Review

A Survey on Selection Bias in Large Language Models

Guoxiu He *, Jinquan Zheng, Fangqing Han

Posted Date: 1 May 2026

doi: 10.20944/preprints202604.2234.v1

Keywords: large language models; selection bias; position bias; label bias; LLM-as-a-judge; bias mitigation

 doi:10.20944/preprints202604.2234.v1

Review

A Survey on Selection Bias in Large Language Models

Guoxiu He *, Jinquan Zheng and Fangqing Han

School of Economics and Management, East China Normal University, Shanghai, China
* Correspondence: gxhe@fem.ecnu.edu.cn

Abstract

Selection bias in Large Language Models has emerged as a fundamental obstacle to reliability, fairness, and robustness. Defined operationally as systematic decision changes under equivalence-preserving input perturbations, including option permutation, label renaming, candidate-order swapping, and evidence relocation, the phenomenon is examined across four representative task families: multiple-choice question answering, in-context classification, LLM-as-a-Judge evaluation, and long-context or retrieval-augmented generation. Selection bias is first analyzed through a causal chain that links biased behavior to training-data priors, architectural asymmetries, and post-training amplification. Existing mitigation methods are then synthesized through an intervention-level taxonomy spanning inference-time calibration and prompt optimization, architecture-level modification, and training-level debiasing. The evaluation landscape is unified by summarizing commonly used metrics, benchmark families, and application settings, with the lack of standardized and cross-task-comparable protocols identified as a central bottleneck. Selection bias is best understood as a failure of invariance under non-semantic reformatting, and mitigating it is essential for trustworthy, robust, and selection-invariant language models.

Keywords: large language models; selection bias; position bias; label bias; LLM-as-a-judge; bias mitigation

1. Introduction

In recent years, natural language processing (NLP) technologies represented by large language models (LLMs) have made substantial progress [1–3]. Their capabilities in comprehension, generation, and reasoning have made them a central component in the development of contemporary artificial intelligence, as exemplified by frontier systems and large-scale instruction-tuned model families [4,5]. These models are typically pre-trained on massive text corpora and subsequently instruction-tuned to follow human directives, enabling them to demonstrate strong in-context learning (ICL) capabilities even on unseen tasks without further task-specific training [6–13]. From open-domain dialogue and code generation to complex logical reasoning, the applications of LLMs have improved productivity and reshaped many human–computer interaction workflows.

However, as LLMs are increasingly used in academic research and industrial applications, their vulnerabilities and limitations have become more apparent. A widely observed phenomenon is that LLM performance is highly sensitive to the prompts they receive, a property often referred to as prompt brittleness [14–16]. Even trivial modifications to a prompt, such as reordering examples, rearranging multiple-choice options, or altering punctuation, can lead to substantial or even contradictory variations in model outputs [7,17–25].

 doi:10.20944/preprints202604.2234.v1

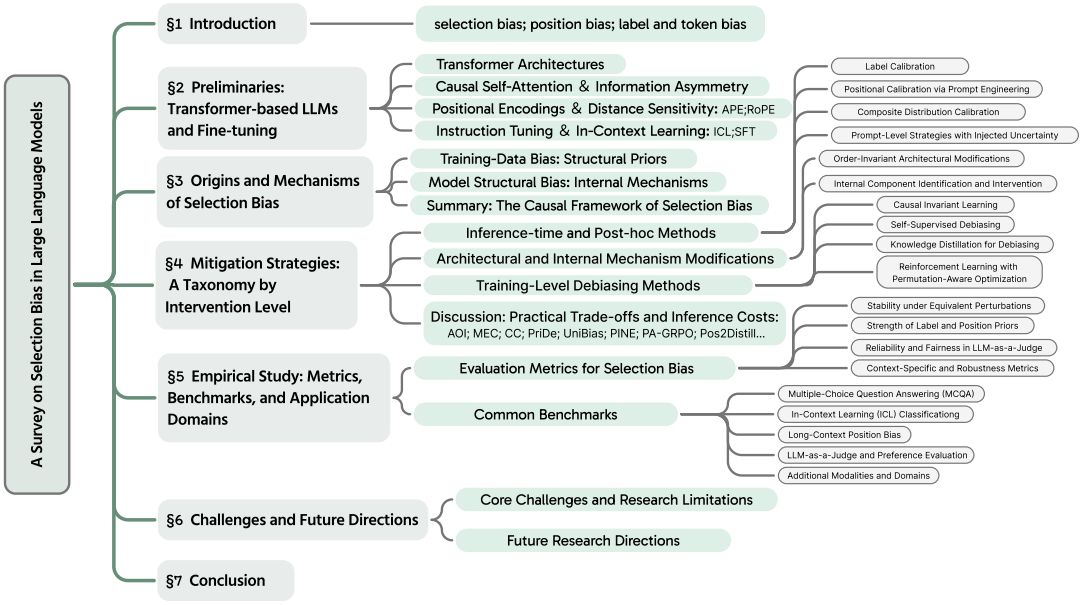

Figure 1. Overview of the survey structure. A hierarchical outline of the survey. The article contains seven sections: Introduction; preliminaries on Transformer-based LLMs and fine-tuning; origins and mechanisms of selection bias; mitigation strategies organized by intervention level; empirical study of metrics, benchmarks, and application domains; challenges and future directions; and conclusion. The outline also summarizes major topics, including Transformer architectures, causal self-attention, positional encodings, instruction tuning, training-data priors, model-internal mechanisms, inference-time calibration, architectural intervention, training-level debiasing, evaluation metrics, common benchmarks, and application domains such as MCQA, ICL classification, long-context position bias, LLM-as-a-Judge evaluation, and additional modalities and domains.

Among these manifestations of brittleness, selection bias is particularly important because it directly affects decision-making in tasks that require a model to choose among alternatives. In this survey, selection bias refers to systematic changes in model decisions caused by equivalence-preserving input perturbations. Under such perturbations, the semantic content of the task remains unchanged, while non-semantic presentation factors such as option order, option labels, candidate positions, or evidence locations are modified. A robust model should preserve its decision under these changes. When the output varies systematically, the model is influenced by task-irrelevant factors rather than by the content required for the decision [14,21–23,26,27]. Typical manifestations include position bias, where the model prefers options or evidence at particular locations, and label bias, where the model favors choices denoted by labels such as “A” or “1”. Such phenomena have been widely observed in multiple-choice question answering (MCQA), LLM-as-a-Judge evaluation, in-context classification, and long-context reasoning. Selection bias therefore provides a useful lens for examining whether current LLMs can support robust reasoning and reliable decision-making [28–31].

1.1. Motivation and Significance

The study of selection bias in LLMs is motivated by its direct implications for evaluation validity, reasoning validity, and deployment robustness. These three dimensions are closely related, but they correspond to distinct risks in the use of LLM systems.

First, selection bias compromises the fairness and reliability of model evaluation, thereby weakening the foundation on which technological progress is assessed. As LLM capabilities continue to grow, the research community has increasingly adopted the LLM-as-a-Judge paradigm, in which a strong model is used to assess the quality of outputs produced by other models [28–30,32]. Owing to its scalability and relatively low cost, this approach is often viewed as a practical supplement to expensive and time-consuming human evaluation [28,32–34]. Yet, extensive empirical evidence indicates that even advanced models remain susceptible to position bias. When tasked with pairwise comparison

 doi:10.20944/preprints202604.2234.v1

between two candidate answers, merely swapping their presentation order can lead the model to produce opposite judgments. Such inconsistency implies that evaluation results no longer reflect only the intrinsic quality of the content being assessed, but are also affected by irrelevant positional noise. Comparisons across models therefore become less reliable, and the evaluation of alignment techniques such as reinforcement learning from human feedback (RLHF) can be compromised. The presence of substantial positional bias in strong LLM-based evaluators also motivates the use of stronger and sometimes proprietary judges for more stable automatic evaluation [29,30,35], which raises research costs, limits accessibility, and slows open comparison across systems. Addressing selection bias is therefore important for building scientific and equitable evaluation systems and for supporting the healthy development of the LLM field.

Second, selection bias raises fundamental questions about the reliability of LLM reasoning. Scholars continue to debate whether LLMs engage in human-like, rule-based reasoning, or whether their apparent reasoning reflects complex statistical pattern matching learned from massive datasets [36]. A reliable reasoner should produce stable inferences that remain invariant under superficial changes in problem formulation or irrelevant contextual shifts. However, studies show that LLM performance on classic logical fallacy problems, such as conjunction or syllogistic fallacies, is strongly correlated with the presence of certain high-frequency biased tokens in the problem statement. This suggests that models may rely on lexical or formatting shortcuts rather than on the underlying logical structure [1]. Systematically investigating selection bias therefore provides a controlled setting for distinguishing robust reasoning from spurious pattern learning.

Third, selection bias directly constrains real-world deployment, especially in settings where models must integrate dispersed evidence across long contexts or multiple interaction turns. In long-context understanding and retrieval-augmented generation (RAG), LLMs exhibit a pronounced lost-in-the-middle phenomenon [37–43]: even when crucial evidence is present, models tend to underweight information located in the middle of the context while over-attending to content near the beginning or the end [44–46]. As a result, performance in multi-document question answering, long-form summarization, and retrieval-intensive reasoning can degrade substantially as context length grows. Related problems also arise in conversational systems, where recency bias causes models to overemphasize the latest turns while neglecting earlier instructions, commitments, or causal information, leading to shallow, generic, or contextually inconsistent responses [38,47,48]. These deployment failures show that selection bias is not merely an evaluation artifact. It is a practical obstacle to building LLM systems that are robust, faithful, and dependable in real-world use.

1.2. Survey Scope and Selection Criteria

This survey focuses on selection bias in LLMs under the operational definition introduced above: systematic changes in model decisions caused by equivalence-preserving input perturbations, such as option permutation, label renaming, candidate-order swapping, or relocating relevant evidence within a long context. Under this definition, the semantic content of the task remains unchanged, and any output variation indicates sensitivity to non-semantic presentation factors rather than task understanding [14,22,23,38].

Accordingly, the survey primarily covers four task families in which such invariance can be clearly diagnosed: multiple-choice question answering (MCQA), in-context classification and verbalizer-sensitive prompting, LLM-as-a-Judge and pairwise preference evaluation, and long-context or retrieval-augmented generation settings. These task families are selected because they allow controlled equivalence-preserving perturbations and therefore make selection bias empirically identifiable and comparable. We include studies that identify or analyze position-, label-, token-, or order-related bias, propose mitigation methods at the prompting, calibration, architectural, or training level, or introduce metrics and benchmarks for evaluating such bias [14,22,28,38].

To maintain conceptual clarity, we do not treat all forms of bias discussed in the broader LLM literature as part of the core scope of this survey. In particular, social stereotype bias, toxicity-related bias, and the notion of selection bias in recommender systems or causal inference are considered adja-

 doi:10.20944/preprints202604.2234.v1

cent but distinct research areas unless they directly study invariance violations induced by equivalent input reformatting. Likewise, we do not attempt to exhaustively review generic prompt sensitivity phenomena unless they are explicitly connected to choice-format or position-dependent selection behavior.

Our organizational principle is therefore mechanistic and intervention-oriented. We review how selection bias arises, how it is measured, and how it can be mitigated at different stages of the LLM pipeline, from prompts and calibration layers to internal representations and training objectives.

1.3. Contributions of This Survey

This survey makes four main contributions to the study of selection bias in LLMs. First, it clarifies the operational definition and conceptual boundary of selection bias by framing it as a failure of invariance under equivalence-preserving input perturbations. Second, it develops a mechanistic account that connects biased behavior to training-data priors, architectural asymmetries, and post-training amplification. Third, it organizes the mitigation literature through an intervention-level taxonomy spanning inference-time, architectural, and training-level methods. Fourth, it synthesizes the evaluation landscape by summarizing commonly used metrics, benchmark families, and representative application settings, and it identifies major open challenges, including fragmented evaluation protocols, limited cross-task comparability, and incomplete mechanistic understanding.

1.4. Core Concepts and Bias Types

1.4.1. Selection Bias

Selection bias in LLMs denotes a systematic violation of decision invariance under semantically equivalent reformulations. In an ideal choice-based decision process, if the underlying task content and the set of candidate answers remain unchanged, the model's decision should remain stable after changes to option order, option labels, prompt format, or evidence location. The examples shown in Figure 2 illustrate this choice-level phenomenon. When the high-quality answer is relabeled or moved to a different position, a selection-invariant model should continue to select it. If the model instead continues to output the same label or the same position, the decision reflects a prior preference for surface form rather than an assessment of answer quality.

