


Figure 1: Nouns are linked with correlated pronouns

KeNNT is based on Dijkstra's Algorithm. Dijkstra's Algorithm searches for the shortest path from one node to all other nodes in a weighted graph through a greedy process (Fan and Shi, 2010; Solka et al., 1995). Dijkstra's has been used in machine learning processes before but is primarily used as a backbone for reinforcement learning processes or adversarial networks (Liu et al., 2020). Its integration and motivation are further elaborated in section 4.

3 Problem Framing

Throughout this paper, the traditional view of the Transformer model, and gender biases is altered. The research depicts the Transformer model as a **search algorithm** since it auto-regressively searches for the next best token at inference time (Chang et al., 2024). Specific to this research, gender bias is considered as a case of local minima convergence (Mishra, 2018). Local minima convergence occurs when a search algorithm finds itself optimizing towards the best solution in a segment of the complete solution space, seen in Gradient Descent where if the gradient traverser explores a hole that is not the deepest, it will never be able to find the global minimum output (Jentzen and Riekert, 2022; Mishra, 2018). Since a Transformer model uses the text it has already generated as context for the next token it generates, once there is already bias in the context, it is challenging for the model to climb out of the bias hole, leading to continued bias. Note that this is how our research frames the gender bias problem in LLMs and this view can vary amongst other research. KeNNT aims to solve this issue because the general methods to avoid local minima convergence in search algorithms do not apply well to gender debiasing, regardless of the problem framing. For example, in Gradient Descent, the current methods of improvement are random restarts, momentum optimizations, and noisy optimizations (Mishra, 2018). However, these optimizations do not have much impact on gender bias resolution directly, motivating the need for a new approach towards gender bias resolution: KeNNT.

Next, the Transformer's search space is framed as a neural network. Since the Transformer searches through its vocabulary during every inference step for the most probable next token (Chang et al., 2024), the search space can be viewed as a dense graph where the Transformer creates a smooth line from the first column of the graph to the last. This idea is seen in Figure 2. Let's motivate the concept of KeNNT through intuition. If you were at a junction of dark tunnels—one leading to a prize and the rest to a consequence—instead of taking a risk and going into one, you can send explorers to explore each tunnel for you and then follow only the successful one. Likewise, the Transformer with KeNNT will split up into K variations at certain junctions, creating a lightning shape com-

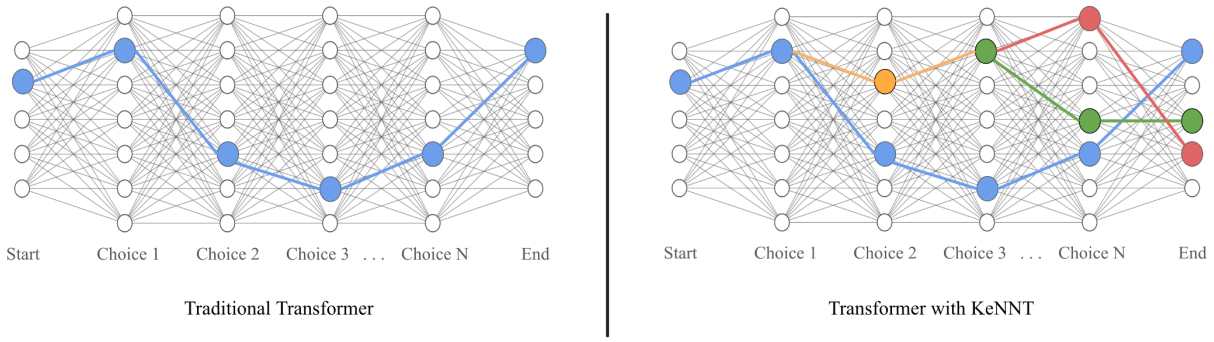


Figure 2: Comparison of Traditional Transformer and Transformer with KeNNT in the problem framing

pared to the straight line. Remember that each node in the graph is a certain token so splitting would mean choosing to use a different word at a certain location (elaborated in section 4).

4 Algorithm Overview and Design

Note that KeNNT is not a new Transformer model or any form of a Transformer model. Refer to the following flowchart in Figure 3.

