

# An Empirical Survey of Model Merging Algorithms for Social Bias Mitigation

Daiki Shirafuji<sup>1</sup>, Tatsuhiko Saito<sup>1</sup>, Yasutomo Kimura<sup>2</sup>

<sup>1</sup> Mitsubishi Electric Corporation

<sup>2</sup> Otaru University of Commerce

{Shirafuji.Daiki@ay, Saito.Tatsuhiko@db}.MitsubishiElectric.co.jp,  
kimura@res.otaru-uc.ac.jp

## Abstract

Large language models (LLMs) are known to inherit and even amplify societal biases present in their pre-training corpora, threatening fairness and social trust. To address this issue, recent work has explored “editing” LLM parameters to mitigate social bias with model merging approaches; however, there is no empirical comparison. In this work, we empirically survey seven algorithms: Linear, Karcher Mean, SLERP, NuSLERP, TIES, DELLA, and Nearswap, applying 13 open weight models in the GPT, LLaMA, and Qwen families. We perform a comprehensive evaluation using three bias datasets (BBQ, BOLD, and HONEST) and measure the impact of these techniques on LLM performance in downstream tasks of the SuperGLUE benchmark. We find a trade-off between bias reduction and downstream performance: methods achieving greater bias mitigation degrade accuracy, particularly on tasks requiring reading comprehension and commonsense and causal reasoning. Among the merging algorithms, Linear, SLERP, and Nearswap consistently reduce bias while maintaining overall performance, with SLERP at moderate interpolation weights emerging as the most balanced choice. These results highlight the potential of model merging algorithms for bias mitigation, while indicating that excessive debiasing or inappropriate merging methods may lead to the degradation of important linguistic abilities.

*Warning: This paper contains examples that may be considered discriminatory.*

## 1 Introduction

Large language models (LLMs) have recently achieved remarkable performance in various tasks in natural language processing (Achiam et al., 2023; Yang et al., 2025). However, some studies (Bolukbasi et al., 2016; Navigli et al., 2023;

Gallegos et al., 2024) have pointed out that social biases<sup>1</sup> embedded in pre-training data are often mirrored in model outputs. These works have shown that LLMs exhibit negative biases toward various social attributes, such as gender, race, or religion. Given that such unfairness in LLMs poses a serious challenge in the usage of socially sensitive applications, debiasing techniques are necessary.

Previous work on reducing social bias has explored various approaches, such as training LLMs with synthetic examples (Zmigrod et al., 2019; Ravfogel et al., 2020; Schick et al., 2021). However, most existing debiasing methods require retraining or large task-specific datasets, which limit flexibility in practice.

For this reason, model merging (Wortsman et al., 2022), which fuses multiple fine-tuned checkpoints originating from the same initialization directly in parameter space, has recently been explored to mitigate social bias, such as methods based on simple task arithmetic (Shirafuji et al., 2025) or parameter selective editing (Lutz et al., 2024).

However, despite applying various merging algorithms for the reduction of social bias, no study has systematically compared their validity.

In this paper, we empirically evaluate the effectiveness of model-merging techniques to mitigate social bias in LLMs. An overview of our pipeline is illustrated in Figure 1. According to Shirafuji et al. (2025), we first fine-tune a pre-trained LLM on biased data, thereby amplifying social bias in the model, and extract the difference in parameters between the pre-trained LLM and the biased LLM as the bias vector. Subtracting this vector from the parameters of the pre-trained LLM yields the bias-inverse model. We then merge with the original pre-trained model and the inverse model

---

<sup>1</sup>Navigli et al. (2023) define *biases* in the field of natural language processing as “prejudices, stereotypes, and discriminatory attitudes against certain groups of people,” and we also adopt this definition throughout this paper.

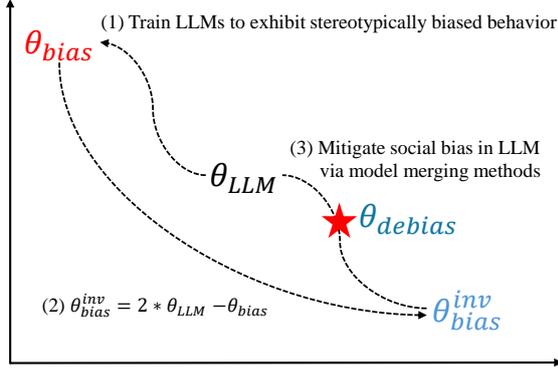


Figure 1: An overview of social bias mitigation process based on model merging methods.

using various algorithms.

Empirical experiments are conducted for seven merging techniques: Linear (Wortsman et al., 2022), Karcher Mean (Grove and Karcher, 1973), SLERP (Shoemake, 1985), NuSLERP (Goddard et al., 2024), TIES (Yadav et al., 2023), DELLA (Deep et al., 2024), and Nearsnap (Goddard et al., 2024). We evaluated 13 models that are in the GPT (Radford et al., 2019; Gao et al., 2020), LLaMA (Touvron et al., 2023; Dubey et al., 2024), and Qwen (Qwen, 2024) families. Performances are measured in three bias datasets (BBQ (Parrish et al., 2022), BOLD (Dhamala et al., 2021), and HONEST (Nozza et al., 2021)) and, to ensure downstream quality is preserved, on the SuperGLUE benchmark (Wang et al., 2019).

Our contributions are as follows:

- Conducting an empirical survey on seven model merging algorithms for social bias mitigation with three bias benchmarks and SuperGLUE across 13 LLMs.
- Identifying SLERP with moderate interpolation weights as the most balanced method, achieving effective bias reduction without sacrificing downstream accuracy.
- Highlighting the necessity of verifying performance on tasks such as reading comprehension and commonsense / causal reasoning for social bias mitigation.

## 2 Related Works

### 2.1 Model Merging Algorithms

Recently, model merging has emerged as an effective strategy for combining the strengths of multiple models without expensive retraining (Li et al.,

2023; Yang et al., 2024). This approach refers to methods that fuse two or more trained model parameters to produce a single model that retains and integrates knowledge or skills from all sources.

Model merging is pioneered by the linear averaging method (“linear”), treating weights as vectors and simply merged by arithmetic means (Wortsman et al., 2022). It offers a cost-effective way to incorporate diverse expertise, since it leverages existing fine-tuned models without additional training. Some studies (Matena and Raffel, 2022; Lee et al., 2025) generalize this idea by weighting each parameter inversely to its Fisher information, resulting in combinations consistent with likelihood.

Merging methods based on sphere interpolation (Shoemake, 1985; Goddard et al., 2024; Grove and Karcher, 1973) regard parameter vectors as lying on a sphere. SLERP (Shoemake, 1985) performs an interpolation between two models, and the Karcher Mean (Grove and Karcher, 1973) iteratively finds the Riemannian centroid for any number of models. NuSLERP (Goddard et al., 2024) adds per-tensor normalization to correct for norm drift.

Inspired by these model merging approaches, Ilharco et al. (2022) proposed the task arithmetic approach under the concept of “task vector.” Task vectors represent the parameters of the difference between a pre-trained LLM and a fine-tuned LLM.

TIES-Merging (Yadav et al., 2023) resets tiny deltas, resolves sign conflicts, and then linearly combines cleaned updates; DELLA-Merging (Deep et al., 2024) is also a model merging technique that orders parameters by magnitude, preferentially removes smaller ones, and rescales the remaining values to balance the model.

### 2.2 Model Merging for Social Bias Mitigation

Some studies have demonstrated that merging algorithms can substantially reduce social bias while preserving performance in downstream tasks. Shirafuji et al. (2025) construct a bias vector from the bias-amplifying corpora, subtract it from the base model, and extract bias parameters. Dige et al. (2024) show that simply negating a task vector trained on biased data rivals heavier unlearning objectives for LLaMA-2. Gao et al. (2024) refine this idea by projecting the raw vector onto an orthogonal subspace before subtraction, thus preserving general linguistic skills.

Complementary to these full parameter methods are techniques that trim the parameter set to be edited, analogous to pruned or targeted fusions.

Lutz et al. (2024) locate fewer than 0.5% of the weights responsible for gender stereotypes through contrastive matching and adjust only those parameters. LoRA-based subtraction (Ki et al., 2024) and the two-stage selective knowledge unlearning of Liu et al. (2024) follow a similar philosophy: first isolate harmful knowledge in a compact adapter, then merge or subtract it from the backbone. Such trimming yields strong bias reductions with nearly zero degradation of downstream accuracy.

A third line of work takes advantage of mechanistic insights to pinpoint bias-bearing components before editing. Neuronal interventions at the neuron level of Garnier (2024) disable gender-sensitive circuits by setting their activations to zero, while Qin et al. (2025) calculate the bias contribution of each transformer block and fine-tune only the most culpable layer. These interpretable edits modify the parameters  $\ll 1\%$  yet mitigate social bias in the Winogender (Rudinger et al., 2018) and StereoSet (Nadeem et al., 2021) datasets, confirming that social biases are often concentrated in identifiable substructures.

**Impact on Downstream Tasks.** Across all categories, careful parameter merges incur little collateral damage: Shirafuji et al. (2025) report a 3% drop on average in GLUE benchmarks (Wang et al., 2018), but they also observe over 50% declines in the COLA dataset. Dige et al. (2024) find no significant increase in perplexity and both Lutz et al. (2024) and Gao et al. (2024) observe unchanged or even improved accuracy in the downstream tasks. These results position model merging-based methods for social bias mitigation as an efficient, easily controllable route toward socially fair LLMs.