Selection bias is therefore a unifying concept that covers several related but distinguishable forms of input sensitivity. Option-order bias, label bias, token bias, and context-location bias all reflect the same underlying failure to learn task invariances, such as permutation invariance over options. These failures may originate from statistical imbalances in training data, architectural asymmetries in sequence processing, and post-training procedures that amplify shortcuts already present in the model [14,15,23].

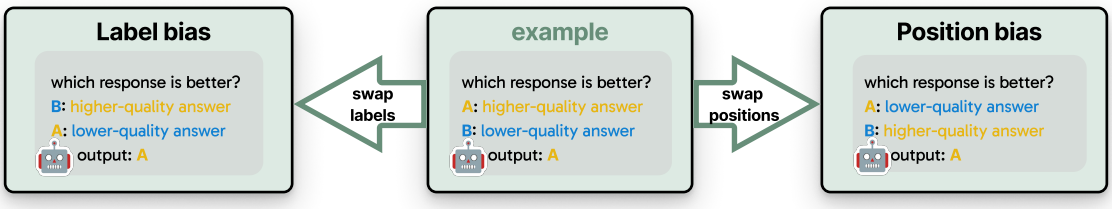

Figure 2. Illustration of label bias and position bias in choice-format evaluation. Three side-by-side prompt boxes illustrating selection bias. The center box shows a prompt asking which response is better, where option A is a higher-quality answer, option B is a lower-quality answer, and the model outputs A. The left box shows label bias: after the labels are swapped, option B is the higher-quality answer and option A is the lower-quality answer, but the model still outputs A. The right box shows position bias: after the positions are swapped, option A is the lower-quality answer appearing first and option B is the higher-quality answer appearing second, but the model still outputs A.

 doi:10.20944/preprints202604.2234.v1

1.4.2. Position Bias

Position bias refers to systematic variation in a model's preferences for information or options depending on their relative position within the input sequence [14,30,38]. It appears in at least two settings that should be distinguished. At the option level, the model may prefer the first, last, or otherwise salient candidate in a list, even when the content of the candidates is unchanged. This includes primacy and recency effects in pairwise or multiple-choice settings, where merely swapping the order of candidate answers can reverse the evaluation outcome [35]. At the context level, the model may assign unequal importance to evidence depending on where it appears in a long input. The lost-in-the-middle phenomenon is a representative case: LLMs tend to underweight information in the middle of long contexts, producing a U-shaped performance curve [38,39]. We analyze the architectural causes of these effects in Section 3.

1.4.3. Label Bias and Token Bias

Label bias and token bias are closely related but not identical. In this survey, token bias refers to the broader tendency of an LLM to rely on particular lexical items or surface forms as shortcuts, even when those tokens are not semantically justified by the task. Label bias is a more specific case in which the privileged tokens are the identifiers used to denote answer options, such as "A", "B", "C" or "i", "ii", "iii". We therefore treat label bias as a structured subtype of token bias that is especially salient in choice-format settings and can be cleanly diagnosed through equivalence-preserving perturbations such as label renaming or option relabeling. More broadly scoped lexical shortcut effects are considered adjacent to this survey unless they directly distort selection behavior under semantically equivalent reformulations [14,20,49–51].

Label bias occurs when the model demonstrates prior preferences for specific characters or words used to denote options. Such tendencies may stem from frequency imbalances in pretraining corpora, where certain symbols or verbalizers appear more often than others, leading the model to favor them even in the absence of task-relevant evidence. Existing calibration methods generally estimate and neutralize such content-independent label priors before prediction [19,49,50,52,53]. Token bias is broader: the model may overfit to specific words or phrases in the prompt, question, or options, using them as superficial cues in place of genuine reasoning. For example, in classical logical fallacy questions, the model may rely on high-frequency signal words rather than the underlying logical structure to produce an answer [14]. Within the scope of this survey, token bias is discussed primarily when such lexical priors induce or amplify selection bias under equivalent reformulations.

1.4.4. Composite Selection Bias

Composite selection bias refers to cases in which label priors, option-order effects, positional salience, and lexical shortcuts jointly influence a model's decision. In such cases, the observed instability cannot be attributed to a single source of bias. For example, a model may prefer the first option and also prefer the label "A", making it difficult to determine whether a biased decision is primarily driven by position, label identity, or their interaction. This composite perspective is important for evaluating and mitigating selection bias because many real prompts combine several non-semantic cues at once. It also motivates mitigation strategies that consider the joint distribution of labels, positions, and candidate content rather than treating each bias type independently.

1.5. Paper Organization

The rest of this survey is organized as follows. Section 2 reviews the basic architecture of Transformer-based LLMs and adaptation paradigms. Section 3 discusses the data and architectural origins of selection bias. Section 4 summarizes mitigation strategies and their practical trade-offs. Section 5 presents evaluation metrics, benchmarks, and key application domains. Section 6 outlines open challenges and future directions, and Section 7 concludes. To provide a compact overview of the literature landscape before entering these sections, Table 1 summarizes representative studies on selection bias in LLMs by theme, task setting, bias type, core idea, and survey coverage.

 doi:10.20944/preprints202604.2234.v1

Table 1. Representative Studies on Selection Bias in Large Language Models.

Ref.	Theme	Task setting	Bias type	Core idea	Survey coverage
[14]	Foundations	Choice tasks / perturbation analysis	Selection bias	Frames selection bias through content-equivalent perturbations and analyzes order/token sensitivity.	Definition; mitigation; evaluation
[22]	Foundations	MCQA	Position bias	Shows strong option-order sensitivity in multiple-choice prediction.	Bias types; MCQA evaluation
[19]	Label calibration	Few-shot classification	Label bias	Introduces contextual calibration to remove content-independent label priors.	Inference-time calibration
[35]	Positional calibration	LLM-as-a-Judge / pairwise evaluation	Position bias	Provides BPC, MEC, and HITLC as prompt- and calibration-based judge debiasing strategies.	Positional calibration; judge evaluation
[28]	Composite calibration	LLM-as-a-Judge	Composite selection bias	CalibraEval learns a permutation-invariant calibration function without gold labels.	Composite calibration; judge evaluation
[38]	Long-context bias	Long-context QA / retrieval	Position bias	Establishes the canonical lost-in-the-middle phenomenon and the U-shaped context-usage pattern.	Position bias; long-context evaluation
[54]	Architectural invariance	Order-sensitive inputs	Position bias	PINE reduces position bias through dynamic position reassignment and bidirectional option attention.	Mechanisms; architectural mitigation; long-context evaluation
[55]	Architectural invariance	Ordered-choice inputs	Position bias	Set-based prompting enforces order invariance with identical positional encodings and masked inter-option communication.	Architectural mitigation
[7]	Internal intervention	White-box LLM debiasing	Composite selection bias	UniBias identifies and masks biased attention heads and FFN components through mechanistic analysis.	Internal mechanism intervention
[56]	Training-level debiasing	Fine-tuning	Composite selection bias	Causal-Debias suppresses non-causal cues through invariant risk minimization during fine-tuning.	Training-level debiasing
[57]	Training-level debiasing	Multiple NLP tasks	Position bias	SOD uses self-supervised responses and multi-objective optimization for position debiasing.	Self-supervised debiasing
[58]	Training-level debiasing	Long-context retrieval / reasoning	Position bias	Pos2Distill transfers knowledge from advantageous to disadvantaged positions and distinguishes token-shifting from thought-shifting.	Knowledge distillation; long-context evaluation
[59]	Training-level debiasing	MCQA and LLM-as-a-Judge	Composite selection bias	PA-GRPO internalizes permutation consistency through cross-permutation advantage and consistency-aware rewards.	Reinforcement learning; MCQA and judge evaluation

2. Preliminaries: Transformer-Based LLMs and Fine-Tuning

2.1. Transformer Architectures

The Transformer architecture serves as the fundamental backbone for contemporary Large Language Models (LLMs). While the original architecture utilized a full encoder-decoder structure, the landscape of modern LLMs has evolved into two dominant paradigms for generative tasks: the encoder-decoder architecture and the decoder-only architecture. This distinction is pivotal for understanding how models perceive input sequences and, consequently, how they exhibit sensitivity to input order [4,6,16,60–63].

Encoder-decoder models, exemplified by T5 and BART, employ a bidirectional encoder to process the input sequence, allowing the model to attend to all tokens simultaneously with full bidirectional visibility. This is followed by an autoregressive decoder for generation. While effective for sequence-to-sequence tasks such as translation, this architecture requires distinct training objectives and is less common in the current generation of general-purpose generative models [61,62].

In contrast, the decoder-only architecture has become the *de facto* standard for state-of-the-art LLMs, including the GPT series, LLaMA, and PaLM. These models process the entire sequence, concatenating instructions, examples, and incomplete responses, through a unified autoregressive stack. A defining characteristic of decoder-only models is the causal attention mechanism, which enforces a strict unidirectional flow of information so that the representation of any given token depends only on its predecessors. This structural constraint is crucial for efficient generative pre-training and also introduces an informational asymmetry across positions, the consequences of which are analyzed in Section 3 [6,16,60,63].

2.2. Causal Self-Attention and Information Asymmetry

The core computational engine of modern decoder-only LLMs is the scaled dot-product self-attention mechanism, which allows the model to dynamically weigh the importance of different tokens within a sequence. Formally, given the query Q, key K, and value V matrices derived from the input representations (or hidden states), the attention output is computed as [60]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \tag{1}$$

where d_k represents the dimension of the key vectors, and M denotes the attention mask matrix. In contrast to the bidirectional visibility characteristic of encoder components, decoder-only models utilize a causal attention mask, typically realized as a lower-triangular matrix where:

 doi:10.20944/preprints202604.2234.v1

$$M_{ij} = \begin{cases} 0 & \text{if } j \leq i \\ -\infty & \text{if } j > i \end{cases} \tag{2}$$

This causal mask strictly enforces a unidirectional flow of information, constraining the model to attend only to preceding tokens when predicting the current one. The autoregressive property is essential for coherent text generation, but it also distinguishes early and late positions in how their representations propagate through the layer stack. The behavioral consequences of this asymmetry, together with the role of positional encodings, are analyzed in Section 3 [38,48,54].

2.3. Positional Encodings and Distance Sensitivity

Self-attention alone does not encode token order, so an explicit signal is needed to inject positional information. This is achieved through positional encodings, which have evolved from absolute formulations to relative schemes [60,64–68].

Early architectures typically employed Absolute Positional Embeddings (APE), where a unique learnable or sinusoidal vector is added to each token's representation based on its absolute index. APE is effective for fixed-length contexts but often struggles with length extrapolation, failing to generalize to positions beyond those seen during training [60]. Modern LLMs increasingly adopt Relative Positional Encodings, with Rotary Positional Embeddings (RoPE) being the prominent standard. RoPE encodes position by rotating the query and key vectors in the complex plane by an angle proportional to their absolute position, which induces a distance-dependent interaction in the attention calculation (QK^T) [38,64,69]. The interaction patterns of these encodings, and how they combine with causal masking to shape the model's effective use of position, are discussed in Section 3.

2.4. Instruction Tuning and In-Context Learning

While pre-training endows LLMs with broad linguistic knowledge, their capability to follow user directives and perform specific tasks such as multiple-choice reasoning is substantially shaped through post-training adaptation. Two widely adopted paradigms are central to this process: gradient-based instruction tuning and inference-time in-context learning (ICL). Both paradigms shape how models perceive and select among options, and they are therefore central to the emergence of selection bias [5,6,10,15,17–19,70–74].

Instruction tuning, implemented via Supervised Fine-Tuning (SFT) and preference-based alignment methods such as RLHF and DPO, aligns the model with human intent by training on instruction–response data and preference feedback. This stage is critical for defining the model's behavior in choice-based tasks. The adaptation process can also introduce or amplify systematic biases. If the training data exhibits statistical skews, for example when correct answers are disproportionately labeled "A" or appear at specific positions, the model tends to internalize these patterns as shortcuts, resulting in persistent label or position biases even during inference on balanced test sets [14,70,75].