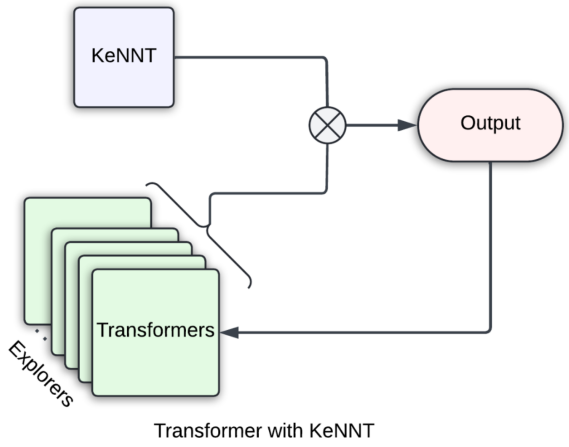


Figure 3: Relationship between KeNNT and Transformer

KeNNT is rather an algorithm that can attach to an AI search algorithm, or here, a Transformer model, and its purpose is to steer the attached model away from generating a biased answer or falling into a local minimum. This interdependence is visualized in the flowchart.

The Transformer with KeNNT design uses the same Transformer model that a traditional Transformer loop would use to generate text, but it stores multiple instances of different contexts (generated sentences). The distinct contexts are called explorers as they each explore a different path of the neural network of decisions. Each explorer represents a different generated text, so it is a different context for the Transformer, which is essentially the same as each explorer being a Transformer itself. KeNNT does not generate output texts, rather, KeNNT helps guide an explorer through its output possibilities at every step. This may seem counterintuitive since the idea is that the Transformer model should be able to guide itself through search space. However, the Transformer is not guiding itself with a goal of gender bias reduction whereas that is KeNNT's goal. Thus incorporating KeNNT with the Transformer effectively allows the model to optimize grammatical accuracy and gender bias mitigation. KeNNT is not trained and does not learn information, while the Transformer model does.

At any given time before the algorithm finishes executing, there is a set of activated explorers. The idea is that when a current explorer of the graph feels unsure about the next node it should traverse to, it will branch into K explorers (expansion)—hence the name—that follow K new, distinct paths. This creates two parameters: K and the expansion criteria. Essentially, it avoids taking a risk that could lead to the local minima convergence framed in section 3. Let's explain why Dijkstra's algorithm is used here. We can now have numerous paths toward the end token, but we want to choose the path that has the least bias. Instead of trying all the paths, which would make the runtime grow exponentially at a rate of K, we can use a shortest path-finding algorithm. So, the algorithm calculates a score for each explorer, which is its total bias, and concurrently runs Dijkstra's algorithm to determine which explorer to process next.

KeNNT only processes the best-scoring (lowest bias score) explorer of the set of explorers until one of them reaches the end token (end layer of the graph). Each time an explorer is processed (one more token is generated) its score is updated, allowing multiple distinct explorers to be processed throughout. The metric used to evaluate the bias of an explorer is discussed in the exploration choice section in section 4.1. Due to this, KeNNT processes a drastically smaller amount of explorers, reducing the runtime. This fact is seen in the results in sections 5.4 and 5.5.

4.1 Pseudocode

Exploration Choice: The exploration choice is parallel with Dijkstra's algorithm as it focuses on finding the best path from start to end with a maximal score while simultaneously reducing the runtime drastically. A scoring evaluation method is required to quantify the bias of an explorer so that KeNNT can choose the best explorer to traverse all current explorers. In the case of gender biases, the scoring metric is the intensity of gender polarity of a generated text, or simply, how gender-biased it is. This scoring evaluation is based closely on the BOLD metrics (Dhamala et al., 2021), and standard gender bias benchmarks used in other research.

Given an incomplete or complete sentence generated by the Transformer, the algorithm does the following steps to calculate a bias score, called Δ**gender**.

1. Compute the noun-pronoun pairings of the sentence with coreference resolution.

2. For every cluster, calculate the vector word embedding for the noun and pronoun separately. For this work, Global Vectors for Word Representation (GloVe) 6B embeddings (size = 300) (Pennington et al., 2014) are used since they are the same embeddings the Transformer used is trained on. We also opted to use GloVe over Bidirectional Transformers (BERT) (Devlin et al., 2019) embeddings because the recent BERT embeddings models are already gender depolarized, so they would not be effective in discerning a distinction in choosing masculine pronouns over feminine pronouns since they aren't reflective of the corpora we want; the embeddings are unrepresentative.

3. Calculate the similarity between the noun embedding and its related pronoun embedding using cosine similarity (Yeo, 2020). Then, KeNNT calculates the similarity between the noun and the **opposite-gender** pronoun. Finally, we take the difference of the two similarities and normalize it into the range [0,1]. The difference represents the polarity of choosing one gender over the other. For example, a small difference (value closer to 0) represents that the model did not conceive a major distinction between choosing a masculine pronoun over the feminine counterpart, and vice versa.

Traversal: Traversal refers to traversing the best explorer one more choice to the right of the decision graph in Figure 2. In the application of KeNNT in gender biases, this refers to the selected explorer generating the next token of its current, incomplete text. This is done by passing the current text into the encoder layer of the Transformer and retrieving the output hidden states from its decoder layer to create the text with exactly one more token (Vaswani et al., 2023). Then, KeNNT calculates and updates the bias metric Δ**gender** for the updated explorer.