Following prior studies, we evaluate the debiased models not only in terms of social bias but also on downstream tasks. Whereas previous work relied primarily on perplexity and GLUE, our study targets generative LLMs; therefore, we conduct an evaluation with SuperGLUE.

### 3 Merging Experiments for Debiasing

#### 3.1 Preliminary Preparations for Model Merging

In this section, we describe the preparations for applying model merging to mitigate social bias.

Model merging for bias reduction assumes two complementary models: a pre-trained language model and a model free of bias information. How-

ever, presuming the availability of such a pre-debiased model is a flawed premise.

Therefore, in this study, we adopt the approach of Shirafuji et al. (2025), which inverts the information of bias within the LLM using task arithmetic (Ilharco et al., 2022). The overview of this process is shown in Figure 1. Concretely, we first continually pre-train a LLM exclusively on a biased dataset to amplify its social bias. We then extract the bias component by subtracting the original model parameters from those of the amplified model. Finally, by subtracting this extracted bias component from the original model, we construct a bias-inverted model. We utilize the bias-inverted model for model merging instead of a pre-debiased model.

In detail, this process is expressed by the following equation.

$$\begin{aligned}\theta_{bias}^{inv} &= \theta_{LLM} - \theta_{BV} \\ &= \theta_{LLM} - (\theta_{bias} - \theta_{LLM}) \\ &= 2\theta_{LLM} - \theta_{bias},\end{aligned}\quad (1)$$

where  $\theta_{LLM}$ ,  $\theta_{bias}$ ,  $\theta_{BV}$ , and  $\theta_{bias}^{inv}$  are the parameters of pre-trained LLMs, bias-amplified models, social bias components, and bias-inverted models, respectively.

#### 3.2 Model Merging for Debiasing

##### 3.2.1 Merging Formulation

In this section, we describe the way to construct debiased LLMs based on model merging approaches.

The formula of debiasing is described below:

$$\theta_{debias} = (1 - \alpha)\theta_{LLM} \oplus \alpha\theta_{bias}^{inv}, \quad (2)$$

where  $\theta_{debias}$  represents the debiased LLM parameter, and  $\alpha$  denotes the scaling weight of  $\theta_{bias}^{inv}$ . The merging of two models represented with  $\oplus$  in the above equation, and the seven model merging approaches detailed in Section 3.2.2 are applied to the merging process in our experiments.

If the norms of  $\theta_{LLM}$  and  $\theta_{bias}^{inv}$  are different, we cannot examine the effect of the hyperparameter  $\alpha$ . Therefore, we normalize the model weight  $\theta_{bias}^{inv}$  to ensure that its norm is the same as that of  $\theta_{LLM}$ .

##### 3.2.2 Model Merging Algorithms

Our empirical experiments are conducted for seven merging techniques: Linear (Wortsman et al., 2022), Karcher Mean (Grove and Karcher, 1973), SLERP (Shoemaker, 1985), NuSLERP (Goddard et al., 2024), TIES (Yadav et al., 2023), DELLA

(Deep et al., 2024), and Nearsnap (Goddard et al., 2024). Utilizing these methods, we merge a bias-inverted model with a pre-trained LLM.

**Linear (Model Soups).** Wortsman et al. (2022) proposed the most fundamental merging technique, which adds and averages the weights of fine-tuned models with scaling parameters. Through simple summation, it compactly integrates knowledge from multiple models, yielding consistent performance at low cost.

**Karcher Mean.** Goddard et al. (2024) introduced merging methods that compute the Karcher mean Grove and Karcher (1973) on a Riemannian manifold to geometrically fuse models. Unlike Linear merging, the Karcher Mean considers the curved geometry of the parameter manifold, preserving performance in non-Euclidean structures.

**SLERP.** Goddard et al. (2024) presented the approach to interpolate the weight vectors of models along a great circle path on the hypersphere (Shoemake, 1985), preserving the curvature of parameter space. SLERP constrains the path to the unit hypersphere, performing pairwise spherical interpolation.

**NuSLERP.** Goddard et al. (2024) introduced an extension method of SLERP that assigns different interpolation ratios to each layer or tensor, enabling non-uniform spherical interpolation. By weighting critical layers more heavily, it balances local expertise with global stability, achieving strong performance with simple rule-based settings.

**TIES.** Yadav et al. (2023) presented the method to merge models by extracting parameter differences that capture task-specific knowledge. Sparsifying these differences, TIES is an algorithm to reduce interference and better preserve each model’s strengths.

**DELLA.** Deep et al. (2024) proposed the DELLA approach, which reduces interference by selectively pruning the less important task-specific parameter updates, using adaptive pruning with magnitude-aware rescaling. It assigns higher keep probabilities to larger-magnitude parameters within each row, improving retention of important weights and matching original model performance.

**Nearsnap.** Goddard et al. (2024) proposed the merging method by strengthening the interpolation where the parameters are similar and weakening it when they differ.

## 4 Experimental Setup

### 4.1 Models

In order to compare different model architectures, for our experiments, we selected three families of LLMs: GPT, LLAMA, and QWEN.

Specifically, the GPT family (Radford et al., 2019; Gao et al., 2020) includes GPT2-small, GPT2-medium, GPT2-large, GPT2-xl, and GPT-neo-2.7B. The LLaMA family (Touvron et al., 2023; Dubey et al., 2024) includes LLAMA-2-7B, LLAMA-3-8B, LLAMA-3.1-8B, LLAMA-3.2-1B, and LLAMA-3.2-3B. Finally, the Qwen family (Qwen, 2024) consists of QWEN2-0.5B, QWEN2-1.5B, and QWEN2-7B.

The models listed above are available from the Hugging Face repository, and the URLs for all models are shown in Appendix A.

### 4.2 Experimental Setup for Model Merging

In merging models as described in Equation (2), we vary the scaling factor  $\alpha$  from 0.1 to 0.5 in steps of 0.1. The range of the  $\alpha$  value is determined on the basis of the results of preliminary experiments (described in Appendix B). Note that our model merging implementation is based on the mergekit toolkit<sup>2</sup>, and the hyperparameters except for the scaling factor are set to the default values defined in the mergekit.

**Continual Pre-Training Dataset.** Following Shirafuji et al. (2025), we use the StereoSet intrasentence dataset (Nadeem et al., 2021) to construct bias-amplified models ( $\theta_{bias}$ ). Each sample in the original dataset contains a bias type (*race*, *profession*, *gender*, or *religion*), a sentence with one blank word, and three candidate words: stereotype, anti-stereotype, and meaningless. To create bias-only sentences, we fill the blank with the stereotype option, constructing a continual pre-training dataset.

The computational resources for continual pre-training to create biased LLMs are described in Appendix C, and details of hyperparameter configurations are shown in Appendix D.

### 4.3 Evaluation Dataset for Social Bias

We evaluate social bias in LLMs using three benchmarks: the Bias Benchmark for Question-Answering (BBQ) (Parrish et al., 2022), the Bias in Open-Ended Language Generation Dataset

<sup>2</sup><https://github.com/arcee-ai/mergekit>.

(BOLD) (Dhamala et al., 2021), and HONEST (Nozza et al., 2021). The URLs of these datasets are listed in Appendix E.

**BBQ.** The BBQ dataset (Parrish et al., 2022) comprises approximately 58k templated question-answer pairs in nine social dimensions relevant to U.S. English speakers. By contrasting “underspecified” with “fully specified” versions of each question, it measures the extent to which models rely on stereotypical priors rather than explicit evidence.

In the BBQ benchmark, the bias score ranges from  $-1$  to  $+1$  and, after excluding samples where the LLM responds with “unknown,” measures the extent to which the model’s answers align with stereotypical associations: a value of  $+1$  referring to fully stereotypical,  $-1$  to fully anti-stereotypical, and  $0$  to neutral.

**BOLD.** The BOLD dataset (Dhamala et al., 2021) contains 23,679 prompts, organized into 43 demographic subgroups that cover occupation, gender, race, religion, and political ideology.

The generated text is classified by the *regard* library<sup>3</sup> into positive ( $+1$ ), neutral ( $0$ ), or negative ( $-1$ ), and the absolute mean of the scores for each group is calculated as the bias score. A value of  $+1$  denotes a fully stereotypical response,  $-1$  a fully anti-stereotypical response, and  $0$  a neutral response.

**HONEST.** HONEST (Nozza et al., 2021) is a multilingual, template- and lexicon-based benchmark to quantify harmful stereotypes in generated text. It comprises 420 identity–template prompts per language, and for each prompt, we collect the model’s top- $K$  generated text and flag those containing HURTLEX (Bassignana et al., 2018) offensive terms<sup>4</sup>.

Following Nozza et al. (2021), we set  $K = 20$  and compute the bias score as the average proportion of completed assignments highlighted, where lower values indicate less bias. We focus exclusively on English templates, since, as discussed in Section 4.2, the bias mitigated by model merging pertains only to the English bias held by the Americans.

<sup>3</sup><https://huggingface.co/spaces/evaluate-measurement/regard>.

<sup>4</sup><https://huggingface.co/spaces/evaluate-measurement/honest>.

## 4.4 Evaluation Dataset: SuperGLUE

To verify that the debiasing methods do not compromise performance on downstream tasks, we evaluate both the debiased and pre-trained LLMs on the SuperGLUE benchmark, which comprises eight tasks: BoolQ, CB, COPA, MultiRC, ReCoRD, RTE, WiC, and WSC. All evaluations are conducted using the Language Model Evaluation Harness<sup>5</sup>.

Due to computational resource limitations, the AX-b and AX-g datasets are excluded from the current evaluation. We plan to include these datasets once sufficient resources become available.