In-context learning (ICL), in contrast, adapts the model to tasks without updating parameters by prepending a few demonstration examples to the input context. ICL relies heavily on the self-attention mechanism discussed in Section 2.2, making it highly susceptible to input formatting. A well-documented phenomenon in this setting is prompt brittleness, where the model's prediction fluctuates drastically based on the permutation of few-shot examples or the order of options. This instability indicates that selection bias is not merely a training artifact but also a dynamic phenomenon emerging from the interaction between the model's architectural constraints and the immediate input context [15,17–19,49,50,52].

 doi:10.20944/preprints202604.2234.v1

3. Origins and Mechanisms of Selection Bias

3.1. Training-Data Bias: Structural Priors

The sensitivity of models to input format ultimately stems from the fact that the training data on which they depend are inherently skewed. These biases are absorbed and solidified through the two major stages of model development, pretraining and instruction tuning [14,15,56,76–78].

The first source of skew lies in the pretraining corpora themselves. Pretraining data contain highly imbalanced distributions not only over tokens but also over formatting patterns and structural templates. In many web pages, educational materials, and QA-like content, enumerations and multiple-choice templates frequently use conventional identifiers (*e.g.*, "A/B/C/D" or "1/2/3/4"), where early labels such as "A" or "1" co-occur more often with list onsets and canonical answer formats. When the model is uncertain, it may fall back on these high-frequency formatting cues, forming a label or token prior that can manifest as content-agnostic preferences in choice selection [14,21].

A second source of skew comes from structural conventions and task-specific shortcuts. Beyond token-level frequency, document conventions introduce strong positional priors, for example through repeated introduction–body–conclusion layouts and stylized list structures. During instruction tuning, such priors can be amplified or solidified if training instances are unintentionally skewed, for example when correct options correlate with particular positions, labels, or recurring templates. Models may then learn shortcut features that correlate with training-time success, yielding persistent selection biases even when correct decisions require genuine content-based reasoning [42]. In this view, apparent preferences for certain positions or symbols often reflect learned priors from the training distribution, compounded by the model's limited acquisition of permutation invariance, rather than a logically grounded deduction [3,79].

3.2. Model Structural Bias: Internal Mechanisms

The skewness in training data supplies the raw material of bias, while the internal structure of the Transformer provides the mechanism through which this material is processed into concrete biased behaviors. The architectural choices that endow the model with its capabilities also predispose it to capture and amplify positional preferences and format-sensitive heuristics [38,48,54,65,66,80].

A first source of structural bias is the causal attention mask. Introduced in Section 2.2 as a defining feature of decoder-only LLMs, this mask has direct consequences for selection bias. Because earlier tokens are visible to all subsequent positions, their representations have more downstream opportunities to influence the evolving hidden states than tokens placed near the end of the sequence [54]. From the perspective of selection bias, this asymmetry forms the architectural basis for the model's tendency to emphasize the beginning of a sequence, often referred to as the primacy bias.

A second source is the interaction between positional encodings and distance. Self-attention is permutation-invariant, so models rely on positional encodings to perceive order. As discussed in Section 2.3, modern LLMs typically use Rotary Positional Embeddings (RoPE) and other relative schemes. Empirically, many such models exhibit a reduced ability to leverage far-away tokens as context length grows, which manifests as an effective distance decay favoring nearby information, namely a recency bias. The interaction between this functional decay and the cumulative effect of the causal mask creates a competitive dynamic, often surfacing as the lost-in-the-middle phenomenon [38,81]. Training-length distributions and attention dilution also contribute, but the interaction between mask and encoding provides a useful mechanistic lens for understanding the U-shaped performance curve observed in long-context evaluation.

The downstream effects of position bias also vary by task type. Recent work [58] shows that position bias has different downstream effects depending on whether a task primarily involves retrieval or chain-of-thought reasoning, and that these differences carry implications for the design of mitigation methods. We return to this distinction, and to its naming as token-shifting versus thought-shifting, when discussing the corresponding distillation framework in Section 4.3.

 doi:10.20944/preprints202604.2234.v1

Beyond these macro-level architectural priors, bias is also instantiated within specific micro-level components. Research in mechanistic interpretability indicates that bias is not uniformly distributed but modularized. Specific attention heads and Feed-Forward Network (FFN) vectors act as biased components that consistently transmit systematic preference signals or option-identifier-correlated features regardless of context. These parameters project biased preferences directly into the vocabulary space, producing weight-encoded priors in which selection bias becomes hard-coded into the model's weights [7,27,54].

3.3. Summary: The Causal Framework of Selection Bias

Selection bias in LLMs is a full-chain phenomenon spanning external input formats, training-data priors, and architectural inductive biases. Superficially equivalent variations in input format, such as reordering or relabeling options, act as triggers that expose latent non-invariances. Systematic skews in training data provide the statistical priors that the model can fall back on under uncertainty, and the Transformer's architectural design supplies the mechanisms through which these priors are amplified and instantiated as persistent behavioral tendencies. This causal framework, linking data artifacts to mechanistic realization, provides the conceptual foundation for the evaluation protocols and mitigation strategies discussed in the subsequent sections [8,26].

4. Mitigation Strategies: A Taxonomy by Intervention Level

Selection bias in LLMs has motivated a rich variety of mitigation strategies [19,27,28,54–56,59]. These methods differ in their technical pathways, implementation costs, and applicable scenarios, but share a common objective: to disentangle model decisions from content-irrelevant confounding factors so that judgments reflect the model's reasoning rather than non-semantic surface features.

We organize existing mitigation strategies into three major categories according to their level of intervention. The first category, inference-time and post-hoc methods (Section 4.1), does not modify model parameters but instead manipulates inputs through prompt engineering or calibrates outputs through probability adjustment during inference. Methods in this category are inexpensive, easy to implement, and applicable to black-box models. The second category, architectural and internal mechanism modifications (Section 4.2), intervenes directly in the model's internal components, including attention masks, positional encodings, feed-forward networks, and specific weight parameters, to eliminate the structural basis of bias. These white-box methods can be highly effective but require full access to the model architecture. The third category, training-level debiasing methods (Section 4.3), introduces debiasing objectives during training or fine-tuning to reshape the model's learned representations. Such methods are more resource-intensive but offer deeper and more lasting debiasing with zero additional inference cost.

To illustrate how these three intervention levels map onto the different bias types discussed in Section 1.4, Table 2 provides a two-dimensional taxonomy matrix summarizing the landscape of existing methods. Section 4.4 concludes the chapter with a comparative analysis of practical trade-offs.

Table 2. Two-dimensional taxonomy of debiasing methods. Rows correspond to bias types; columns correspond to intervention levels. Methods addressing multiple bias types simultaneously appear under "Composite Selection Bias."

Bias type	Inference-time / Post-hoc	Architectural / Internal	Training-level
Label Bias	CC, DC, LOOC, ICC, SDC	—	Causal-Debias
Position Bias	BPC, MEC, PORTIA, HITLC	Set-LLM, PINE, RoToR	SOD, CPD, Pos2Distill
Composite Selection Bias	PriDe, CalibraEval, AOI, Gray/Black-box variants	UniBias, BNP	PA-GRPO, Teacher-Student Debiasing

4.1. Inference-Time and Post-Hoc Methods

These methods treat the model as a fixed-function system and achieve debiasing by carefully designing inputs or calibrating outputs. They are valued for their generality, ease of deployment, and compatibility with both open-source and proprietary (black-box) models. We organize them into

 doi:10.20944/preprints202604.2234.v1

four technical families: label calibration, positional calibration, composite distribution calibration, and prompt-level strategies [19,23,28,37].

4.1.1. Label Calibration

As defined in Section 1.4.3, label bias refers to the model's systematic prior preference for specific option identifiers, such as "A", "B", or "1", independent of the actual content. The dominant mitigation strategy is calibration: estimating the content-independent prior bias before the final prediction and then removing it from the output probabilities to yield a content-based decision [32,51].

Calibration methods have evolved from simple heuristics to more sophisticated formulations, mainly differing in how they estimate the prior bias [19,49,50,52,53]. Contextual Calibration (CC) [19] feeds the model with semantically meaningless content such as "N/A" to probe its raw preference for each label. Domain-context Calibration (DC) [49] replaces meaningless content with randomly sampled text from the target corpus, thereby capturing richer domain-specific signals. Leave-One-Out Calibration (LOOC) [50] leverages the few-shot setting itself, iteratively removing each demonstration example and observing the change in predictions to infer prior bias. In-Context Calibration (ICC) [52] extends LOOC by introducing perturbation-based prior estimation: by shuffling the token order within demonstration examples, ICC constructs calibration inputs that intentionally break spurious associations between inputs and labels. Synthetic Data Calibration (SDC) [53] addresses the scarcity of real-world domain data by using the generative capability of LLMs to synthesize diverse calibration data from a small set of demonstration examples, achieving effective calibration even in data-sparse scenarios.

These methods all target label bias by explicitly modeling and correcting the model's inherent prior over label identifiers. The progression from contextual probing to synthetic data generation represents a series of refinements in how the prior is estimated, with the shared goal of restoring an alignment between model predictions and the semantic content of the input [19,49,50,52,53].

4.1.2. Positional Calibration via Prompt Engineering

For positional bias, namely the systematic variation in preferences depending on where information appears in the input sequence, a family of inference-time methods mitigates bias by manipulating the prompt structure or aggregating outputs across multiple orderings [37,82].

Balanced Position Calibration (BPC) [35] evaluates each candidate answer across different positions and averages the results to offset the model's fixed positional preference. Multi-Evidence Calibration (MEC) [35] takes advantage of the generative capabilities of LLMs by compelling the model to generate detailed justifications before providing a final choice; this explain-then-decide process is intended to promote more deliberate and rational decision-making. PORTIA (Split and Merge) [32] simulates human strategies for comparing long texts: it splits candidate answers into fragments, aligns them by length and semantics, and then merges them into a more balanced prompt for evaluation. Human-in-the-Loop Calibration (HITLC) [35] introduces a hybrid workflow that identifies uncertain cases using instability metrics of model predictions and selectively routes them to human experts, balancing automation efficiency with assessment accuracy.

4.1.3. Composite Distribution Calibration

Beyond strategies targeting specific label or positional biases, several methods address composite selection biases that arise from the interaction of multiple bias sources. These methods treat bias as a systemic deviation that distorts the model's predicted probability distribution away from genuine content-based reasoning [23,28].

PriDe (Prior-debiased Re-estimation) [23] employs a two-stage process to explicitly separate the model's prior bias toward specific option identifiers from its predictive distribution. In the first stage, the model performs multiple predictions over a small subset of samples, each time permuting the order of options, to estimate a global prior bias for each option ID. In the second stage, the model's raw predicted probabilities are divided by the corresponding global prior values, canceling systematic preferences and producing content-driven predictions.

 doi:10.20944/preprints202604.2234.v1

CalibraEval [28] reformulates the debiasing task as an optimization problem that learns a projection function mapping biased output distributions to unbiased targets. The optimization constraint enforces invariance under permutation of option positions or identifiers. CalibraEval introduces a Non-parametric Order-preserving Algorithm (NOA), which leverages partial order relations among predicted distributions corresponding to different permutations and derives the optimal calibration function without requiring any ground-truth labels.

Gray-box and black-box variants [14] address different access levels. For gray-box models, where output probabilities are observable, calibration can be performed by averaging forward and reversed permutations to neutralize order-induced artifacts. For black-box models, where probabilities are inaccessible, a three-step double-hop process is used: identifying potentially biased options, comparing forward and reversed queries, and applying decision logic to avoid high-bias candidates.

4.1.4. Prompt-Level Strategies with Injected Uncertainty

A simple yet effective prompt-level strategy mitigates bias by allowing the model to refuse to answer. AOI (Auxiliary Option Injection) [27] is based on the observation that LLMs often produce biased or arbitrary guesses under uncertainty because they are forced to select an option even when information is insufficient. AOI introduces an auxiliary option such as "I don't know" into the list of choices, allowing an explicit exit when confidence is low. After probability computation, the auxiliary option is removed and the highest-probability original option is selected as the final answer. This mechanism redistributes probability mass, reducing bias while improving accuracy.

4.2. Architectural and Internal Mechanism Modifications

While inference-time methods avoid modifying the model, a more fundamental line of research seeks to eliminate bias at the structural level by directly intervening in the model's internal components. These white-box approaches can be grouped into two families: methods that enforce order invariance through modified positional encodings and attention masks, and methods that identify and neutralize specific biased components within the network [7,27,54,55,83–85].