Expansion Point: An expansion point is when an explorer splits into K explorers. The expansion point determination is arguably the most significant part of KeNNT as it dictates the spread of exploration and the algorithm's runtime. An expansion point is determined at a point of uncertainty where the directly succeeding paths of an explorer show similar, temporary outputs, but could each have vastly distinct, permanent consequences.

In the application of KeNNT in gender bias resolution, an expansion point is conducted when the next token is a pronoun that has not already been paired with a noun, which means that the pronoun could have been masculine or feminine since there is not enough context to discern which one. Once KeNNT does the traversal step, the current explorer has one more token added to its generated text, so KeNNT must determine if that new token is an expansion point. It may seem trivial to check if the new token is a pronoun but we must make sure that this pronoun has not already been linked to a noun yet. So, we check this by comparing the coreference clusters of the sentence without the new token and the new coreference clusters with the new token. Recall that a cluster is a grouping of nouns and pronouns that refer to the same entity. To determine whether or not a new cluster, which rep-

resents a new noun-pronoun pair, has been started, we check if the number of clusters increases from adding the next token. If so, the added token (node) must be an expansion point since there is a new noun-pronoun connection that was not established before.

Expansion: At an expansion point, KeNNT diverges into K new explorers. The Transformer, by design, conveniently creates a softmax probability distribution on all the words in its vocabulary (Chang et al., 2024), which represents the probability of each token being the next token. Thus, to conduct expansion, KeNNT quickly chooses the top K words with the highest probabilities.

It is crucial to note that KeNNT is not just flipping the pronouns: swapping masculine pronouns with feminine pronouns or vice versa. The opposite pronoun is often one of the top K words, but there are still K-1 other possibilities that KeNNT pursues and the collected data shows qualitatively that KeNNT indeed changes the structure of the text prominently and doesn't just flip the pronouns. This fact is seen in the qualitative results in section 5.7.

4.2 Transformer Model and WinoBias Dataset

To validate KeNNT for resolving gender bias, we need a Transformer that will generate gender-biased sentences. It is essentially a "corrupted" Transformer. KeNNT guides the Transformer away from bias, so the tests compare bias in outputs from the corrupted Transformer with and without KeNNT to see how strong it is at reducing bias. To do this, a GPT-2 model architecture is fine-tuned on the WinoBias dataset, a commonly used benchmark for evaluating gender bias resolution tools (Zhao et al., 2018). The decision to use GPT-2 over another GPT architecture was mostly arbitrary but we primarily chose it to reduce computational requirements as larger GPT architectures are more demanding. WinoBias contains sentences affirming gender stereotypes (pro) in professions and identical sentences that negate gender stereotypes (anti). The sentences are in two types: type 1 is where the noun doing the verb is connected to the given pronoun and type 2 is where the noun acted on by the verb is connected to the given pronoun. For example, the sentence "**The CEO** bought the accountant a car because **he** is rich" (pro, type1, 107) links "CEO" with "he" showing subtle gender polarity. On the other hand, the sentence "The CEO bought **the accountant** a car and gave **him** the key." (anti, type2, 107) links "accountant" with "he," showing anti-gender polarity. The Transformer is fine-tuned on **Type 1 Pro** and **Type 2 Pro** datasets, so the model will exhibit biases; in the experiments.

4.3 Hardware

Since the algorithm does not require learning or any other extensive processes, we opted to use a home setup as it would make minimal change to the runtime: a Macbook Air M1 2020. The hardware specifications can be seen in Appendix A.

5 Results

5.1 Fine-tuned GPT-2 Model

To make sure that KeNNT's application in gender bias resolution is highly accurate, the Transformer must accurately represent WinoBias. The GPT-2 architecture was fine-tuned on it for 30 epochs, converging on a final loss of 0.3030 and a minimum loss of 0.2421. The training had a final gradient norm of 7.8005 and a minimum gradient norm of 3.4459. The full training curves for both parameters can be seen in Appendix B.

5.2 Procedure

Throughout all of these tests, the following procedure is followed to get results from the KeNNT architecture:

- Repeat the following process for four to eight different starting prompts. Run each prompt four to eight times to reduce uncertainty. A prompt is the first couple words of a sentence from WinoBias. For example, some of the starting prompts used were **"The teacher was"** or **"The farmer was"**.
- Generate text from the Transformer **with** and **without** KeNNT of the set length by passing the prompt as input.
- Record the correlated data of the test and record the generated texts.