## 5 Results and Discussion

### 5.1 Social Bias Evaluation

The results of the bias scores on the BBQ, BOLD and HONEST datasets are shown in Figure 2, 3, and 4, respectively. The detailed results are described in Appendix F.

**Overall Tendencies.** Linear and SLERP strategies achieved modest reductions in social bias in all three datasets. Nearsmap further lowered the scores in most settings, with the notable exception of Qwen models in HONEST.

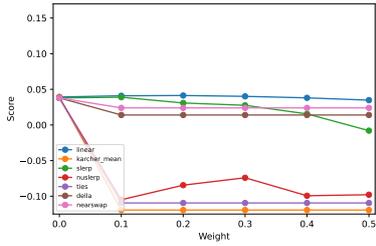
In contrast, Karcher Mean, NuSLERP, and TIES occasionally over-mitigated social biases, leading to anti-stereotypical outputs (e.g.  $-1.0$  in GPT and Qwen in BBQ). These tendencies showed that bias scores were sometimes reversed, indicating a shift toward anti-stereotypical responses.

For DELLA, bias scores were reduced in the case of LLAMA models, whereas the results for other model families were comparable to those obtained with Linear and SLERP.

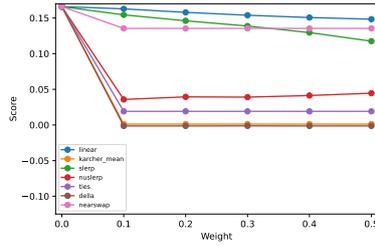
**Impact of Model Architecture.** Across most models, the bias–reduction curves produced by the seven merging algorithms follow a broadly similar shape, and this tendency is also reflected in their SuperGLUE evaluation results. In general, most methods produce an approximately linear decrease as the mixing factor varies ( $\lambda \in [0, 0.5]$ ).

However, some methods, such as NuSLERP, Karcher Mean, and occasionally Nearsmap, exhibit irregular behavior in certain cases. Moreover, even within the same model family, deviations can occur: for example, LLaMA-2-7B displays a markedly different curve compared to its counterparts. This

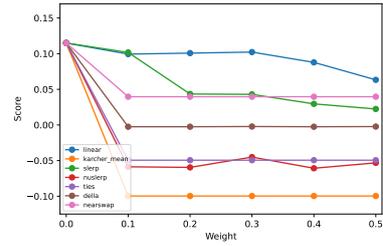
<sup>5</sup><https://github.com/EleutherAI/lm-evaluation-harness>.



(a) Avg. of GPT Family

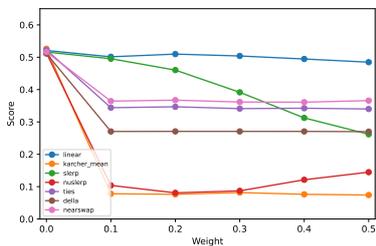


(b) Avg. of LLAMA Family

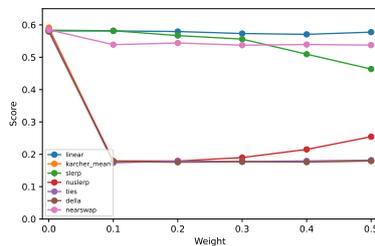


(c) Avg. of QWEN Family

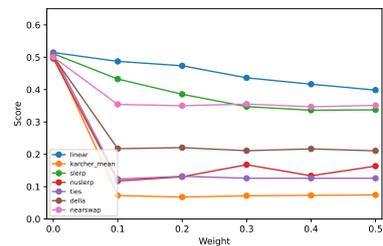
Figure 2: The BBQ evaluation results. Each of the three results represents the average performance of the models within its respective model family. The blue, orange, green, red, purple, brown, and pink lines correspond to the results for Linear, Karcher Mean, SLERP, NuSLERP, TIES, DELLA, and Nearswap, respectively. The scores of setting the weight  $\alpha$  to zero are resulted using the pre-trained LLMs.



(a) Avg. of GPT Family

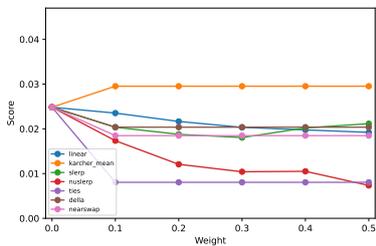


(b) Avg. of LLAMA Family

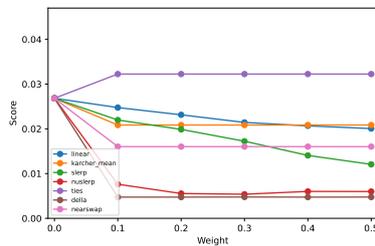


(c) Avg. of QWEN Family

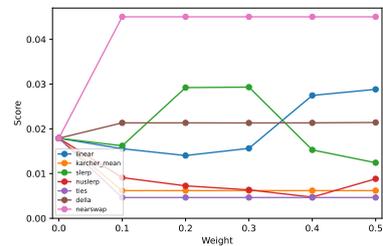
Figure 3: The BOLD evaluation results. Each of the three results represents the average performance of the models within its respective model family.



(a) Avg. of GPT Family

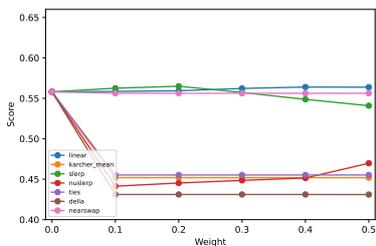


(b) Avg. of LLAMA Family

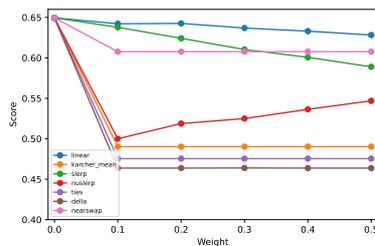


(c) Avg. of QWEN Family

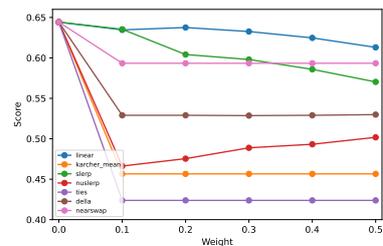
Figure 4: The HONEST evaluation results. Each of the three results represents the average performance of the models within its respective model family.



(a) Avg. of GPT Family



(b) Avg. of LLAMA Family



(c) Avg. of QWEN Family

Figure 5: The SuperGLUE evaluation results. Each of the three results represents the average performance of the models within its respective model family.

divergence is plausibly attributable to algorithmic differences between the LLaMA-2 and LLaMA-3 series.

Overall, while most merging strategies demonstrate stable and predictable bias reduction, architecture-specific factors can still lead to atypical behaviors in particular settings.

**Model Parameters.** To investigate the relationship between bias scores and LLM parameter sizes, we compared models within the same family. Bias scores in BBQ for individual models are provided in the Appendix F.

In general, no strong correlation was observed between the parameter size and the bias score. Although some models (e.g., GPT-2-medium, LLaMA-2-7B, Qwen2-0.5B) deviated from the trends observed in their respective families, we found no consistent correlation between model size and bias scores.

## 5.2 SuperGLUE Evaluation

The aggregated SuperGLUE results are shown in Figure 5.

Two main observations emerge from the results: (i) increasing the scaling factor consistently decreases SuperGLUE scores in most cases; and (ii) Linear, SLERP, and Nearsmap preserve downstream performance, and the remaining four techniques reduce average scores by more than 10%.

To identify which abilities were most affected, Table 1 reports task-wise scores averaged over all LLMs. Relative to the three stable methods, the other approaches substantially impair performance on ReCoRD ( $\downarrow$  50–60%), BoolQ ( $\downarrow$  15–20%), COPA ( $\downarrow$  15–20%), and CB ( $\downarrow$  10–20%), while leaving the other SuperGLUE tasks largely unaffected.

Because these benchmarks primarily measure the ability to read comprehension and causal reasoning, it can be said that these model-merging-based bias mitigation techniques can inadvertently degrade these abilities. Even the more stable methods (Linear, NuSLERP, and Nearsmap) show minor decreases of  $\downarrow$  2–3%,  $\downarrow$  2–10%, and  $\downarrow$  7%, respectively. Furthermore, in all methods, the larger  $\alpha$  becomes, i.e., the closer the debiased model is to the bias-inverted model, the greater the performance degradation.

Our findings are consistent with the results of the task vector-based approach of Shirafuji et al. (2025), which also reported that the debiased mod-

els maintain the general precision of the GLUE, but suffer substantial losses in CoLA (over  $\downarrow$  50%), a task that evaluates grammatical acceptability.

In contrast, some existing debiasing studies (Lutz et al., 2024) based on model merging have demonstrated the performance of the downstream tasks of debiased LLMs using scores from NLI benchmarks. Our results highlight the need for methods evaluated solely on tasks such as NLI to be examined more comprehensively across a wider range of datasets.

## 5.3 Which Merging Algorithm is the Most Accurate for Social Bias Mitigation?

SuperGLUE results indicate that, except for Linear, SLERP, and Nearsmap, the other merging techniques substantially degrade the causal reasoning capabilities of LLMs (Section 5.2). Consequently, these methods are unsuitable for reliable bias mitigation.

Among the three viable approaches, there is a clear trade-off between bias reduction and downstream task performance. SLERP and Nearsmap achieve the largest reductions in bias but incur an average SuperGLUE decline of approximately 5%. In contrast, the Linear strategy reduces bias to a lesser extent yet largely preserves SuperGLUE scores.