4.2.1. Order-Invariant Architectural Modifications

These methods aim to achieve provable permutation invariance by modifying the Transformer's core components, namely the positional encodings and attention masks, so that the model's output remains consistent under any permutation of the input options [55,66–68,83–85].

Set-Based Prompting and Set-LLM transform the input options from an ordered list into an unordered set. They enforce identical positional encodings across all options and modify the attention mask to block inter-option communication, removing the model's ability to perceive or exploit the order of options. This redesign eliminates positional bias at its source [55,83]. PINE (Position-INvariant inferencE) [54] introduces a more refined mechanism that enables bidirectional attention among options and dynamically reassigns virtual positions based on semantic relevance, allowing the model to process options according to their content rather than their original sequence. RoToR [84] builds on PINE by incorporating query-independent ordering and routing mechanisms, improving stability and robustness when handling inputs that inherently contain ordered information.

A shared limitation of these architectural methods is the assumption that input elements are genuinely unordered. The assumption breaks down when options have dependencies or sequential meaning, for example MCQA items containing "none of the above", which implicitly references all preceding options [84].

4.2.2. Internal Component Identification and Intervention

Rather than redesigning the attention or positional encoding scheme globally, these methods take a more targeted approach: they identify specific internal components, including attention heads, feed-forward network (FFN) vectors, and projection-layer weights, that are responsible for encoding biased behavior, and then neutralize them [7,27,54,80].

 doi:10.20944/preprints202604.2234.v1

UniBias [7] integrates mechanistic interpretability into debiasing. It projects the outputs of FFNs and attention heads into the vocabulary space to decode the preference signals they transmit. Using criteria such as correlation, bias intensity, and low variance, UniBias identifies biased components and dynamically masks them during inference, reducing bias while preserving functionality. BNP (Bias Node Pruning) [27] adopts a more permanent, parameter-level approach. It first locates the layers most responsible for bias, identifies parameters with strong bias interactions, and prunes (zeros out) these weights, removing the structural basis of bias from the model. After pruning, the modified architecture incurs zero additional runtime cost.

Both approaches require complete access to the model's internal architecture, which precludes deployment on closed-source systems. Permanent parameter ablation, as in BNP, also carries a risk of degrading the model's broader generalization capabilities, and therefore needs careful validation on held-out tasks [27,54].

4.3. Training-Level Debiasing Methods

The methods discussed above operate at inference time or modify internal components without retraining. They share a common limitation: the debiasing effect must be applied anew for each query. A complementary line of research seeks to eliminate selection bias during the model's training or fine-tuning stage, reshaping the model's learned representations so that its predictions become inherently invariant to irrelevant input-format factors. Although more resource-intensive, training-level methods offer zero additional inference cost once deployed [56–59,86].

4.3.1. Causal Invariant Learning

A principled approach to training-level debiasing draws on the framework of causal invariance. The core insight is that bias-related features such as option position or label identity are non-causal with respect to the correct answer, whereas content-related features are causal. By encouraging the model to rely only on causal features during fine-tuning, spurious correlations with positional or label cues can be suppressed [56,87].

Causal-Debias [56] instantiates this idea through invariant risk minimization during fine-tuning. It first identifies bias-relevant and label-relevant factors for a given downstream task using causal analysis. Interventions are then performed on non-causal factors across different demographic or format groups, and an invariant risk minimization loss penalizes reliance on these non-causal features. Experimental results on multiple downstream tasks indicate that this approach substantially reduces stereotypical associations while maintaining competitive task performance.

CPD (Causal Perception long-term Dialogue) [47] applies causal invariant principles to position bias in long-term dialogue systems. It uses a perturbation-based causal variable discovery method to identify causally relevant utterances from dialogue history, distinguishing them from utterances that are merely positionally salient. The model is then fine-tuned with a debiasing objective that penalizes spurious correlations between response generation and the position distribution of relevant utterances. The approach addresses the myopia problem in which dialogue models over-attend to recent turns while neglecting earlier, causally important exchanges.

4.3.2. Self-Supervised Debiasing

An alternative training-level paradigm avoids the need for external bias annotations by using the model's own outputs as supervision signals for debiasing [57,87].

SOD (Self-Supervised Position Debiasing) [57] mitigates position bias during fine-tuning without requiring external bias knowledge or annotated unbiased samples. SOD operates in three stages. First, a low-bias inference module collects multiple unsupervised responses from pre-trained LLMs using diverse prompting strategies designed to minimize position bias. Second, an Objective Alignment Module (OAM) prunes low-quality responses to prevent them from undermining model comprehension during debiasing. Third, a multi-objective optimization module jointly optimizes the target task loss and a debiasing loss derived from the pruned unsupervised responses. Experiments across eight

 doi:10.20944/preprints202604.2234.v1

datasets and five NLP tasks show that SOD consistently outperforms existing methods in mitigating three types of position bias while sacrificing only minimal performance on biased samples.

4.3.3. Knowledge Distillation for Debiasing

A recent direction combines permutation-based debiasing with knowledge distillation to achieve both thoroughness and efficiency. The motivating observation is that full permutation ensembling produces highly debiased outputs but its factorial cost makes it impractical for deployment [58,86].

Teacher-Student Debiasing [86] addresses this by training a compact student model to emulate the debiased outputs of a computationally expensive teacher. The teacher generates debiased probability distributions by aggregating predictions across all permutations of input options. The student is then trained via KL-divergence minimization to reproduce these debiased distributions from a single forward pass, regardless of option ordering. Two student variants are explored: a pure distillation student that directly learns the debiased distribution, and an error-correction student that takes a single biased teacher prediction and learns to correct it. Experiments demonstrate that small student models (330M parameters) can outperform their larger biased teacher counterparts while maintaining permutation invariance, offering a practical path to deploying debiased models at scale.

Pos2Distill (Position-to-Position Knowledge Distillation) [58] targets position bias in long-context scenarios through inter-position knowledge distillation. The motivating observation is that position bias itself reveals supervisory signals: the model performs well at advantageous positions such as context boundaries and poorly at disadvantaged ones such as the middle of the context. Pos2Distill leverages this disparity by distilling knowledge from advantageous positions to rectify predictions at unfavorable ones. The work also distinguishes how position bias manifests across task types, separating retrieval and reasoning regimes. In retrieval tasks, position bias predominantly causes token-shifting, where the model generates incorrect or shifted tokens when key information lies at a disadvantaged position. In reasoning tasks that rely on Chain-of-Thought, position bias interacts with the reasoning trajectory to cause thought-shifting, where the entire line of reasoning deviates due to positional disadvantage. This distinction motivates two specialized instantiations: Pos2Distill-R1, which mitigates token-shifting through a KL-divergence loss combined with an anchoring mechanism for advantageous positions, and Pos2Distill-R2, which reshapes reasoning trajectories by transferring high-quality reasoning patterns from privileged to suboptimal positions [58].

4.3.4. Reinforcement Learning with Permutation-Aware Optimization

A different training-level paradigm uses reinforcement learning (RL) to internalize permutation invariance as a learned behavior rather than imposing it through external calibration or architectural constraints [59,72–74].

PA-GRPO (Permutation-Aware Group Relative Policy Optimization) [59] addresses a limitation of pointwise training methods, namely that they ignore the constraint that semantically equivalent permutations of the same question should yield consistent answers. PA-GRPO constructs a permutation group for each training instance by generating multiple candidate permutations of the option set, and optimizes the model using two complementary mechanisms. A cross-permutation advantage computes the advantage of each response relative to the mean reward over all permutations of the same instance, and a consistency-aware reward explicitly encourages the model to produce identical decisions across different permutations, penalizing inconsistency even when individual predictions are correct. Experimental results on benchmarks spanning MCQA (MMLU, GPQA, ARC-Challenge) and LLM-as-a-Judge evaluation (MT-Bench, JudgeBench) indicate that permutation-aware RL optimization can both improve accuracy and reduce decision instability under reordering, offering a training-time alternative that avoids the factorial cost of inference-time permutation ensembling [59].

4.4. Discussion: Practical Trade-Offs and Inference Costs

Having surveyed the spectrum of debiasing strategies, we now synthesize their practical trade-offs. Table 3 provides a comparative summary.

 doi:10.20944/preprints202604.2234.v1

Table 3. Practical trade-offs of debiasing methods for selection bias.

Method type	Extra inference cost	Model access	Typical stability
Prompt engineering (*e.g.*, AOI)	≈ 0	Black-box	Medium / task-dependent
Post-hoc calibration (*e.g.*, CC)	$+N$ forward passes	Black-box	High if priors stable
Permutation ensembling (*e.g.*, PriDe)	$O(K!)$ forward passes	Black-box	Very high, impractical online
Internal interventions (*e.g.*, UniBias)	$O(1)$ or $O(N)$	White-box required	High, but limited to open models
Architectural modifications (*e.g.*, PINE)	≈ 0	White-box required	High, order-assumption needed
Causal invariant training (*e.g.*, Causal-Debias)	≈ 0 at inference	White-box required	High, requires causal annotation
Self-supervised debiasing (*e.g.*, SOD)	≈ 0 at inference	White-box required	High, depends on response quality
Position KD (*e.g.*, Pos2Distill)	≈ 0 at inference	White-box required	High, task-type specific
Permutation-aware RL (*e.g.*, PA-GRPO)	≈ 0 at inference	White-box required	Very high, costly to train

Prompt engineering: minimal cost, variable returns.

Prompt-level strategies such as AOI [27] are tuning-free and add minimal computational overhead, typically requiring only a single forward pass with modified input context. Their black-box compatibility allows deployment on proprietary APIs. As shown by Choi *et al.* [27], the efficacy of AOI relies on the model's internal calibration: the mechanism often underperforms when the model exhibits high confidence in a biased prediction. MEC [35] improves deliberation by prompting the model to generate justifications before deciding, but introduces additional generation latency. Prompt-based methods serve as a practical first line of defense rather than as a sufficient remedy for deeply ingrained biases.

Post-hoc calibration: effective but prior-dependent.

Calibration methods such as CC [19], DC [49], LOOC [50], and SDC [53] operate directly on output probabilities without altering model parameters, and remain compatible with both open-source and API-accessible architectures. Each calibration query, however, needs supplementary forward passes: CC and DC impose modest overhead, while LOOC scales linearly with the number of few-shot demonstrations [50]. A central concern is stability. If the bias distribution shifts across domains or tasks, a prior derived from a single dataset transfers poorly and must be re-estimated. SDC mitigates data scarcity through synthetic examples [53], but introduces a dependency on the generative quality of the source model, raising the risk of circularity if the generator harbors the same biases targeted for calibration. As Li *et al.* [28] observe, CC and DC were primarily designed for ICL label bias and may fail to capture the more complex selection bias that arises in LLM-as-a-Judge frameworks.

Permutation-based methods: thoroughness versus scalability.

Permutation-based methods provide rigorous debiasing by evaluating instances across diverse orderings and aggregating outcomes. Full permutation ensembling consistently yields high stability [18,22,23], but its computational overhead scales factorially as $\mathcal{O}(K!)$, which is prohibitive for large-scale deployment. PriDe [23] addresses this by estimating prior bias on a marginal subset (approximately 5% of test samples) and applying it to debias the remainder, with strong interpretability and cross-domain transferability. CalibraEval [28] learns a non-parametric calibration function directly from permuted observations without ground-truth annotations, offering a scalable alternative for evaluation environments.

Internal mechanism interventions: surgical but access-restricted.

UniBias [7] and BNP [27] provide targeted debiasing by isolating and neutralizing specific biased components. UniBias dynamically masks biased attention heads and FFN vectors during inference, introducing marginal per-layer overhead without permanent parameter modifications. BNP permanently prunes biased weights in the final projection layer and incurs zero additional runtime cost after pruning [27]. Both methods require full architectural access, which precludes deployment on closed-source systems. BNP additionally risks degrading generalization, and therefore needs validation on held-out tasks.

 doi:10.20944/preprints202604.2234.v1

Architectural modifications: principled but invasive.

Methods that enforce order invariance, including Set-Based Prompting [55], Set-LLM [83], PINE [54], and RoToR [84], represent the most principled approach to eliminating positional bias. Set-Based Prompting provides mathematical guarantees of invariance by assigning identical positional encodings and blocking inter-option attention [55]. PINE enables bidirectional attention with dynamic position reassignment [54], and RoToR further addresses training-inference distribution mismatch [84]. These methods assume that input elements are genuinely unordered, an assumption that breaks down when options carry sequential or interdependent meaning. Blocking inter-option attention can also degrade output quality when comparative reasoning is required [55]. As shown in Table 2, the architectural column is empty for label bias, reflecting the fact that modifying positional encodings or attention masks does not address content-independent label priors; label bias remains primarily a target for calibration and training-level methods.