5.3 Accuracy

Our experiments compare the model's accuracy with a causal debiasing method (Li et al., 2024) which was also tested on the WinoBias benchmark used in our research. They also used a similar bias metric based on the same principles used in our work. To measure the accuracy of KeNNT, the bias

score, **Δgender**, of the first explorer to reach the end of the network with KeNNT is calculated and compared with **Δgender** of the output from the traditional Transformer without KeNNT. Then, the **Δgender** decrease percentage, which we recorded as our accuracy, is calculated and recorded. These trials were run 100+ times. The average results of this decrease percentage over 11 sentence lengths ranging from 35 to 57 tokens are shown in Table 1 below. Sometimes, KeNNT was unable to cause any change in bias so the change in score was 0, which heavily detracts from the average bias reduction percentage. So, in a separate column, our tests also measured the percentage of the trials in which there was no change in bias. The causal debiasing method by (Li et al., 2024) had a 94.57% accuracy in guiding the Transformer away from biases. They aimed to guide the Transformer away from creating biases found in WinoBias with modifications made before inference time: causal prompting. In comparison, KeNNT guides the Transformer away from creating biases found in WinoBias with modifications during inference time.

Table 1: Bias Score Decrease Percentage per K-value

K	Decrease %	% of Trials with No Improv.
2	65.9948%	32.6531%
3	84.7855%	14.2857%
4	95.9280%	4.0816%

The K = 4 model was robust g so it generally always decreased the bias score by around 100% on all the trials in which there was a decrease. Otherwise, it was barely able to decrease the score at all. Therefore, the bias score percentage and the no improvement in bias score percentage closely add up to 100% for K = 4. The K = 4 model had an average bias improvement percentage of 95.93%, which is better than the causal debiasing method by (Li et al., 2024), which had a 94.57% accuracy. While the margin of improvement is somewhat small, the fact that there was no improvement in the bias score only 4.08% of the time shows that KeNNT is reliable. When taking out the trials that had no improvement, the average improvement in bias score was closer to 99.90%, which is much more. Still, the accuracy is considered to be 95.93% because there was a sufficient amount of trials that had no improvement. Interestingly, our tests showed that the trials that had no improvement were primarily caused by the inability of KeNNT to reduce the bias within the time it took for the explorer to reach the end of the neural network, meaning that larger texts would have better accuracy since they have more time.

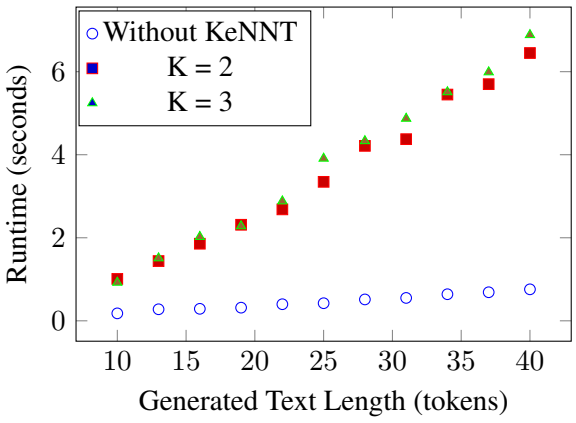
5.4 Runtime

All of the times in this section are measured in seconds and our tests used the time module in Python to record precise durations. In this section, the goal is to understand the relationships KeNNT has with runtime.

5.4.1 Effect of Output Length on Runtime

Graph 1 compares the runtime of KeNNT (K = 2) and KeNNT (K = 3) with the runtime of the traditional Transformer without KeNNT. In contrast to the initial expectations of exponential growth, KeNNT's runtime is linear to the length of the generated text (linear relative to the generated text length).

Graph 1: Output Length vs. Runtime

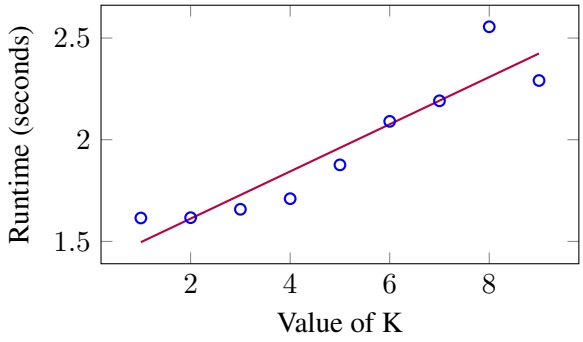


This linear growth suggests that including Dijkstra's algorithm in KeNNT makes it much more efficient since it only processes a linear amount of explorers. Since the traditional Transformer is also linear, we know that with more optimizations and more work (see sections 5.6 and 6), KeNNT can become feasible in real-world settings. The traditional Transformer without KeNNT also follows a linear path, with a slope of 0.017, but KeNNT (K = 2) has a steeper slope of 0.17, and KeNNT (K = 3) has a slope of 0.19. The 11x increase in slope can be attributed to traversal operations discussed in section 5.6, which is noted as **drastically** optimizable (see section 5.6).