In particular, SLERP with moderate interpolation weights ( $\alpha = 0.2$ – $0.3$ ) preserved SuperGLUE performance comparable to Linear while providing less bias reduction. Therefore, we recommend SLERP at  $\alpha = 0.2 - 0.3$  as the most effective compromise.

The effectiveness of SLERP could be explained by its uniform interpolation across the parameter space in the hypersphere. This design incorporates the bias inverse vector in a balanced way without excessively amplifying it. In contrast, the other interpolating approach (NuSLERP) performed normalization at the layer or tensor level, substantially affecting its SuperGLUE scores.

This difference in accuracy arises from the fact that SLERP merges parameters across all layers as a whole, while NuSLERP performs the merging at the level of individual layers. In other words, SLERP preserves the global balance of interpolation and maintains a consistent meaning of  $\alpha$  throughout the model, while NuSLERP rescales each layer separately, which amplifies local variations and leads to unstable behavior when all layers are merged simultaneously.

Methods	$\alpha$	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC
pre-trained	–	0.683	0.501	0.786	0.487	0.854	0.598	0.504	0.490
linear	0.1	0.688	0.495	0.777	0.491	0.854	0.595	0.502	0.464
	0.5	0.683	0.477	0.748	0.504	0.823	0.589	0.503	0.472
karcher-mean	0.1	0.511	0.382	0.592	0.509	0.252	0.517	0.495	0.484
	0.5	0.511	0.382	0.592	0.509	0.252	0.517	0.495	0.484
slerp	0.1	0.686	0.497	0.766	0.500	0.848	0.594	0.505	0.470
	0.5	0.656	0.426	0.660	0.520	0.755	0.563	0.501	0.448
nuslerp	0.1	0.505	0.357	0.584	0.491	0.304	0.517	0.499	0.500
	0.5	0.507	0.352	0.625	0.489	0.532	0.528	0.498	0.524
ties	0.1	0.511	0.331	0.570	0.500	0.214	0.521	0.500	0.499
	0.5	0.511	0.331	0.570	0.500	0.214	0.521	0.500	0.499
della	0.1	0.565	0.277	0.573	0.513	0.272	0.538	0.501	0.490
	0.5	0.565	0.279	0.573	0.512	0.272	0.537	0.501	0.491
nearsrap	0.1	0.666	0.446	0.713	0.508	0.788	0.588	0.510	0.459
	0.5	0.666	0.446	0.713	0.508	0.788	0.588	0.510	0.459

Table 1: SuperGLUE evaluation scores on each task with pre-trained LLMs and the debiased LLMs by the model merging methods, setting a scaling factor  $\alpha$  to 0.1 or 0.5. Results highlighted in red indicate scores that are more than 15% lower than those of the pre-trained LLM.

These findings suggest that, unlike SLERP, most recent model-merging methods cannot be directly applied for bias mitigation without risking substantial losses in reasoning performance.

## 6 Conclusions and Future Works

This work presented the first comprehensive study of how seven model-merging algorithms influence social bias in LLM. By evaluating 13 models spanning the GPT, LLaMA, and Qwen families on three social bias datasets and the SuperGLUE benchmark, we revealed a trade-off between fairness and utility.

Linear, SLERP, and Nearsrap consistently mitigated stereotypical tendencies across all architectures, whereas Karcher Mean, NuSLERP, TIES, and DELLA often reduced social bias excessively, resulting in LLMs that exhibit anti-stereotypical behavior. Among the seven methods, SLERP with moderate interpolation weights ( $\alpha = 0.2$ – $0.3$ ) proved to be the most balanced approach, achieving a greater bias reduction than Linear while maintaining downstream accuracy.

Our analysis also revealed that bias reduction patterns were broadly consistent across architectures, with the notable exception of LLaMA2-7B. Trends with respect to the scaling factor  $\alpha$  also remained stable regardless of model size, suggesting that parameter scale alone does not alter the fundamental dynamics of merging.

In addition, the four methods (Karcher Mean, NuSLERP, TIES, and DELLA) substantially degraded performance on tasks requiring reading comprehension and commonsense or causal reasoning, such as ReCoRD, COPA, CB, and BoolQ in the SuperGLUE benchmark. Some existing debiasing methods based on model merging have demonstrated their debiased LLMs’ downstream-task performance using scores from NLI benchmarks. However, we revealed that it is also essential to verify accuracy on tasks for reading comprehension and commonsense / causal reasoning.

In future work, to preserve these capabilities of debiased LLMs, we plan to jointly merge models specialized for these tasks during bias mitigation via model merging.

## Ethics Statement

Navigli et al. (2023) define *bias* in natural language processing as “prejudices, stereotypes, and discriminatory attitudes against certain groups of people.” We adopt this definition throughout this paper.

For simplicity, we use the term “bias” to refer to both stereotypes and biases, while acknowledging that they are distinct concepts. We also recognize that the stereotypical data (StereoSet) used in our experiments reflect the biases of U.S. residents (Nadeem et al., 2021).

Our work specifically addressed bias mitigation in LLMs by leveraging stereotypes. Biases arise

when concepts that should not be associated with particular social groups are unfairly linked. If LLM systems exhibit such biases, they may leave a negative impression on users. Our study examines the applicability of a task-arithmetic approach to mitigate bias, with the aim of reducing LM bias using the proposed methods.

We recognize the importance of maintaining an objective position. Therefore, we emphasize that the content of this study is not influenced by any political positions, stereotypes, or biases of the authors. Our research is guided by the ethical principle of fairness in scientific inquiry and seeks to make constructive and responsible contributions to the development of AI technologies.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Elisa Bassignana, Valerio Basile, Viviana Patti, and 1 others. 2018. Hurltlex: A multilingual lexicon of words to hurt. In *CEUR Workshop proceedings*, volume 2253, pages 1–6. CEUR-WS.
- 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. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.
- Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. 2024. [Della-merging: Reducing interference in model merging through magnitude-based sampling](#). *Preprint*, arXiv:2406.11617.
- 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*, page 862–872, New York, NY, USA. Association for Computing Machinery.
- Omkar Dige, Diljot Arneja, Tsz Fung Yau, Qixuan Zhang, Mohammad Bolandraftar, Xiaodan Zhu, and Faiza Khan Khattak. 2024. [Can machine unlearning reduce social bias in language models?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 954–969, Miami, Florida, US. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. [Bias and fairness in large language models: A survey](#). *Computational Linguistics*, 50(3):1097–1179.
- Lei Gao, Yue Niu, Tingting Tang, Salman Avestimehr, and Murali Annamaram. 2024. [Ethos: Rectifying language models in orthogonal parameter space](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2054–2068, Mexico City, Mexico. Association for Computational Linguistics.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Guillaume Garnier. 2024. [Decomposing with unknown noise through several independents channels](#). *Preprint*, arXiv:2405.10588.
- Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee’s mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*.
- Karsten Grove and Hermann Karcher. 1973. How to conjugate  $c$  1-close group actions. *Mathematische Zeitschrift*, 132(1):11–20.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.
- Kyung Seo Ki, Bugeun Kim, and Gahgene Gweon. 2024. [Inspecting soundness of AMR similarity metrics in terms of equivalence and inequivalence](#). In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (\*SEM 2024)*, pages 402–409, Mexico City, Mexico. Association for Computational Linguistics.
- Sanwoo Lee, Jiahao Liu, Qifan Wang, Jingang Wang, Xunliang Cai, and Yunfang Wu. 2025. [Dynamic fisher-weighted model merging via Bayesian optimization](#). In *Proceedings of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4923–4935, Albuquerque, New Mexico. Association for Computational Linguistics.

- Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. 2023. Deep model fusion: A survey. *arXiv preprint arXiv:2309.15698*.
- Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. [Towards safer large language models through machine unlearning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1817–1829, Bangkok, Thailand. Association for Computational Linguistics.
- Marlene Lutz, Rochelle Choenni, Markus Strohmaier, and Anne Lauscher. 2024. [Local contrastive editing of gender stereotypes](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21474–21493, Miami, Florida, USA. Association for Computational Linguistics.
- Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.
- Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. [StereoSet: Measuring stereotypical bias in pretrained language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.
- Roberto Navigli, Simone Conia, and Björn Ross. 2023. [Biases in large language models: Origins, inventory, and discussion](#). *J. Data and Information Quality*, 15(2).
- Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. [HONEST: Measuring hurtful sentence completion in language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online. Association for Computational Linguistics.
- Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. Bbq: A hand-built bias benchmark for question answering. *Findings of the Association for Computational Linguistics: ACL 2022*.
- Zhanyue Qin, Yue Ding, Deyuan Liu, Qingbin Liu, Junxian Cai, Xi Chen, Zhiying Tu, Dianhui Chu, Cuiyun Gao, and Dianbo Sui. 2025. Lftf: Locating first and then fine-tuning for mitigating gender bias in large language models. *arXiv preprint arXiv:2505.15475*.
- Team Qwen. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. [Null it out: Guarding protected attributes by iterative nullspace projection](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.
- Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. [Gender bias in coreference resolution](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.
- Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. [Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp](#). *Transactions of the Association for Computational Linguistics*, 9:1408–1424.
- Daiki Shirafuji, Makoto Takenaka, and Shinya Taguchi. 2025. [Bias vector: Mitigating biases in language models with task arithmetic approach](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2799–2813, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ken Shoemake. 1985. [Animating rotation with quaternion curves](#). *SIGGRAPH Comput. Graph.*, 19(3):245–254.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrusti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. [Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: resolving interference when merging models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jixi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. *Qwen2.5 technical report*. Preprint, arXiv:2412.15115.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

## A Model List

In this section, we show the model list and these URLs available in HuggingFace repositories.