Training-level methods: deep but resource-intensive.

Training-level methods such as PA-GRPO [59] and Pos2Distill [58] offer zero additional inference cost once trained, which makes them attractive for deployment. They differ in the supervisory signal used during training: permutation-aware reinforcement learning shapes consistency across reorderings, while inter-position distillation transfers knowledge between advantageous and disadvantaged positions. Their shared advantage over inference-time approaches is permanence: the debiasing effect persists across downstream applications without per-query calibration. The cost is that training-level methods require white-box access, dedicated data construction, and substantial compute, which restricts their applicability for closed-source deployments.

Cross-cutting considerations.

Several overarching factors are worth noting. The choice of debiasing strategy must be informed by deployment context: real-time systems demand low-latency methods such as prompt engineering or pre-pruned architectures, while offline evaluation pipelines can absorb higher computational costs [28,35]. Many techniques are complementary rather than mutually exclusive. For example, combining BNP with AOI yields cumulative gains, with pruning handling white-box debiasing and prompting extending coverage to black-box environments [27]. No single method generalizes across all bias types: calibration methods excel at mitigating label bias [19,49,50] but often fail to resolve positional bias, while architectural interventions enforce positional invariance and leave token-level preferences unaddressed [54,55]. A layered, multi-strategy framework therefore appears necessary for comprehensive debiasing. Most existing studies evaluate on mid-sized architectures (7B–13B parameters) [7,27]; whether findings transfer to significantly larger models remains an open empirical question.

Having surveyed the technical landscape of mitigation strategies, we now turn to how selection bias is measured and where it manifests in practice. This section reviews evaluation metrics, commonly used benchmarks, and the impact of bias across key application domains.

5. Empirical Study: Metrics, Benchmarks, and Application Domains

5.1. Evaluation Metrics for Selection Bias

Selection bias manifests when the model output changes systematically under semantically equivalent format perturbations, such as option permutation, label/token renaming, or swapping candidate positions in pairwise judging [14,23,35]. Consequently, evaluation metrics are organized around two core axes: stability under equivalence-preserving perturbations, and the strength of position or label priors [18,22,28,30].

 doi:10.20944/preprints202604.2234.v1

Setup and Notation.

For each instance i, let the candidate set be $\mathcal{A}_i = \{a_{i1}, \ldots, a_{iK}\}$ and the perturbation set be Π (*e.g.*, permutations, symbol renaming) [14,18,20,23]. Let $\pi \in \Pi$ denote a perturbation and $p_\theta(\cdot \mid x_i, \pi)$ the model distribution. The predicted choice under π is

$$\hat{y}_i^{(\pi)} = \arg\max_{k \in [K]} p_\theta(a_{ik} \mid x_i, \pi). \tag{3}$$

5.1.1. Stability Under Equivalent Perturbations

Sensitivity Gap (Worst-case Swing).

A widely used metric is the max–min performance swing over perturbations, termed the sensitivity gap [14,18,22,23]:

$$\Delta S = \max_{\pi \in \Pi} S(\pi) - \min_{\pi \in \Pi} S(\pi), \tag{4}$$

where $S(\pi)$ represents a metric such as accuracy, F1 score, or win-rate [18,23,29,35].

Mean–Variance Stability.

Beyond worst-case swings, average stability is measured as [18,22]:

$$\mu_S = \mathbb{E}_{\pi \sim \Pi}\big[S(\pi)\big], \tag{5}$$

$$\sigma_S = \sqrt{\mathbb{E}_{\pi \sim \Pi}\big[(S(\pi) - \mu_S)^2\big]}. \tag{6}$$

In practice, Π is approximated by sampled perturbations [18,22,23].

Flip and Conflict Rates.

Instance-level consistency quantifies how often predictions change across perturbations [14,23]:

$$\text{FlipRate} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big[\exists \pi, \pi' \in \Pi : \hat{y}_i^{(\pi)} \neq \hat{y}_i^{(\pi')}\Big]. \tag{7}$$

For pairwise judging, this manifests as the conflict rate under A/B swapping [28,30,32,35]:

$$\text{Conflict} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big[J_i \not\equiv \text{swap}(J_i')\Big], \tag{8}$$

where J_i is the judgment on (A, B), J_i' on (B, A), and $\text{swap}(\cdot)$ maps outcomes to the inverse order [30,32,35].

Probability-level Instability.

When access to logits is available, the variance of the gold option $y_i^\star$ provides a granular stability measure [14,19,23]:

$$\text{VarProb} = \frac{1}{N} \sum_{i=1}^{N} \text{Var}_{\pi \in \Pi}\Big(p_\theta(y_i^\star \mid x_i, \pi)\Big). \tag{9}$$

Choice-Distribution Shift (cKLD).

Bias strength can be quantified via the divergence of the empirical choice distribution $q_i(k)$ from the uniform distribution U [50,55,83]:

$$\text{cKLD} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q_i(k) \log \frac{q_i(k)}{1/K}, \tag{10}$$

where $q_i(k) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \mathbb{I}[\hat{y}_i^{(\pi)} = k]$.

 doi:10.20944/preprints202604.2234.v1

5.1.2. Strength of Label and Position Priors

Content-Free Label Prior

Standard practice estimates a content-independent prior $p_0(l)$ using neutral inputs to quantify deviation from uniformity [19,49]:

$$\mathrm{TV}(p_0, U) = \frac{1}{2}\sum_{l=1}^{K}\left|p_0(l) - \frac{1}{K}\right|, \tag{11}$$

$$\mathrm{KL}(p_0 \| U) = \sum_{l=1}^{K} p_0(l)\log\frac{p_0(l)}{1/K}. \tag{12}$$

Position Advantage

In pairwise settings, position advantage measures the shift in winning probability based on order [28,30,32,35]:

$$\mathrm{PosAdv} = |\Pr(A \succ B \mid A \text{ first}) - \Pr(A \succ B \mid A \text{ second})|. \tag{13}$$

5.1.3. Reliability and Fairness in LLM-as-a-Judge

Inter-Rater Agreement

Agreement is typically measured using Cohen's κ (two raters) or Fleiss' κ (multiple raters) [28]. Cohen's κ is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{14}$$

where p_o is observed agreement and p_e is chance agreement.

Intraclass Correlation Coefficient (ICC)

For continuous scores, the one-way random-effects ICC assesses reliability across k raters [28]:

$$\mathrm{ICC} = \frac{\mathrm{MS}_B - \mathrm{MS}_W}{\mathrm{MS}_B + (k-1)\mathrm{MS}_W}, \tag{15}$$

where MS_B and MS_W denote between-item and within-item mean squares.

Judge Repetition Stability

Self-consistency is measured by repeatedly querying the model for the same instance [30]:

$$\mathrm{RepStab} = \frac{1}{N}\sum_{i=1}^{N}\frac{\max_o c_{io}}{R}, \tag{16}$$

where c_{io} is the count of outcome o over R repetitions.

Position Consistency (Swap Agnostic)

A core diagnostic for position bias is consistency under candidate swapping [30,32,35]. Let $J_i = J(A_i, B_i)$ and $J_i' = J(B_i, A_i)$. The swap-normalized consistency is:

$$\mathrm{PosCons} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[J_i = \mathrm{swap}(J_i')\right], \tag{17}$$

where $\mathrm{swap}(\cdot)$ maps $A \succ B \leftrightarrow B \succ A$. The corresponding inconsistency rate is $\mathrm{Conflict} = 1 - \mathrm{PosCons}$.

Preference Fairness (Symmetry)

Even if judgements are consistent, a judge may exhibit aggregate position preference. Let W_i and W_i' be indicators that the first-position candidate wins under the original and swapped orders,

 doi:10.20944/preprints202604.2234.v1

respectively. The first-position win rate is $\text{WinRate}_{\text{first}} = \frac{1}{2N}\sum_{i=1}^{N}(W_i + W_i')$. Preference fairness is defined as deviations from 0.5 [30,32,35]:

$$\text{PrefFair} = 1 - 2\left|\text{WinRate}_{\text{first}} - \frac{1}{2}\right|. \tag{18}$$

5.1.4. Context-Specific and Robustness Metrics

Outcome Disparity (RStd)

To capture imbalances induced by priors, label-wise recall dispersion is computed as [28,50]:

$$\text{RStd} = \sqrt{\frac{1}{K}\sum_{l=1}^{K}(r_l - \bar{r})^2}, \tag{19}$$

where r_l is the recall for label l, and $\bar{r} = \frac{1}{K}\sum_{l=1}^{K} r_l$ is the mean recall across all labels.

External Validity

Rank correlations (Kendall's τ, Spearman's ρ) are standard for validating judge-based rankings against human references [29,30].

Long-Context Position Bias

The lost-in-the-middle effect is quantified by the gap between edge and middle performance [39,84,85]. Let $A(t)$ denote the model performance (*e.g.*, accuracy or exact match) when the relevant evidence is placed at position bucket t, let T be the total number of evaluated position buckets, and let $\mathcal{M}$ denote the set of middle buckets. Then:

$$\text{LITMGap} = \frac{A(1) + A(T)}{2} - \min_{t \in \mathcal{M}} A(t). \tag{20}$$

A larger LITMGap indicates a stronger degradation in performance when relevant evidence appears in the middle of the context.

Mechanistic Bias Signals

Bias can be localized to components via logit differences (Δ_{logit}) between preferred and non-preferred options, often analyzed under ablation [7,20,27,54].

5.2. Common Benchmarks

Selection-bias evaluation typically relies on benchmarks whose inputs admit equivalence-preserving format perturbations such as option permutation, label renaming, swapping candidate order, or relocating evidence within a long context. These perturbations ensure that any systematic output change can be attributed to position or label priors rather than semantic understanding. We discuss these benchmarks by task family, detailing their utility in diagnosing and mitigating bias. Table 4 summarizes the benchmark datasets discussed in this section, organized into five groups: MCQA, ICL classification, long-context evaluation, LLM-as-a-Judge, and dialogue or multimodal settings [14,22,28,33,38,88–90].

 doi:10.20944/preprints202604.2234.v1

Table 4. Main benchmark datasets explicitly discussed in Section 5.2.

Dataset / Benchmark	Task type	Input format	Notes
MMLU	MCQA	Multiple-choice options	Broad-domain benchmark; standard for option permutation and label renaming
MMLU-Redux	MCQA	Multiple-choice options	Curated subset for reducing ambiguity and label noise
ARC-Challenge	MCQA	Multiple-choice options	Robustness against shallow cues
CommonsenseQA (CSQA)	MCQA	Multiple-choice options	Semantically related distractors
HellaSwag	MCQA	Multiple-choice options	Adversarially filtered distractors
WinoGrande	MCQA	Multiple-choice options	Reduced annotation artifacts
RACE	MCQA / reading comprehension	Passage + options	Formatting and option-symbol effects
RACE++	MCQA / reading comprehension	Passage + options	Augmented permutation evaluation
ReClor	Logical reasoning MCQA	Multiple-choice options	Shortcut bias *vs.* real reasoning
BIG-bench (Logical Deduction)	Logical reasoning MCQA	Structured answer options	Controlled candidate swapping
SST-2 / SST-5	ICL classification	Label words / demonstrations	Verbalizer bias and demo-order sensitivity
GLUE / SuperGLUE (MNLI, RTE, COPA)	ICL classification	Label words or label letters	Calibration and verbalizer sensitivity
HotpotQA (adapted)	Long-context QA	Long context with relocated evidence	Evidence-position sensitivity
RULER	Long-context evaluation	Needle-in-a-haystack style input	Clean position-bias curves
LongBench	Long-context evaluation	Diverse long-context inputs	Cross-task long-context generalization
MT-Bench	LLM-as-a-Judge	Pairwise evaluation prompts	Swap consistency diagnosis
Chatbot Arena	LLM-as-a-Judge	Pairwise comparative evaluation	Pairwise judging with verbosity confound
RewardBench	LLM-as-a-Judge	Preference / reward pairs	Verifiable preference direction
PreferenceBench	LLM-as-a-Judge	Pairwise preference evaluation	Human-preference consistency
Vicuna Benchmark	Multi-turn evaluation	Long-form dialogue comparison	Length and order interaction
SummEval	Summarization evaluation	Comparative summary assessment	Summary-order sensitivity
ESConv	Dialogue / interactive systems	Multi-turn conversation history	Recency effects in supportiveness
MSC2	Dialogue / interactive systems	Multi-turn conversation history	Persona consistency under recency bias
ImageNet variants	Vision-language classification	Class-name prompts / verbalizers	Label bias in multimodal zero-shot setting
Few-shot CLIP suites	Vision-language classification	Class-name prompts + few-shot context	Verbalizer choice effects

5.2.1. Multiple-Choice Question Answering (MCQA)

MCQA benchmarks form the primary substrate for selection bias research because the same question can be re-rendered easily by permuting option order or renaming labels (*e.g.*, A/B/C/D) while keeping the content fixed [14,22,23].