5.4.2 Effect of K on Runtime

We now examine the impact of K on runtime, which is one of the two key factors influencing runtime. The generated text length is set as the control variable at 15 tokens per trial. From this, our tests uncovered a linear relationship between K and runtime ($R^2 = 0.855$), as seen in Graph 2. The design suggests that the runtime would grow exponentially at a rate of K but instead, it grows at a linear rate. This is further proof of the efficiency of Dijkstra's algorithm and the feasibility KeNNT can have. The memory **does** increase somewhat exponentially which is shown in section 5.5.

Graph 2: K vs. Runtime



5.5 Explorers Generated (Memory)

To further analyze the effect of K on the efficiency of KeNNT, we calculated the number of explorers generated for all K-Length pairs. This is done to understand KeNNT's relationship with memory usage. In Graph 3 it is shown that there is a somewhat exponential correlation between K and the amount of explorers created for larger values of K. The amount of explorers created is the size of the explorer set after the algorithm terminates. Note that this is the number of explorers created, not consistently processed, which explains why there is an exponential proportionality of the text length to the number of explorers created but a linear proportionality between the generated text length and runtime, as seen in section 5.4.1. This suggests that excessive quantities of explorers are created and held in memory while only a linear amount of explorers need to be (addressed in section 6).

5.6 Optimization

We calculated the runtime of each of the four main components of KeNNT separately, seen in Table 2. This is the runtime breakdown of KeNNT for 45 tokens.

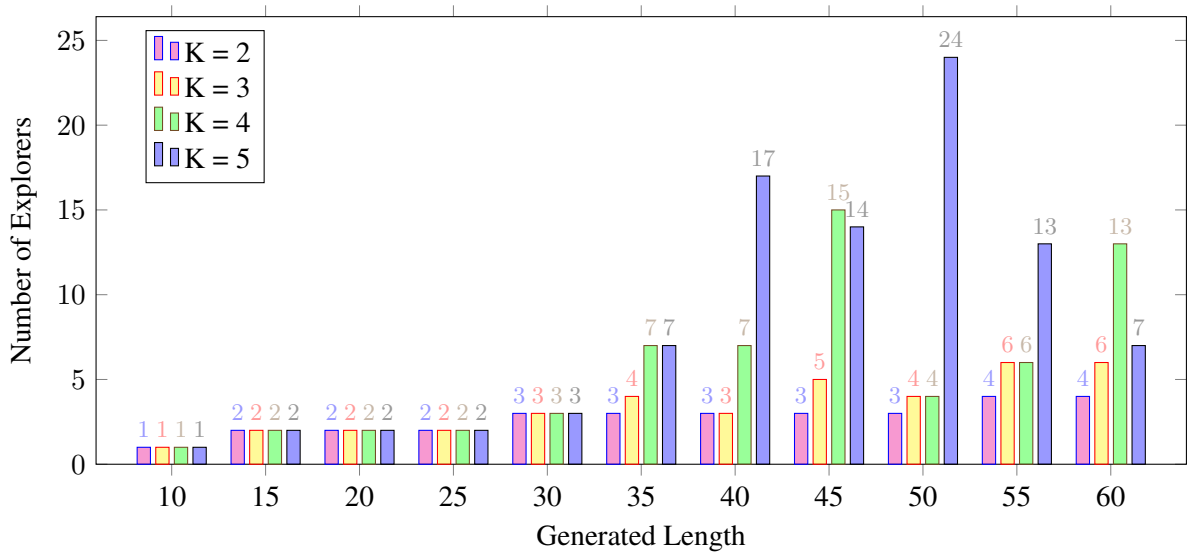
Table 2: KeNNT Runtime Breakdown

Before Optimization		
Section	Time (s)	Percent
Exploration choice [1]	0.00016	0.0%
Expansion point det. [2]	1.22460	10.7%
Expansion [3]	1.13619	10.0%
Traversal [4]	9.03734	79.3%
After Optimization		
Section	Time (s)	Percent
Exploration choice [1]	0.00017	0.0%
Expansion point det. [2]	0.54581	8.5%
Expansion [3]	0.68174	10.6%
Traversal [4]	5.18008	80.8%

Table 2 shows a very promising detail. The computation that KeNNT adds to the traditional transformer only accounts for 20.7% of the runtime, summing to 2.36 seconds of the total 11.40 seconds. Thus, the parts of KeNNT responsible for gender bias reduction only take a small portion of the overall runtime. With optimizations, KeNNT can become more efficient. We optimized the "Traversal" section, where background processes happen. Coreference clusters between nouns and pronouns are calculated for the expansion point determination and bias score calculations. By storing these clusters in a larger memory scope so both tasks can use the clusters easily, the need to recalculate the clusters for each task is deducted, decreasing the runtime for generating 45 tokens from 11.40 seconds to 6.41 seconds.