- GPT2-small: <https://huggingface.co/openai-community/gpt2>,
- GPT2-medium: <https://huggingface.co/openai-community/gpt2-medium>,
- GPT2-large: <https://huggingface.co/openai-community/gpt2-large>,
- GPT2-xl: <https://huggingface.co/openai-community/gpt2-xl>,
- GPT-neo-2.7B: <https://huggingface.co/EleutherAI/gpt-neo-2.7B>,
- LLAMA-2-7B: <https://huggingface.co/meta-llama/Llama-2-7b-hf>,
- LLAMA-3-8B: <https://huggingface.co/Undi95/Meta-Llama-3-8B-hf>,
- LLAMA-3.1-8B: <https://huggingface.co/meta-llama/Llama-3.1-8B>,
- LLAMA-3.2-1B: <https://huggingface.co/meta-llama/Llama-3.2-1B>,
- LLAMA-3.2-3B: <https://huggingface.co/meta-llama/Llama-3.2-3B>,
- QWEN2-0.5B: <https://huggingface.co/Qwen/Qwen2-0.5B>,
- QWEN2-1.5B: <https://huggingface.co/Qwen/Qwen2.5-1.5B>,
- QWEN2-7B: <https://huggingface.co/Qwen/Qwen2-7B>.

## B Preliminary Experiments

This section describes preliminary experiments conducted to narrow down the appropriate range for the hyperparameter  $\alpha$ .

We first evaluated GPT-based models using HONEST and GLUE (Wang et al., 2018) in advance to determine the effective range of  $\alpha$ . In this experiment,  $\alpha$  was set to 0.1, 0.2, 0.5, 1, 2, 5, and 10, and the model merging method followed the approach proposed by Shirafuji et al. (2025).

The experimental results of HONEST with  $K = 20$  are shown in Figure 6, and the results of the

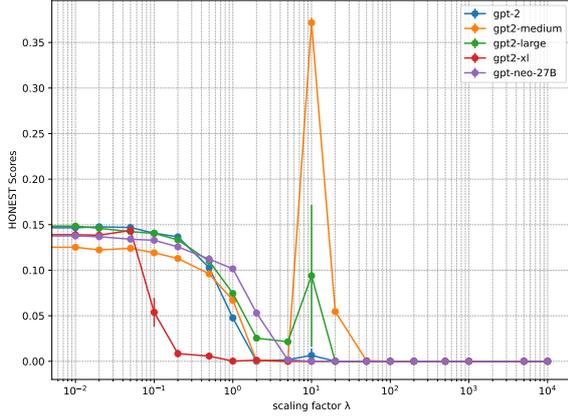


Figure 6: HONEST evaluation results of GPT model families on Preliminary experiments.

Methods	cola	avg.
GPT2-small	<b>0.449</b>	<b>0.760</b>
w/ Bias Vector ( $\alpha = 0.1$ )	0.396	0.754
w/ Bias Vector ( $\alpha = 0.2$ )	0.440	0.759
w/ Bias Vector ( $\alpha = 0.5$ )	0.362	0.754
w/ Bias Vector ( $\alpha = 1$ )	0.050	0.702
w/ Bias Vector ( $\alpha = 2$ )	0.012	0.705
w/ Bias Vector ( $\alpha = 5$ )	0.000	0.669
w/ Bias Vector ( $\alpha = 10$ )	0.016	0.590

Table 2: GLUE evaluation results of GPT2-small on the preliminary experiments.

evaluation of GPT2-small in GLUE are presented in Table 2. From the evaluation, we found that for values of  $\alpha$  around 5, bias was nearly eliminated for all models. However, in certain downstream tasks (COLA), performance began to gradually degrade from  $\alpha = 0.5$  and dropped to almost zero at  $\alpha = 1$ .

Based on these results, the main experiments in this paper restrict  $\alpha$  to the range of 0.1 to 0.5.

## C Computational Environment

All LLM training for the stereotypical bias experiments was performed on AWS p4d.24xlarge instances, each equipped with eight NVIDIA H100 GPUs. Models with up to 3 billion parameters were trained on four H100 GPUs, while larger models used all eight.

For the evaluation experiments on SuperGLUE, BBQ, BOLD, and HONEST, all runs – except those for the GPT-based model family – were conducted on NVIDIA H100 GPUs: models with up to 3 billion parameters used a single GPU for inference and scoring, and larger models were allocated two GPUs. GPT-based models were evaluated on an

Model	lr	scheduler
GPT2-small	3e-5	linear
GPT2-medium	3e-5	linear
GPT2-large	2e-5	linear
GPT2-xl	1e-5	linear
GPT2-neo-2.7B	1e-5	linear
LLAMA-2-7B	1e-5	cosine
LLAMA-3-8B	1e-5	cosine
LLAMA-3.1-8B	1e-5	cosine
LLAMA-3.2-1B	2e-5	cosine
LLAMA-3.2-3B	1e-5	cosine
QWEN2-0.5B	1e-4	cosine
QWEN2-1.5B	2e-5	cosine
QWEN2-7B	1e-5	cosine

Table 3: Hyperparameter configurations for LLM training. “lr” denotes the learning rate, and “scheduler” indicates the learning rate scheduling strategy.

NVIDIA Quadro RTX 8000.

## D Hyperparameter Configurations

The experimental setup for continual learning is designed as follows. We utilize the HuggingFace AutoModelForCausalLM library for model training. To reduce GPU memory consumption, the maximum sequence length (max\_length) is set to 512, the batch size is set to 64. Training is carried out for 30 epochs with a weight decay of 0.01 and a warm-up ratio of 0.1.

The hyperparameters specific to each model, namely the learning rate and the learning rate scheduler, are described in Table 3.

Note that the scheduler was set to linear for the GPT family but cosine for the other models, since we followed the configuration of Shirafuji et al. (2025), which established the linear scheduler as the default choice for GPT.

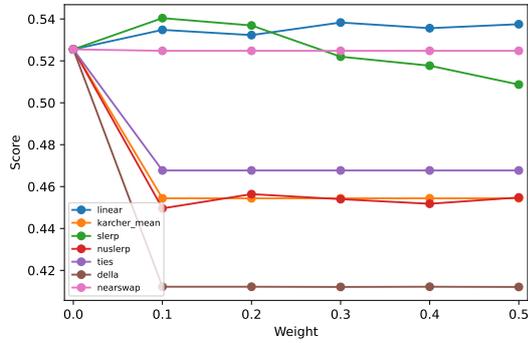
## E List of Evaluation Datasets

The URLs of the social bias evaluation datasets are listed as follows:

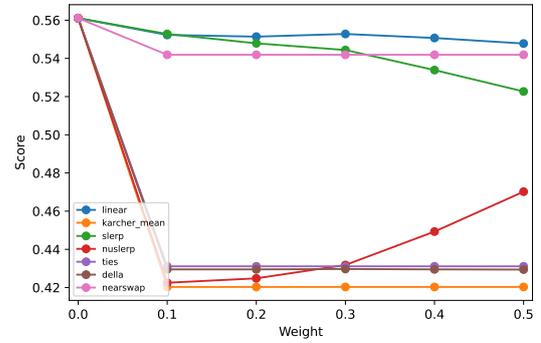
- BBQ: <https://huggingface.co/datasets/heegy/bbq>;
- BOLD: <https://huggingface.co/datasets/AmazonScience/bold>;
- HONEST: <https://huggingface.co/datasets/MilaNLPProc/honest>.

## **F Each LLM Result on SuperGLUE, BBQ, BOLD, and HONEST**

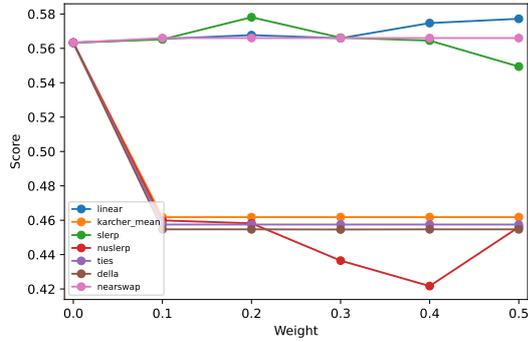
This section shows the results of each LLM evaluated with SuperGLUE, BBQ, BOLD, and HONEST benchmarks. The results are shown in Figure 7 (GPT on SuperGLUE), 8 (LLAMA on SuperGLUE), 9 (Qwen on SuperGLUE), 10 (GPT on BBQ), 11 (LLAMA on BBQ), 12 (Qwen on BBQ), 13 (GPT on BOLD), 14 (LLAMA on BOLD), 15 (Qwen on BOLD), 16 (GPT on HONEST), 17 (LLAMA on HONEST), and 18 (Qwen on HONEST).