For broad, multi-domain evaluation, MMLU is the standard choice, covering humanities, social sciences, and STEM. Its wide subject coverage allows researchers to measure whether bias is consistent across domains or amplifies in specific subjects. Standard protocols generate multiple equivalent renderings per item to report invariance metrics such as flip rate. To ensure that observed biases are not artifacts of dataset noise, MMLU-Redux is frequently employed; this curated subset corrects ambiguous questions and wrong labels, so that if a mitigation method reduces permutation sensitivity on MMLU but not on Redux, the bias may be conflated with label errors. The ARC benchmark, specifically the Challenge split, is used to assess robustness against shallow cues, since it was constructed to baffle retrieval baselines. In bias studies, ARC is typically subjected to option order permutations, answer-letter renaming, and formatting changes such as line breaks to isolate non-semantic priors [78,91–94].

Commonsense tasks are particularly useful for studying option-order sensitivity because distractors are often semantically related, making models vulnerable to priors when uncertain. CommonsenseQA (CSQA) is a canonical example where distractors are derived from a structured knowledge base, requiring models to discriminate among concepts without relying on positional cues. HellaSwag adds rigor through its adversarial filtering construction; if a model exhibits strong position or label bias even with such distractors, the effect is likely due to decoding priors rather than dataset artifacts. WinoGrande complements this by focusing on pronoun and coreference resolution with systematic

 doi:10.20944/preprints202604.2234.v1

bias reduction, and is commonly used to diagnose whether choose-first or choose-last tendencies persist after explicit annotation artifacts have been minimized [95–97].

Several benchmarks probe specialized reasoning types where priors can dominate. OpenBookQA pairs questions with a core fact, testing whether reliance on answer-position priors increases when multi-hop reasoning is required. PIQA (physical commonsense) and SocialIQA (social interaction) are frequently used in permutation-invariant prompting because their candidate solutions are often comparably plausible, and this ambiguity amplifies the role of positional priors. CosmosQA extends the same idea to questions requiring implicit causal or intent inference beyond explicit text spans [98–101].

For longer contexts, RACE is widely used to study formatting-induced artifacts. Its passages and options are verbose, allowing researchers to examine the impact of choice labeling styles and option-symbol changes in addition to pure permutation. Permutation-debiasing studies also evaluate augmented variants of RACE, but the exact construction should always be documented explicitly. Logical reasoning benchmarks such as ReClor and BIG-bench Logical Deduction are critical for distinguishing superficial bias from true reasoning failures. ReClor explicitly separates bias-exploitable items from harder ones, while the structured reasoning tasks in BIG-bench allow controlled candidate swapping without semantic drift, serving as a rigorous test for order sensitivity [102–104].

5.2.2. In-Context Learning (ICL) Classification

In few-shot ICL, the answer is often a label token, making evaluations sensitive to token priors and demonstration order. SST (SST-2/SST-5) is a standard testbed in which bias protocols swap label words such as positive and negative, or permute demonstrations to quantify recency effects. NLU suites such as GLUE and SuperGLUE, including MNLI, RTE, and COPA, support multiple equivalent output heads (label words versus letters) and are essential for evaluating calibration and verbalizer sensitivity, ensuring that performance is not driven by a model's prior preference for specific tokens such as "Yes" or "No" [24,25,105–107].

5.2.3. Long-Context Position Bias

Position bias also manifests as performance degradation when evidence is moved within a long context, namely the lost-in-the-middle phenomenon. HotpotQA, adapted for long contexts, is used to test this by placing supporting facts at different positions, revealing whether models disproportionately weight early or late tokens. Synthetic benchmarks such as RULER provide a cleaner signal by extending needle-in-a-haystack probing, allowing the target item to be moved without changing semantics and generating precise position-bias curves. LongBench offers a unified evaluation across diverse tasks (QA, summarization, code), enabling measurement of whether bias mitigation techniques generalize across task types and context lengths [108–115].

5.2.4. LLM-as-a-Judge and Preference Evaluation

In pairwise judging, selection bias appears as decision inconsistency when candidate order is swapped. MT-Bench and Chatbot Arena are widely used for these diagnostics, though they also expose verbosity bias. RewardBench provides a more controlled setting for reward models, in which each pair has a verifiable preference direction, making it suitable for checking whether calibration reduces order effects. PreferenceBench is used similarly, to test whether order-invariant methods preserve correlation with human preferences [29,33,88–90,116–122].

5.2.5. Dialogue, Summarization, and Multimodal Settings

Selection bias also surfaces in benchmarks beyond MCQA, ICL, and pairwise judging. SummEval studies summarization evaluation, where bias manifests as summary-order sensitivity in comparative scoring. Multi-turn dialogue benchmarks such as ESConv and MSC2 expose recency effects in supportive response generation and persona consistency, where models tend to over-attend to the most recent turn while underutilizing earlier but causally relevant context. The Vicuna Benchmark surfaces interactions between length and order in long-form dialogue evaluation. In multimodal

 doi:10.20944/preprints202604.2234.v1

settings, the bias landscape carries over from text: ImageNet variants and few-shot CLIP suites probe label bias in zero-shot vision-language classification, where the choice of class-name verbalizer or prompt template can shift predictions even when the underlying visual content is fixed. We note that the benchmarks summarized in Table 4 are predominantly English-centric, an observation we return to when discussing the limitations of the current evaluation ecosystem in Section 6.

5.3. Bias Impact and Mitigation in Key Application Domains

Selection bias has practical consequences across the domains in which LLMs are deployed [1,29,30,38,123]. Examining these domains gives a clearer picture of the harms caused by bias and of the practical value of the corresponding mitigation techniques. We begin with multiple-choice question answering, the core experimental paradigm, and then examine the manifestations and mitigation of bias in other application areas.

5.3.1. Multiple-Choice Question Answering

Multiple-choice question answering is the standard evaluation paradigm for the knowledge, reasoning, and language comprehension abilities of LLMs, and it is also the domain where selection bias manifests most prominently. Empirical studies have repeatedly shown that the accuracy and reliability of LLMs in MCQA tasks are disrupted by multiple forms of bias [1,20,57,76,77,123–125]. Positional bias leads models to exhibit irrational preferences for options located in particular positions such as the first or last. Token bias and option-symbol bias produce unjustified preferences for particular option identifiers, for example when comparing "A/B/C/D" against "i/ii/iii/iv". In some cases, simply changing the option symbols can cause an accuracy difference of approximately 10%. These biases indicate that LLMs often base their selections on superficial, content-irrelevant heuristics rather than a genuine understanding of the question content. The resulting distortion compromises the reliability of MCQA benchmarks as measures of model capability, and poses a fundamental challenge to deployment in high-stakes contexts such as standardized testing.

To address these issues within MCQA, the research community has implemented a diverse set of mitigation strategies. Probability distribution calibration methods such as PriDe and related calibration baselines adjust output probabilities to remove prior biases [19,26]. Prompt-level strategies such as Auxiliary Option Injection introduce uncertainty-handling mechanisms, while internal component interventions such as Bias Node Pruning and UniBias modify the model components responsible for bias [7,27]. More fundamental architectural approaches, including Set-Based Prompting, enforce permutation invariance within the model structure to eliminate order dependence at its root [55,83,85]. At the training level, complementary fine-tuning approaches internalize order invariance into model parameters, avoiding inference-time overhead at the cost of additional training resources.

5.3.2. LLM-as-a-Judge

With the advancement of LLM capabilities, the LLM-as-a-Judge paradigm has emerged as an approach for scalable and low-cost automated evaluation, intended to replace or supplement time-consuming human assessments. Studies have shown that the reliability of this paradigm is undermined by selection bias. Positional bias is particularly prominent and affects even state-of-the-art models. When tasked with pairwise comparisons between candidate responses, merely swapping their presentation order can lead to inconsistent or completely reversed judgments. The resulting inconsistency compromises the fairness, reliability, and validity of evaluation outcomes, casting doubt on model ranking systems and performance comparisons derived from such judgments [29,30,33,35,89,90,118–122].

To enhance the reliability of LLM-based evaluators, researchers have developed targeted mitigation techniques primarily at the levels of post-processing and prompt engineering. Balanced Position Calibration compensates for positional preference, while Multi-Evidence Calibration encourages more evidence-grounded decision-making [28,32,37]. Input reconstruction systems such as PORTIA simulate human reading strategies to create balanced prompt structures. Optimization-based approaches such as CalibraEval frame debiasing as a projection problem and learn a calibration function that restores distributional fairness [28,32]. Complementary training-level approaches enforce evaluator consistency

 doi:10.20944/preprints202604.2234.v1

directly through fine-tuning rather than through per-query calibration. Related evaluator-side biases have also been documented in verifier and error-detection settings for mathematical reasoning [78,126–128].

5.3.3. Long-Context and RAG

In tasks involving long or multi-document inputs, positional bias commonly manifests as the lost-in-the-middle phenomenon. LLMs display a U-shaped pattern in information utilization, capturing content from the beginning and end of long contexts effectively but frequently overlooking information in the middle. The pattern undermines the performance of Retrieval-Augmented Generation systems: if the retriever places key documents in the middle of the prompt, the generator often ignores them, resulting in incorrect or incomplete outputs. Similar patterns have been observed in long-form summarization, where factual consistency across the summary exhibits the same U-shaped curve, with the lowest fidelity for information located in the middle of the source text [38,111–113,129,130].

Solutions to the lost-in-the-middle problem have been proposed at multiple levels. Some rely on prompting interventions such as explicitly guiding model attention through precise document indexing, while others target the internal mechanisms of the model. Representative works include plug-and-play or internal positional interventions that reweight attention over long contexts and dynamically redistribute positional representations to enforce position invariance [69,130]. Set Encoding [85] redefines positional encoding schemes to eliminate sequential dependencies. At the training level, Pos2Distill [58] offers a knowledge distillation approach that transfers capabilities from advantageous positions to disadvantaged ones, with specialized variants for retrieval (Pos2Distill-R1, addressing token-shifting via a KL-divergence loss) and reasoning (Pos2Distill-R2, reshaping reasoning trajectories). The approach is data-efficient and reduces the U-shaped performance gap characteristic of the lost-in-the-middle phenomenon.

5.3.4. Additional Modalities and Domains

Selection bias also affects interactive LLM systems and extends beyond text-only settings. In long-horizon dialogue and agentic settings where the model must integrate information across multiple turns, position bias often manifests as a strong recency preference: the model overweights the most recent user turns while underutilizing earlier but causally relevant context. The resulting responses can be shallow, repetitive, or contextually inconsistent, even when the missing information has already appeared in the conversation history. In multi-step agentic workflows, similar effects arise when earlier instructions, retrieved evidence, or intermediate plans are deprioritized relative to later prompt segments, causing unstable decision-making and reduced task reliability [123,131–138].

Mitigation in interactive settings focuses on improving the model's ability to preserve and re-use long-range contextual dependencies. Existing approaches include prompt and memory reorganization strategies that foreground previously established goals or constraints, retrieval-based mechanisms that re-insert relevant earlier turns into the current decision context, and causality-aware fine-tuning methods that strengthen sensitivity to long-distance dependencies [47,135,137,138]. These efforts aim to reduce the model's reliance on superficial recency cues and to encourage decisions that remain invariant to non-semantic changes in the placement of interaction history.

Selection bias also surfaces in multimodal and cross-lingual settings. In vision-language classification, ImageNet variants and few-shot CLIP suites probe label bias in zero-shot and few-shot regimes, where the choice of class-name verbalizer or prompt template can shift predictions even when the underlying visual content is fixed. Recent evidence from video-language models suggests further patterns of bias in multimodal reasoning that remain underexplored [26,117]. These observations motivate the broader extension of selection-bias research beyond text, a direction we discuss in Section 6.2.