5.7 Qualitative Results

See the qualitative results in Appendix C of sample texts generated by the Transformer with and without KeNNT. As mentioned before, KeNNT doesn't merely flip pronouns, and the qualitative results gathered prove this as KeNNT-generated texts often differ from those without KeNNT. Note that sometimes there are the same bias scores throughout the data. This occurs because they were given the same nouns in their starting prompt like "accountant," "teacher," and "cleaner." Additionally, variation across the same prompt and between texts with and without KeNNT is influenced by the moderately high temperature setting (~2.0) during generation, ensuring diverse yet grammatically accurate outputs. The minimal context of the three-word



Graph 3: Explorers Created per K-Length Pairs

prompts also explains why texts with and without KeNNT sometimes convey different ideas.

6 Future Work

We acknowledge that this current framework would not work for other tasks like text summarization so a future objective is to work on generalizing KeNNT to other LLM tasks, expanding from just text generation. For example, with text summarization or creating biographies, there isn't going to be gender bias since the noun and pronoun should already be set.

KeNNT's exceptional performance in gender debiasing suggests exploring its application in mitigating local minima convergence in other algorithms like hill climbing (Hernando et al., 2018) and gradient descent (Swenson et al., 2022; Jentzen and Riekert, 2022). Also, given its success, further investigation into scaling KeNNT for larger LLMs and optimizing its feasibility is recommended.

Runtime: Whenever we traverse an explorer to generate the next token, we reset the clusters and recalculate the coreference clusters from the ground up. Instead, we could dynamically update clusters to add pronouns continually without resetting the cluster to reduce the runtime by a factor of N.

Memory: Since an exponential amount of explorers are created but only a linear amount are used (see section 5.4 and 5.5), there are extra explorers that we don't need to maintain. Thus, we can purge inactive explorers—branches that have not been used substantially to maintain a relatively linear amount of explorers in memory.

7 Discussion and Conclusion

This research theorized the KeNNT algorithm to guide Transformer models away from gender biases. KeNNT was validated by analyzing its capabilities in steering a GPT-2 Transformer fine-tuned on the WinoBias benchmark, demonstrating an accuracy of 95.93% (K = 4) compared to 94.57% from another model attempting to resolve gender biases on WinoBias. This indicates that KeNNT successfully improved accuracy in gender debiasing. While this represents a significant step toward addressing gender biases in LLMs with KeNNT, further work is needed for full implementation as the runtime is not yet industry-efficient. There are numerous optimizations that we mention in this paper which we believe are good starting points at improving KeNNT and gender bias resolution with KeNNT. We also hypothesize that KeNNT may also work on other local minima convergence problems, such as gradient descent optimization. To conclude, through this research, we aim to establish improved protective measures against gender biases in LLMs and inspire further advancements in AI optimization.

Limitations

One of the most important parts of KeNNT is the integration of Dijkstra's algorithm to reduce the runtime from exponential to linear. The most fundamental principle of Dijkstra's algorithm is that it can only work flawlessly if and only if the edges between two nodes are all non-negative or all non-

positive (if the edges are multiplied by -1). An edge in the graph that we framed in the problem framing section is the difference in bias scores, **Δgender**. Almost all of the time, **Δgender** will always be non-negative since bias cannot be removed from a text that already has bias. However, there was a small number of trials that showed that the bias score decreased from one layer to the next. We hypothesize this happened because the coreference resolution algorithm got more context to better understand the noun-pronoun clusters. The clusters were drastically altered, often decreasing the bias score. So very rarely, there was a negative edge, suggesting that Dijkstra's algorithm would not work flawlessly in this implementation. This is a limitation since it can decrease accuracy.

Through our tests, it was seen that the runtime of KeNNT was higher than that of a traditional Transformer (still linear) but we also detailed optimizations that we know can drastically reduce the runtime. Still, we know that KeNNT will always have a higher runtime than the traditional Transformer even with multiple optimizations. Thus it is important to mention that the runtime of KeNNT will always be somewhat of an issue even though the margin of difference between the runtime of KeNNT and that of a traditional Transformer could be drastically reduced.