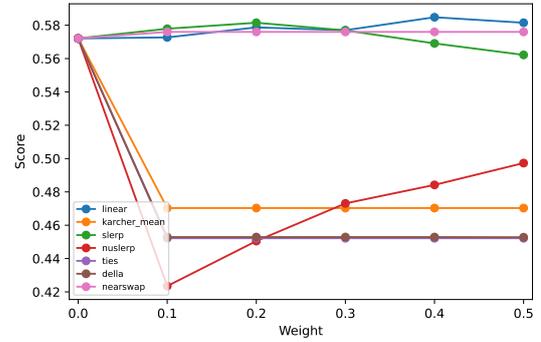
(a) GPT2-small



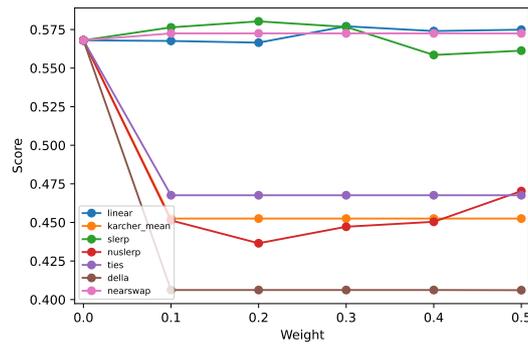
(b) GPT2-medium



(c) GPT2-large

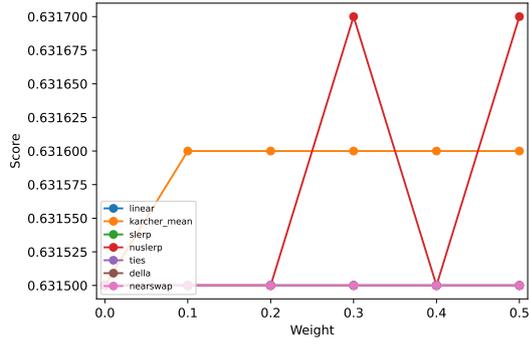


(d) GPT2-XL

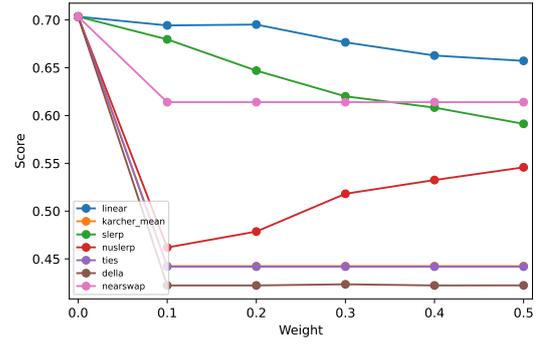


(e) GPT-Neo-2.7B

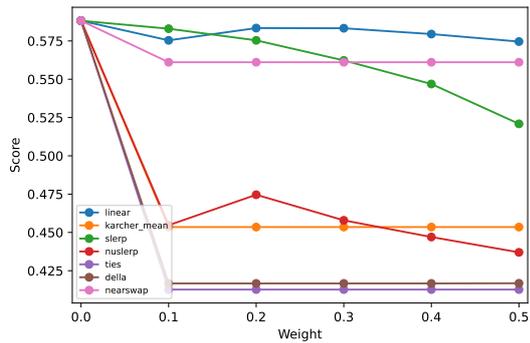
Figure 7: The SuperGLUE evaluation results in GPT models. The blue, orange, green, red, purple, brown, and pink lines correspond to the results for Linear, Karcher Mean, SLERP, NuSLERP, TIES, DELLA, and Nearswap, respectively. The scores of setting the weight  $\alpha$  to zero are resulted using the pre-trained LLMs.



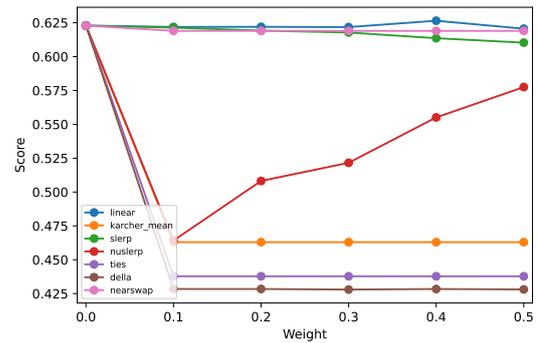
(a) LLAMA2-7B



(b) LLAMA-3.1-8B

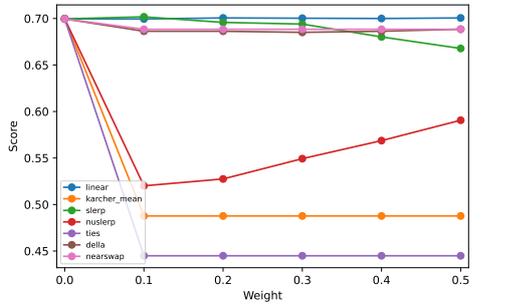


(c) LLAMA-3.2-1B

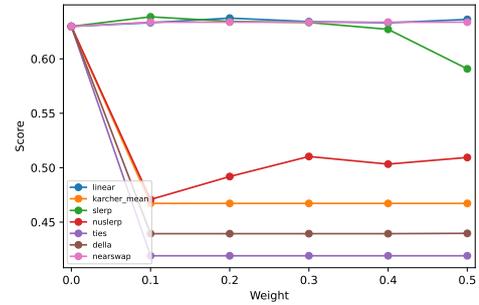


(d) LLAMA-3.2-3B

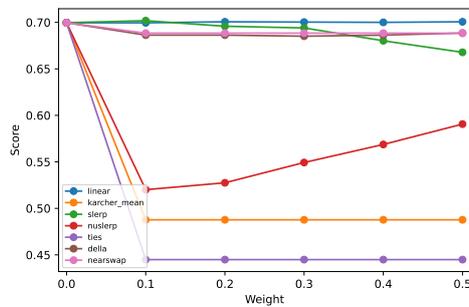
Figure 8: The SuperGLUE evaluation results in LLAMA models.



(a) QWEN2-0.5B

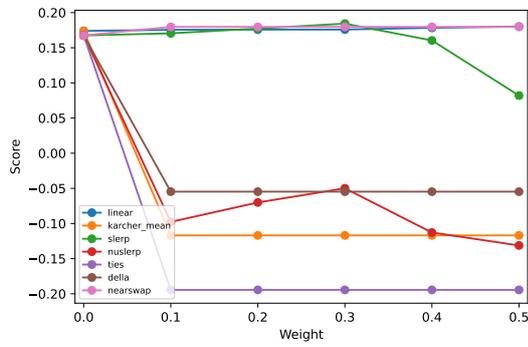


(b) QWEN2-1.5B

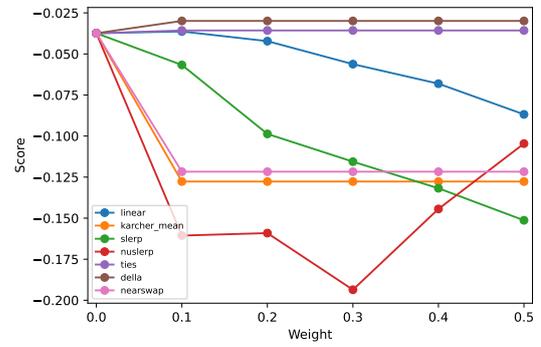


(c) QWEN2-7B

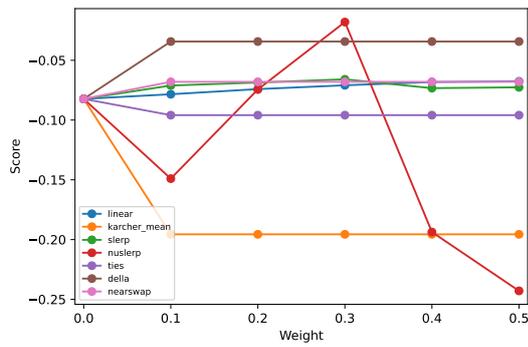
Figure 9: The SuperGLUE evaluation results in QWEN models.



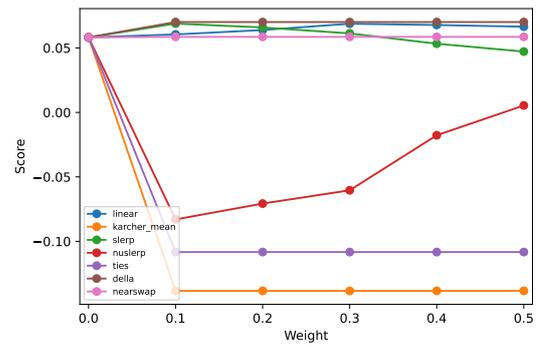
(a) GPT2-small



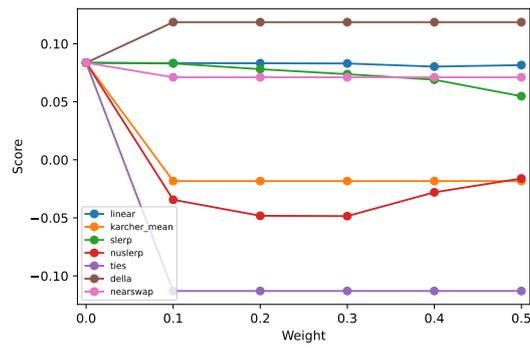
(b) GPT2-medium



(c) GPT2-large

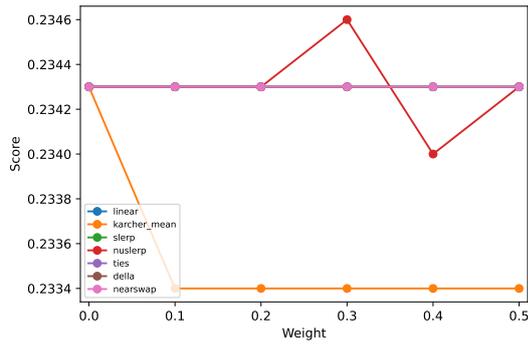


(d) GPT2-XL

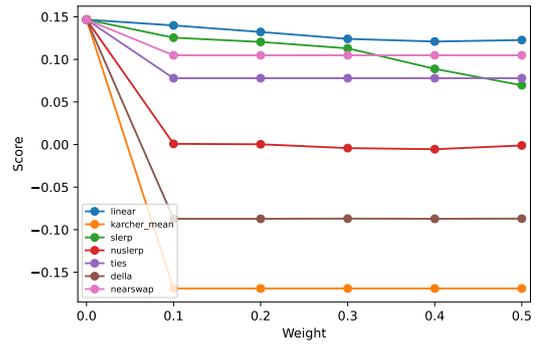


(e) GPT-Neo-2.7B

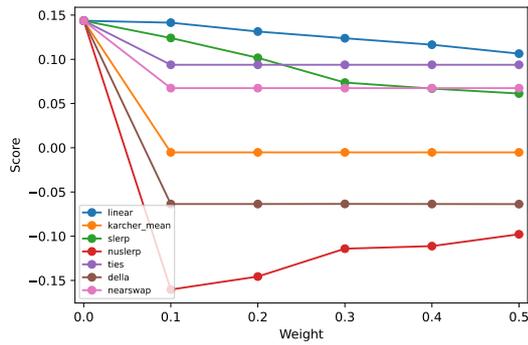
Figure 10: The BBQ evaluation results in GPT models.