The preceding analysis indicates that, while substantial progress has been made in both understanding and mitigating selection bias, several fundamental challenges remain unresolved. We synthesize these challenges and outline future research directions in the next section.

 doi:10.20944/preprints202604.2234.v1

6. Challenges and Future Directions

Following the systematic review of the causes, mitigation methods, evaluation systems, and key application domains of selection bias in large language models, the area has seen meaningful progress [14,28,30,38,54,56]. Selection bias is, however, deeply intertwined with model architectures, data characteristics, and task structures, and several limitations in the existing literature remain. This section synthesizes the key challenges confronting the field and outlines future research directions.

6.1. Core Challenges and Research Limitations

Despite the growing diversity of mitigation strategies, current studies face several overarching challenges, including limited theoretical depth, incomplete or domain-specific solutions, and an underdeveloped evaluation system.

The first challenge concerns the theoretical understanding of bias formation. Most existing works, including many effective mitigation methods, address observed manifestations of bias without fully uncovering the mechanisms by which data properties, training objectives, and architectural components interact to produce systemic bias. Causal attention and positional encoding have been identified as key sources of position bias, but the interactions between other Transformer components such as residual connections and feed-forward networks and the attention mechanism remain poorly understood. Foundational questions, including which data characteristics induce selection bias and why different model families exhibit distinct bias patterns, remain largely unanswered [7,38,48,54,80,87].

The second challenge concerns the limitations of current mitigation approaches. Many strategies are narrowly applicable to specific contexts. Calibration-based methods such as CalibraEval are primarily designed for pairwise comparison tasks and are difficult to extend to more complex list-ranking scenarios. Gray-box approaches depend on access to probability outputs from model APIs, which are often unavailable in proprietary systems. Set encoding methods enforce permutation invariance with theoretical rigor but require direct modification of model internals, which makes them inapplicable to closed-source models. These methods also struggle with tasks involving interdependent options, for example items containing "A and C are both correct". Many methods rely on external resources: causal-debiasing techniques depend on manually curated bias lexicons that can never achieve full coverage, and self-supervised methods such as SOD are constrained by the quality of their unbiased pretraining models [28,55–57,83].

The third challenge concerns the evaluation ecosystem. Existing evaluation practices for selection bias remain fragmented and task-specific. Current studies rely on heterogeneous benchmark families, including MCQA datasets with option permutation, pairwise judging benchmarks with candidate-order swapping, and long-context benchmarks with evidence relocation, but the field still lacks a unified evaluation protocol that enables direct comparison across tasks and methods. Many benchmarks overemphasize pairwise or single-answer settings while underrepresenting more realistic scenarios such as listwise ranking, multiple-correct-answer reasoning, and retrieval-augmented generation with conflicting evidence. As noted in Section 5.2, the current ecosystem is also limited in linguistic and cultural diversity, since most evaluations are conducted in English-centric settings. Computational constraints mean that many studies still focus on mid-sized models, moderate context windows, and relatively small evaluation sets, leaving open whether current conclusions generalize to frontier-scale models and real deployment conditions [4,22,28,38,67,68,88,89,109,110,116].

6.2. Future Research Directions

Building on these challenges, future research on selection bias in LLMs can advance along several directions.

A first direction is to move toward deeper mechanistic understanding and debiasing by design. Future studies should transition from *post hoc* remedies to preventive design, investigating not only which components cause bias but also why and how they encode it. Tools such as graph-based analysis, causal inference, and mechanistic interpretability can support the development of Transformer

 doi:10.20944/preprints202604.2234.v1

architectures with intrinsic bias immunity, namely models that exhibit predictable, task-aligned positional behaviors through intentional design choices [54,65,66,80].

A second direction is to develop more general and adaptive mitigation frameworks. This includes extending calibration-based methods such as CalibraEval beyond pairwise comparison to broader ranking tasks, and applying causal-debiasing techniques to multi-source or cross-domain bias settings. Future work should also reduce reliance on external resources by exploring debiasing approaches that draw on the model's internal knowledge. The identification of globally biased components, namely internal modules that consistently propagate bias signals across tasks, is another promising direction.

A third direction is to build a comprehensive evaluation ecosystem. The community should develop a standardized evaluation framework, including reliable bias metrics that are valid across cultures and languages, and should test mitigation methods on larger models, longer contexts, and more complex scenarios such as RAG tasks involving multiple correct answers or conflicting evidence. Evaluation paradigms should evolve beyond pairwise setups to include list ranking, score calibration, and context-sensitive bias detection [33,88–90,122].

A fourth direction is to expand to new domains, modalities, and paradigms. The study of selection bias should extend beyond text-based LLMs to multimodal and cross-lingual settings. Emerging evidence from video-language models points to underexplored patterns of bias in multimodal reasoning. The principles of debiasing can also inform generative models beyond language, such as diffusion models in image generation, where phenomena such as reward over-optimization reflect related bias mechanisms. Investigating how cognitive biases manifest in human–AI collaborative decision-making is also important for the safety and reliability of intelligent systems [26,139–144].

A fifth direction is to leverage reinforcement learning and self-corrective signals for dynamic and adaptive debiasing. Reinforcement learning provides a feedback-driven alternative to static post-hoc calibration, allowing models to iteratively align their internal decision processes with bias-invariant objectives. Recent work in this direction explores reward shaping and policy optimization frameworks such as Proximal Policy Optimization (PPO), Direct Preference Optimization (DPO), and Generalized Reward Policy Optimization (GRPO) for bias-aware fine-tuning [72–75]. Incorporating causal or uncertainty-aware components into these objectives may enable models to autonomously discover and suppress bias-inducing patterns during training. A complementary line of work uses the performance disparity created by position bias as supervision rather than eliminating it directly, illustrated by inter-position knowledge distillation [58], suggesting that bias signals themselves can serve as a training resource for self-corrective and bias-resilient language understanding.

7. Conclusion

This survey has framed selection bias in large language models as a failure of decision invariance under equivalence-preserving input perturbations, including option permutation, label renaming, candidate-order swapping, and evidence relocation [1,14,30,38]. Under this framing, selection bias unifies several related phenomena that have so far been studied in isolation, including option-order bias, label and token bias, and the lost-in-the-middle effect in long-context reasoning. Treating these phenomena as different surface manifestations of the same invariance failure clarifies why a method that mitigates one form of selection bias often does not address another, and why progress in this area depends on the joint design of evaluation protocols and mitigation strategies rather than on either component alone.

Three broad lessons emerge from the literature reviewed in this survey. First, selection bias is best understood as a full-chain phenomenon. It originates in statistical asymmetries embedded in pretraining and instruction-tuning data, is amplified by the structural properties of decoder-only Transformers (causal masking and the distance sensitivity of relative positional encodings), and is consolidated by post-training procedures that reward content-correlated shortcuts [1,14,15,38,48,54,80,87]. Mitigation methods should therefore be evaluated not only by how much they reduce bias on a single benchmark, but also by where in this chain they act and what they leave untouched. Second, the intervention-level

 doi:10.20944/preprints202604.2234.v1

taxonomy developed in Section 4 suggests a clear pattern of trade-offs: inference-time methods are inexpensive and broadly applicable but rarely sufficient on their own [19,26–28,37]; architectural modifications offer principled invariance but require white-box access and can break down when options are interdependent [7,27,54,55,69,85,130]; training-level methods provide the deepest debiasing but at the cost of training resources and benchmark-specific design [51,56,58,59]. The most promising near-term progress is likely to come from combining methods across levels, for example pairing prompt-level uncertainty injection with internal component pruning or with permutation-aware fine-tuning [19,27,28,54–56,59]. Third, the evaluation ecosystem remains the field's central bottleneck. Without standardized cross-task protocols, comparable metrics, and benchmarks that go beyond pairwise English-centric MCQA, claims of debiasing remain difficult to compare across studies, and the transferability of findings to frontier-scale models is largely untested [29,91,106,107,109,110,116,142]. The same observation extends to deployed application domains, where selection bias has measurable consequences for MCQA, dialogue, retrieval-augmented generation, and agentic interaction [1,29,30,38,47,123,135].

Looking forward, two specific questions appear most urgent. The first is mechanistic: which architectural and data factors are necessary for the emergence of position bias, and which are merely correlated with it? Answering this question would move the field from symptom-level mitigation to architectural design that is permutation-aware by construction. The second is empirical: how do the findings reviewed here generalize as model scale, context length, and tool use increase? Most current evidence comes from mid-sized open-source models on relatively short contexts, while deployed systems increasingly operate at much larger scale and in agentic settings where selection bias can compound across turns. Progress on both fronts will be necessary for selection-invariant language models to become a reliable foundation for evaluation, reasoning, and real-world deployment [14,30,38,54,56].

References

1. Jiang, B.; Xie, Y.; Hao, Z.; Wang, X.; Mallick, T.; Su, W.J.; Taylor, C.J.; Roth, D. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y.; Bansal, M.; Chen, Y., Eds., Miami, Florida, USA, 2024; pp. 4722–4756. https://doi.org/10.18653/v1/2024.emnlp-main.272.
2. Kosinski, M. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences* **2024**, *121*, e2405460121.
3. Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Li, Y.; Lundberg, S.; et al. Sparks of Artificial General Intelligence: Early experiments with GPT-4, 2023, [arXiv:cs.CL/2303.12712].
4. OpenAI. GPT-4 Technical Report, 2023, [arXiv:cs.CL/2303.08774].
5. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research* **2024**, *25*, 1–53.
6. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
7. Zhou, H.; Feng, Z.; Zhu, Z.; Qian, J.; Mao, K. Unibias: Unveiling and mitigating llm bias through internal attention and ffn manipulation. *Advances in Neural Information Processing Systems* **2024**, *37*, 102173–102196.
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2019; pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.
9. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention, 2021, [arXiv:cs.CL/2006.03654]. International Conference on Learning Representations (ICLR 2021).
10. Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N.A.; Khashabi, D.; Hajishirzi, H. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023; pp. 13484–13508. https://doi.org/10.18653/v1/2023.acl-long.754.

 doi:10.20944/preprints202604.2234.v1

11. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.H.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 2022; Vol. 35, pp. 24824–24837.
12. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners, 2022, [arXiv:cs.CL/2205.11916].
13. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.H.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models, 2022, [arXiv:cs.CL/2203.11171].
14. Wei, S.L.; Wu, C.K.; Huang, H.H.; Chen, H.H. Unveiling Selection Biases: Exploring Order and Token Sensitivity in Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024; Ku, L.W.; Martins, A.; Srikumar, V., Eds., Bangkok, Thailand, 2024; pp. 5598–5621. https://doi.org/10.18653/v1/2024.findings-acl.333.
15. Si, C.; Friedman, D.; Joshi, N.; Feng, S.; Chen, D.; He, H. Measuring Inductive Biases of In-Context Learning with Underspecified Demonstrations. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Rogers, A.; Boyd-Graber, J.; Okazaki, N., Eds., Toronto, Canada, 2023; pp. 11289–11310. https://doi.org/10.18653/v1/2023.acl-long.632.
16. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023, [arXiv:cs.CL/2307.09288].
17. Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; Chen, W. What Makes Good In-Context Examples for GPT-3? In Proceedings of the Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Dublin, Ireland and Online, 2022; pp. 100–114. https://doi.org/10.18653/v1/2022.deelio-1.10.
18. Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; Stenetorp, P. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity, 2021, [arXiv:cs.CL/2104.08786].
19. Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; Singh, S. Calibrate Before Use: Improving Few-shot Performance of Language Models. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning, Virtual, 2021; Vol. 139, *Proceedings of Machine Learning Research*, pp. 12697–12706.
20. Yang, Z.; Jian, P.; Li, C. Option Symbol Matters: Investigating and Mitigating Multiple-Choice Option Symbol Bias of Large Language Models. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Albuquerque, New Mexico, 2025; pp. 1902–1917. https://doi.org/10.18653/v1/2025.naacl-long.95.
21. Wang, H.; Zhao, S.; Qiang, Z.; Xi, N.; Qin, B.; Liu, T. LLMs May Perform MCQA by Selecting the Least Incorrect Option, 2024, [arXiv:cs.CL/2402.01349].
22. Pezeshkpour, P.; Hruschka, E. Large Language Models Sensitivity to The Order of Options in Multiple-Choice Questions. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, 2024; pp. 2006–2017. https://doi.org/10.18653/v1/2024.findings-naacl.130.
23. Zheng, C.; Zhou, H.; Meng, F.; Zhou, J.; Huang, M. Large Language Models Are Not Robust Multiple Choice Selectors, 2024, [arXiv:cs.CL/2309.03882]. ICLR 2024 poster.
24. Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; Zettlemoyer, L. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 2022; pp. 11048–11064.
25. Reynolds, L.; McDonell, K. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm, 2021, [arXiv:cs.CL/2102.07350].
26. Loginova, O.; Bezrukov, O.; Shekhar, R.; Kravets, A. Addressing Blind Guessing: Calibration of Selection Bias in Multiple-Choice Question Answering by Video Language Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 3216–3246. https://doi.org/10.18653/v1/2025.acl-long.162.
27. Choi, H.K.; Xu, W.; Xue, C.; Eckman, S.; Reddy, C.K. Mitigating Selection Bias with Node Pruning and Auxiliary Options. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 5190–5215. https://doi.org/10.18653/v1/2025.acl-long.259.