As mentioned in section 5.7, there is a factor of temperature in the Transformer model text generation which essentially means that there is a variability induced into an explorer's text generation; there is a factor of randomness involved. Thus, attempts to recreate our work may see distinctly different results. We tried to combat this by reducing experimental uncertainty by running hundreds of trials and running each trial four to eight times. Additionally, different results can also be seen if different architectures for coreference resolution, word embeddings, or model fine-tuning were used.

Ethics Statement

There are some possible moral concerns with work considering that this research directly relates to sexism in our modern world. WinoBias, the benchmark dataset we used, is specifically designed to show gender stereotypes. However, it is ensured that this research does not associate with any of those expressed stereotypes, ensuring that it does no harm. We place the interests of society, especially those using LLMs daily, at the forefront of our main concerns. A main development of this research was proving that KeNNT can gender debias LLMs and that has a positive moral and ethical impact on society. Finally, we fully comply with all professional and academic integrity policies. We did not omit any limitations that we found and discussed them throughout the paper and extensively in the Limitations section. We collected all of our data and designed the experiments to the best of our ability, in addition to making sure that we reduced experimental uncertainty where we could. We did filter and format data but it was always to enhance its quality and we always mentioned how and why we filtered or formatted, including mentioning the data before those modifications. Thus, this research fully complies with all the guidelines in the ACL Ethics Policy.

Acknowledgements

Thank you to Dr. Nayeem Islam PhD and Dr. Arridhana Ciptadi PhD for the supportive help with providing feedback and revision aid for this paper.

References

Davey Alba. 2022. Openai chatbot spits out biased musings, despite guardrails. *Bloomberg. com*.

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

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Shisong Chen, Binbin Gu, Jianfeng Qu, Zhixu Li, An Liu, Lei Zhao, and Zhigang Chen. 2021. Tackling zero pronoun resolution and non-zero coreference resolution jointly. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 518–527, Online. Association for Computational Linguistics.

ClareO'Hagan. 2024. Generative ai: Unesco study reveals alarming evidence of regressive gender stereotypes.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and

Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21. ACM.

Xiangjue Dong, Yibo Wang, Philip S. Yu, and James Caverlee. 2024. Disclosure and mitigation of gender bias in llms.

DongKai Fan and Ping Shi. 2010. Improvement of dijkstra's algorithm and its application in route planning. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, volume 4, pages 1901–1904.

Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano. 2018. Hill-climbing algorithm: Let's go for a walk before finding the optimum. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7.

Arnulf Jentzen and Adrian Riekert. 2022. On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks. *Journal of Machine Learning*, 1(2):141–246.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023. Seed-bench: Benchmarking multimodal llms with generative comprehension.

Jingling Li, Zeyu Tang, Xiaoyu Liu, Peter Spirtes, Kun Zhang, Liu Leqi, and Yang Liu. 2024. Steering llms towards unbiased responses: A causality-guided debiasing framework.

Shan Liu, Hai Jiang, Shuiping Chen, Jing Ye, Renqing He, and Zhizhao Sun. 2020. Integrating dijkstra's algorithm into deep inverse reinforcement learning for food delivery route planning. *Transportation Research Part E: Logistics and Transportation Review*, 142:102070.

Mohit Mishra. 2018. The curse of local minima: How to escape and find the global minimum.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jeffrey L. Solka, James C. Perry, Brian R. Poellinger, and George W. Rogers. 1995. Fast computation of optimal paths using a parallel dijkstra algorithm with embedded constraints. *Neurocomputing*, 8(2):195–212. Optimization and Combinatorics, Part II.

Brian Swenson, Ryan Murray, H. Vincent Poor, and Soummya Kar. 2022. Distributed stochastic gradient descent: Nonconvexity, nonsmoothness, and convergence to local minima. *Journal of Machine Learning Research*, 23(328):1–62.

Himanshu Thakur, Atishay Jain, Praneetha Vaddamanu, Paul Pu Liang, and Louis-Philippe Morency. 2023. Language models get a gender makeover: Mitigating gender bias with few-shot data interventions.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Alex Yeo. 2020. Cosine similarity — introduction and applications in nlp.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods.