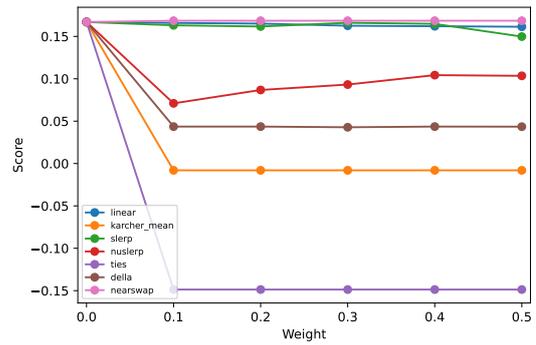
(a) LLAMA2-7B



(b) LLAMA-3.1-8B

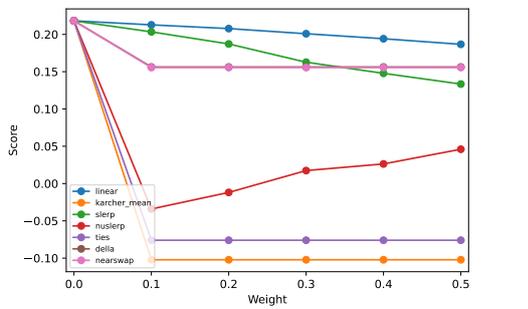


(c) LLAMA-3.2-1B

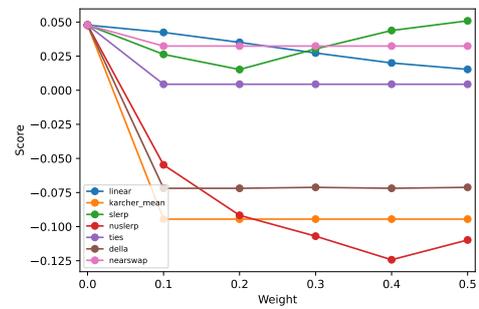


(d) LLAMA-3.2-3B

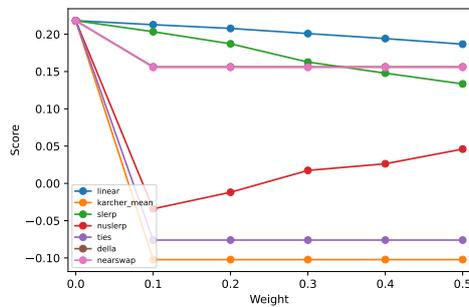
Figure 11: The BBQ evaluation results in LLAMA models.



(a) QWEN2-0.5B

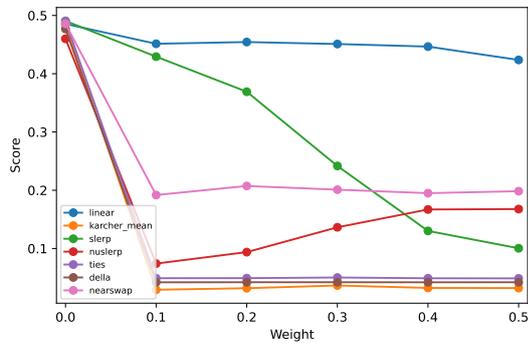


(b) QWEN2-1.5B

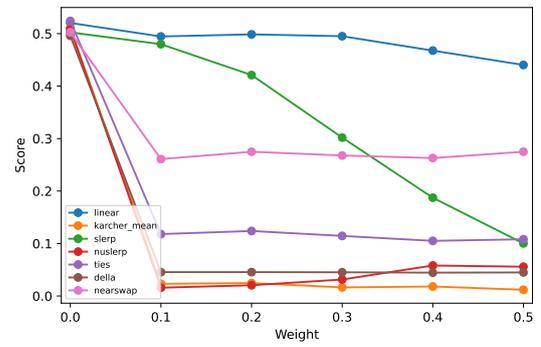


(c) QWEN2-7B

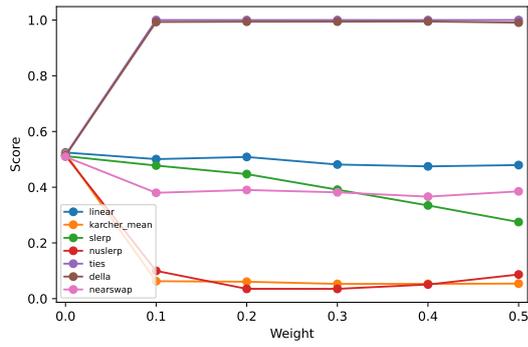
Figure 12: The BBQ evaluation results in QWEN models.



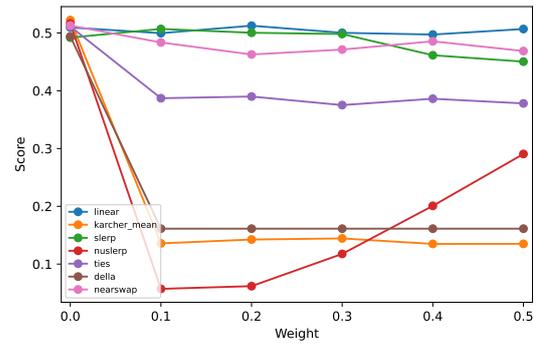
(a) GPT2-small



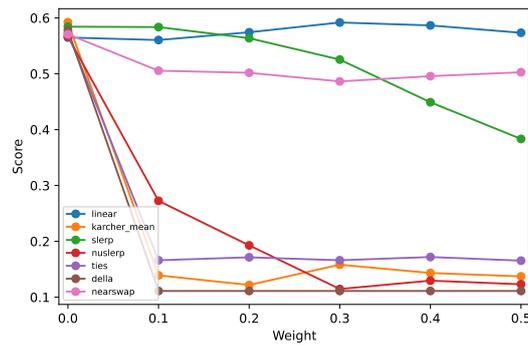
(b) GPT2-medium



(c) GPT2-large

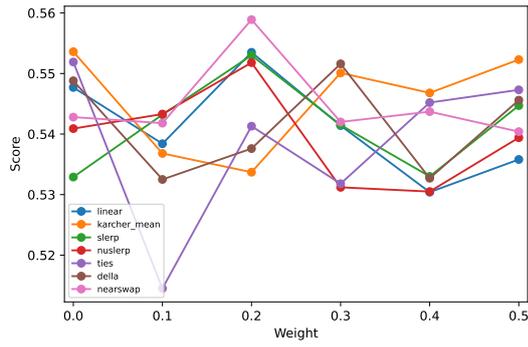


(d) GPT2-XL

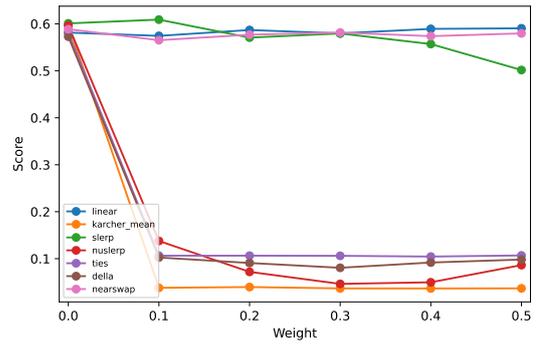


(e) GPT-Neo-2.7B

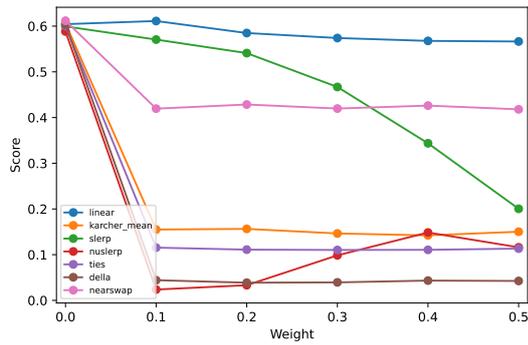
Figure 13: The BOLD evaluation results in GPT models.