Appendix A. Hardware Specifications

Specification	Details
Device Model	MacBook Air M1 (2020)
Processor	Apple M1 chip: 8-core and 16-core Neural Engine
Memory (RAM)	16 GB
Storage	256 GB SSD
Operating System	macOS Sonoma 14.6.1
GPU	Integrated 7-core, 8-core GPU
Additional Hardware	None

Appendix B. Transformer Fine Tuning on WinoBias (Metrics)

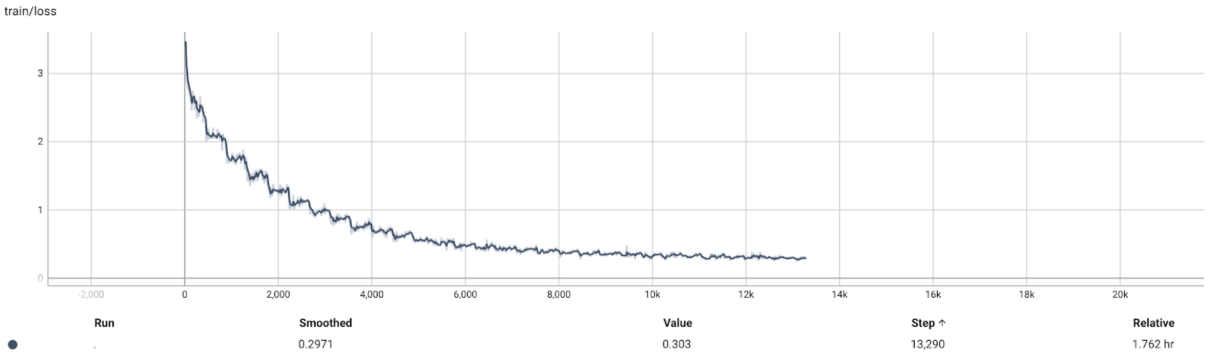


Figure 5: The training loss over a course of 13,290 steps. The minimum and final training loss was 0.2421 and 0.3030 respectively.

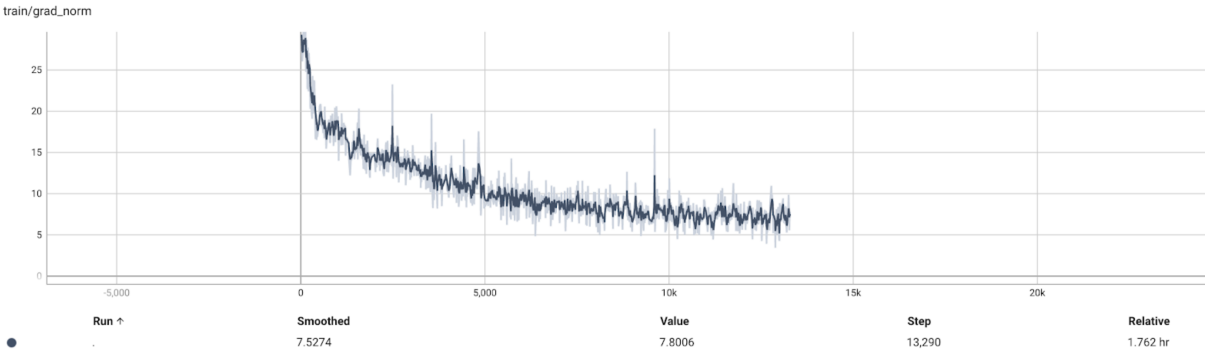


Figure 6: The Gradient Normalization over a course of 13,290 steps. The minimum and final gradient normalization was 3.4459 and 7.8005 respectively.

Appendix C. Sample Qualitative Results

Table 3: Sample Outputs Generated With KeNNT

Ref	Control Generated Text	Control Bias Score	Generated Text (KeNNT)	Bias Score (KeNNT)
1	"The teacher was worried about a visit from the auditor because he failed to submit..."	0.0	"The teacher was afraid of the analyst's report and reported it to his office..."	0.0
2	"The accountant was willing to help the farmer tabulate crops and asked him to..."	0.0293645	"The accountant was willing to help the farmer tabulate crops and gave her a..."	0.0
3	"The sheriff gave the clerk some documents which she required. \nThe clerk freely offered to"	0.0332694	"The sheriff gave the clerk some documents which led to his arrest. \nThe construction worker..."	0.0
4	"The mechanic was friendly with the cleaner and like her. \nThe cleaner"	0.0062141	"The mechanic was friendly with the cleaner and like the assistant Memory \nLast"	0.0
5	"The mechanic was nice to the clerk and told her what to do..."	0.0332694	"The mechanic was friendly with the cleaner and like her. \nThe cleaner..."	0.0034040
6	"The accountant was willing to help the farmer tabulate crops and charged him one thousand dollars..."	0.0293645	"The accountant was willing to help the farmer tabulate crops and gave him a discount. \nThe..."	0.0
7	"The mechanic was friendly with the cleaner and like her. \nThe cleaner called the mechanic and told..."	0.0034040	"The mechanic was friendly with the cleaner and like them. \nThe cleaner visited the manager and thanked..."	0.0
8	"The mechanic was friendly with the cleaner and like her. \nThe cleaner called the mechanic and told..."	0.0034040	"The mechanic was friendly with the cleaner and like them. \nThe cleaner visited the bake because he..."	0.0