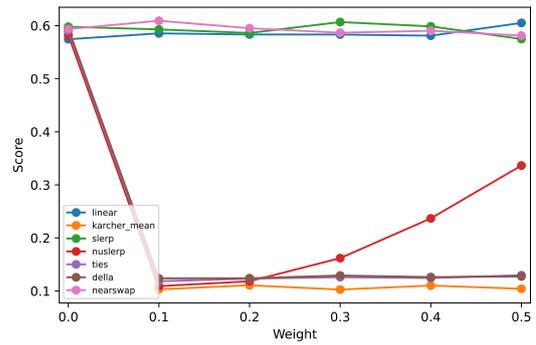
(a) LLAMA2-7B



(b) LLAMA-3.1-8B

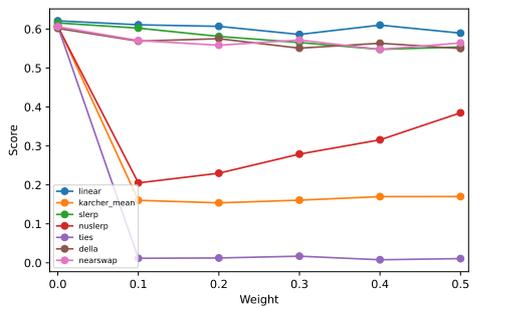


(c) LLAMA-3.2-1B

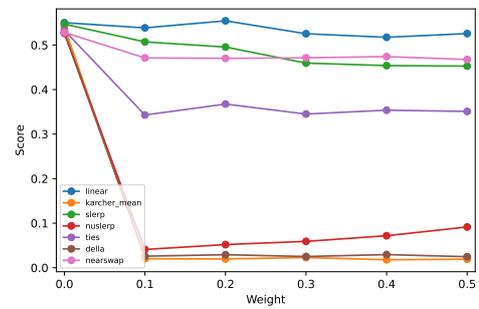


(d) LLAMA-3.2-3B

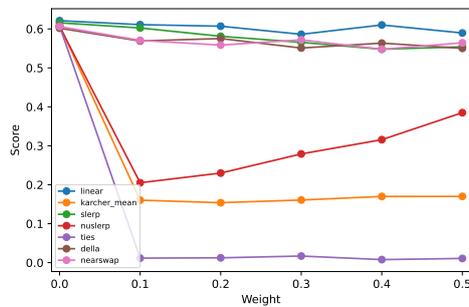
Figure 14: The BOLD evaluation results in LLAMA models.



(a) QWEN2-0.5B

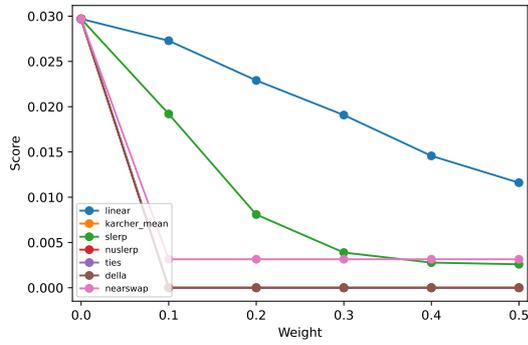


(b) QWEN2-1.5B

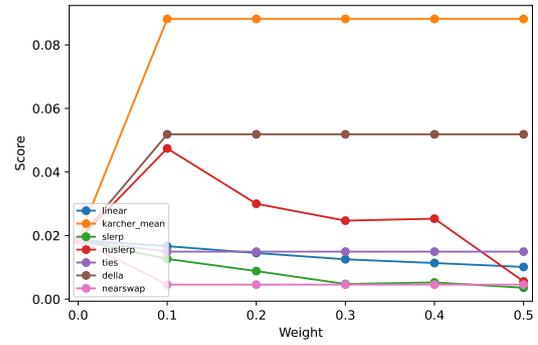


(c) QWEN2-7B

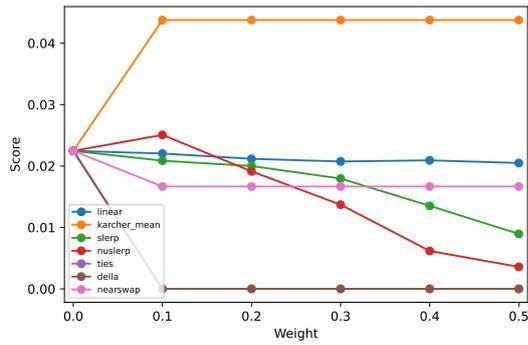
Figure 15: The BOLD evaluation results in QWEN models.



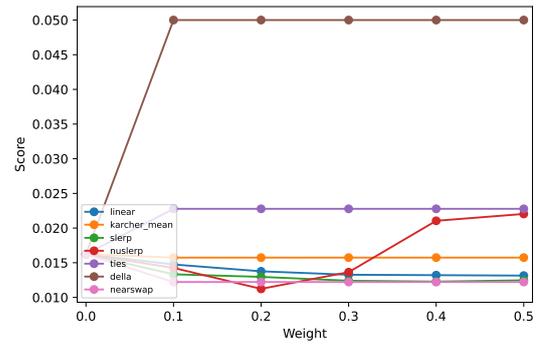
(a) GPT2-small



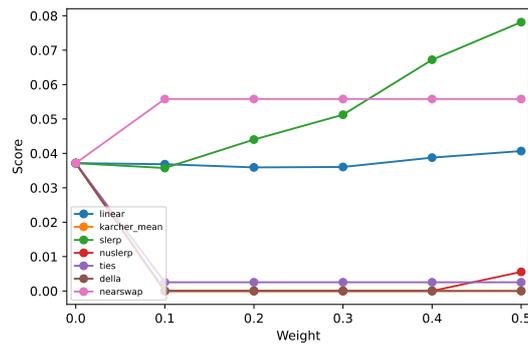
(b) GPT2-medium



(c) GPT2-large

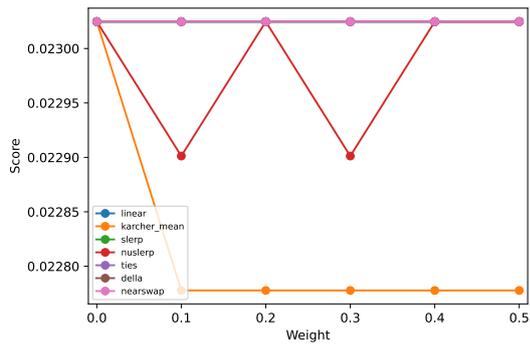


(d) GPT2-XL

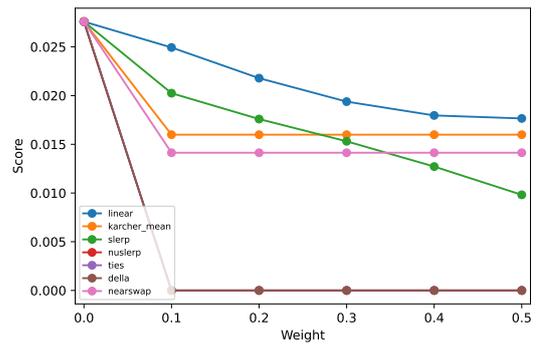


(e) GPT-Neo-2.7B

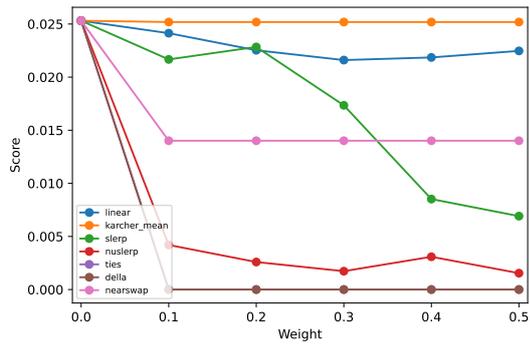
Figure 16: The HONEST evaluation results in GPT models.



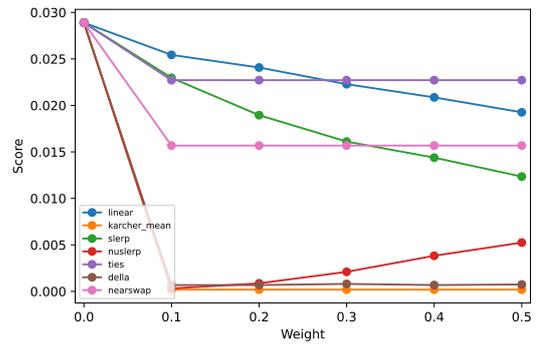
(a) LLAMA2-7B



(b) LLAMA-3.1-8B

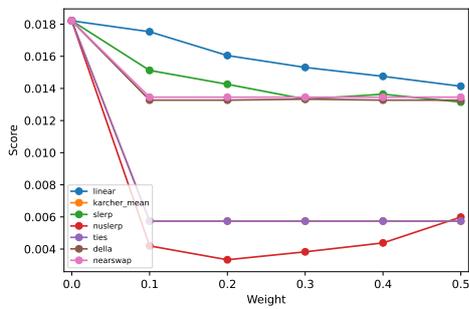


(c) LLAMA-3.2-1B

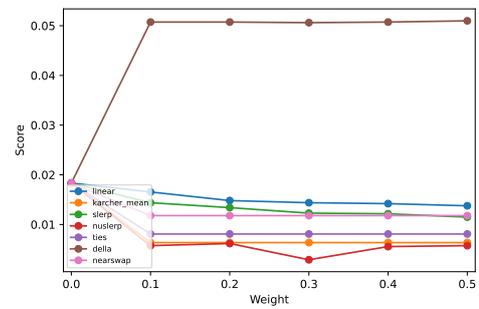


(d) LLAMA-3.2-3B

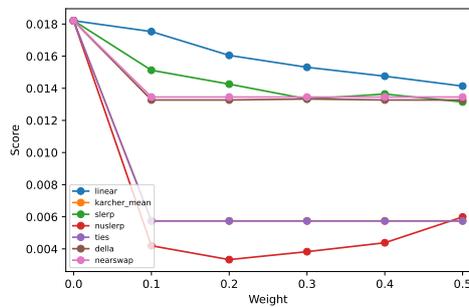
Figure 17: The HONEST evaluation results in LLAMA models.



(a) QWEN2-0.5B



(b) QWEN2-1.5B



(c) QWEN2-7B

Figure 18: The HONEST evaluation results in QWEN models.