 doi:10.20944/preprints202604.2234.v1

28. Li, H.; Chen, J.; Ai, Q.; Chu, Z.; Zhou, Y.; Dong, Q.; Liu, Y. CalibraEval: Calibrating Prediction Distribution to Mitigate Selection Bias in LLMs-as-Judges. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 16537–16552. https://doi.org/10.18653/v1/2025.acl-long.808.
29. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems* **2023**, *36*, 46595–46623.
30. Shi, L.; Ma, C.; Liang, W.; Ma, W.; Vosoughi, S. Judging the Judges: A Systematic Study of Position Bias in LLM-as-a-Judge, 2024, [arXiv:cs.CL/2406.07791].
31. Raina, V.; Liusie, A.; Gales, M. Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, 2024; pp. 7499–7517. https://doi.org/10.18653/v1/2024.emnlp-main.427.
32. Li, Z.; Wang, C.; Ma, P.; Wu, D.; Wang, S.; Gao, C.; Liu, Y. Split and Merge: Aligning Position Biases in LLM-based Evaluators. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, 2024; pp. 11084–11108. https://doi.org/10.18653/v1/2024.emnlp-main.621.
33. Zeng, Z.; Yu, J.; Gao, T.; Meng, Y.; Goyal, T.; Chen, D. Evaluating Large Language Models at Evaluating Instruction Following, 2023, [arXiv:cs.CL/2310.07641].
34. Karpinska, M.; Akoury, N.; Iyyer, M. The Perils of Using Mechanical Turk to Evaluate Open-Ended Text Generation. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 2021; pp. 1265–1285. https://doi.org/10.18653/v1/2021.emnlp-main.97.
35. Wang, P.; Li, L.; Chen, L.; Cai, Z.; Zhu, D.; Lin, B.; Cao, Y.; Kong, L.; Liu, Q.; Liu, T.; et al. Large Language Models are not Fair Evaluators. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 2024; pp. 9440–9450. https://doi.org/10.18653/v1/2024.acl-long.511.
36. Lampinen, A.K.; Dasgupta, I.; Chan, S.C.Y.; Sheahan, H.R.; Creswell, A.; Kumaran, D.; McClelland, J.L.; Hill, F. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus* **2024**, *3*, pgae233. https://doi.org/10.1093/pnasnexus/pgae233.
37. Zhang, M.; Meng, Z.; Collier, N. Can We Instruct LLMs to Compensate for Position Bias? In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, 2024; pp. 12545–12556. https://doi.org/10.18653/v1/2024.findings-emnlp.732.
38. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics* **2024**, *12*, 157–173.
39. Yu, Y.; Jiang, H.; Luo, X.; Wu, Q.; Lin, C.Y.; Li, D.; Yang, Y.; Huang, Y.; Qiu, L. Mitigate Position Bias in Large Language Models via Scaling a Single Hidden States Channel. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, 2025; pp. 6092–6111.
40. Li, T.; Zhang, G.; Do, Q.D.; Yue, X.; Chen, W. Long-context LLMs Struggle with Long In-context Learning, 2025, [arXiv:cs.CL/2404.02060]. Accepted by Transactions on Machine Learning Research (TMLR).
41. Li, J.; Wang, M.; Zheng, Z.; Zhang, M. LooGLE: Can Long-Context Language Models Understand Long Contexts? In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 2024; pp. 16304–16333. https://doi.org/10.18653/v1/2024.acl-long.859.
42. Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E.H.; Schärli, N.; Zhou, D. Large Language Models Can Be Easily Distracted by Irrelevant Context. In Proceedings of the Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA, 2023; Vol. 202, *Proceedings of Machine Learning Research*, pp. 31210–31227.
43. Tang, R.; Kong, D.; Huang, L.; Xue, H. Large Language Models Can be Lazy Learners: Analyze Shortcuts in In-Context Learning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, 2023; pp. 4645–4657. https://doi.org/10.18653/v1/2023.findings-acl.284.

 doi:10.20944/preprints202604.2234.v1

44. Weller, O.; Khan, A.; Weir, N.; Lawrie, D.; Van Durme, B. Defending Against Disinformation Attacks in Open-Domain Question Answering. In Proceedings of the Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), St. Julian's, Malta, 2024; pp. 402–417. https://doi.org/10.18653/v1/2024.eacl-short.35.
45. Oh, P.; Thorne, J. Detrimental Contexts in Open-Domain Question Answering. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 2023; pp. 11589–11605. https://doi.org/10.18653/v1/2023.findings-emnlp.776.
46. Fang, J.; Meng, Z.; MacDonald, C. TRACE the Evidence: Constructing Knowledge-Grounded Reasoning Chains for Retrieval-Augmented Generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, 2024; pp. 8472–8494. https://doi.org/10.18653/v1/2024.findings-emnlp.496.
47. Fan, S.; Wei, W.; Li, W.; Mao, X.L.; Xie, W.; Chen, D. Position Debiasing Fine-Tuning for Causal Perception in Long-Term Dialogue. In Proceedings of the Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24, Jeju, Korea, 2024; pp. 6261–6269. https://doi.org/10.24963/ijcai.2024/692.
48. Liu, Y.; Zeng, X.; Shao, C.; Meng, F.; Zhou, J. Instruction Position Matters in Sequence Generation with Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 2024; pp. 11652–11663. https://doi.org/10.18653/v1/2024.findings-acl.693.
49. Fei, Y.; Hou, Y.; Chen, Z.; Bosselut, A. Mitigating Label Biases for In-context Learning. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023; pp. 14014–14031. https://doi.org/10.18653/v1/2023.acl-long.783.
50. Reif, Y.; Schwartz, R. Beyond Performance: Quantifying and Mitigating Label Bias in LLMs. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Mexico City, Mexico, 2024; pp. 6784–6798. https://doi.org/10.18653/v1/2024.naacl-long.378.
51. Zhang, Y.; Li, B.; Ling, Z.; Zhou, F. Mitigating Label Bias in Machine Learning: Fairness through Confident Learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, Canada, 2024; Vol. 38, pp. 16917–16925. https://doi.org/10.1609/aaai.v38i15.29634.
52. Jang, J.; Jang, S.; Kweon, W.; Jeon, M.; Yu, H. Rectifying Demonstration Shortcut in In-Context Learning. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Mexico City, Mexico, 2024; pp. 4294–4321. https://doi.org/10.18653/v1/2024.naacl-long.242.
53. Wang, X.; Liu, X. Beyond Generation: Leveraging LLM Creativity to Overcome Label Bias in Classification. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, 2025; pp. 25500–25506. https://doi.org/10.18653/v1/2025.findings-acl.1307.
54. Wang, Z.; Zhang, H.; Li, X.; Huang, K.H.; Han, C.; Ji, S.; Kakade, S.M.; Peng, H.; Ji, H. Eliminating Position Bias of Language Models: A Mechanistic Approach, 2025, [arXiv:cs.CL/2407.01100]. Published as a conference paper at ICLR 2025.
55. McIlroy-Young, R.; Brown, K.; Olson, C.; Zhang, L.; Dwork, C. Set-Based Prompting: Provably Solving the Language Model Order Dependency Problem, 2024, [arXiv:cs.LG/2406.06581].
56. Zhou, F.; Mao, Y.; Yu, L.; Yang, Y.; Zhong, T. Causal-Debias: Unifying Debiasing in Pretrained Language Models and Fine-tuning via Causal Invariant Learning. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023; pp. 4227–4241. https://doi.org/10.18653/v1/2023.acl-long.232.
57. Liu, Z.; Chen, Z.; Zhang, M.; Ren, Z.; Ren, P.; Chen, Z. Self-Supervised Position Debiasing for Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 2024; pp. 2897–2917. https://doi.org/10.18653/v1/2024.findings-acl.170.
58. Wang, Y.; Xiong, F.; Wang, Y.; Li, L.; Chu, X.; Zeng, D.D. Position Bias Mitigates Position Bias: Mitigate Position Bias Through Inter-Position Knowledge Distillation. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP), Suzhou, China, 2025; pp. 1495–1512.
59. Zheng, J.; Yuan, J.; Yao, J.; Gu, C.; Zheng, P.; He, G. Mitigating Selection Bias in Large Language Models via Permutation-Aware GRPO, 2026, [arXiv:cs.CL/2603.21016]. https://doi.org/10.48550/arXiv.2603.21016.
60. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, California, USA, 2017; pp. 5998–6008.

 doi:10.20944/preprints202604.2234.v1

61. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* **2020**, *21*, 1–67.
62. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020; pp. 7871–7880.
63. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; et al. PaLM: Scaling Language Modeling with Pathways, 2022, [arXiv:cs.CL/2204.02311].
64. Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2021, [arXiv:cs.CL/2104.09864].
65. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-Attention with Relative Position Representations. In Proceedings of the Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, 2018; pp. 464–468. https://doi.org/10.18653/v1/N18-2074.
66. Press, O.; Smith, N.A.; Lewis, M. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation, 2022. International Conference on Learning Representations.
67. Peng, B.; Quesnelle, J.; Fan, H.; Shippole, E. YaRN: Efficient Context Window Extension of Large Language Models, 2023, [arXiv:cs.CL/2309.00071].
68. Ding, Y.; Zhang, L.L.; Zhang, C.; Xu, Y.; Shang, N.; Xu, J.; Yang, F.; Yang, M. LongRoPE: Extending LLM Context Window Beyond 2 Million Tokens, 2024, [arXiv:cs.CL/2402.13753].
69. Zhang, Z.; Chen, R.; Liu, S.; Yao, Z.; Ruwase, O.; Chen, B.; Wu, X.; Wang, Z.; et al. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *Advances in Neural Information Processing Systems* **2024**, *37*, 60755–60775.
70. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback, 2022, [arXiv:cs.CL/2203.02155].
71. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; et al. Constitutional AI: Harmlessness from AI Feedback, 2022, [arXiv:cs.CL/2212.08073].
72. Christiano, P.F.; Leike, J.; Brown, T.B.; Martic, M.; Legg, S.; Amodei, D. Deep Reinforcement Learning from Human Preferences. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 2017; Vol. 30, pp. 4299–4307.
73. Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; Christiano, P.F. Learning to Summarize with Human Feedback. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 2020; Vol. 33, pp. 3008–3021.
74. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms, 2017, [arXiv:cs.LG/1707.06347].
75. Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C.D.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2023, [arXiv:cs.LG/2305.18290].
76. Lin, S.; Hilton, J.; Evans, O. TruthfulQA: Measuring How Models Mimic Human Falsehoods, 2021, [arXiv:cs.CL/2109.07958].
77. Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; Gardner, M. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2019; pp. 2368–2378. https://doi.org/10.18653/v1/N19-1246.
78. Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. Training Verifiers to Solve Math Word Problems, 2021, [arXiv:cs.LG/2110.14168].
79. Goldfarb-Tarrant, S.; Marchant, R.; Sánchez, R.M.; Pandya, M.; Lopez, A. Intrinsic Bias Metrics Do Not Correlate with Application Bias. In Proceedings of the 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), Online, 2021; pp. 1926–1940. https://doi.org/10.18653/v1/2021.acl-long.150.

 doi:10.20944/preprints202604.2234.v1

80. Adiga, R.; Nushi, B.; Chandrasekaran, V. Attention Speaks Volumes: Localizing and Mitigating Bias in Language Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 26403–26423. https://doi.org/10.18653/v1/2025.acl-long.1281.
81. Ratner, N.; Levine, Y.; Belinkov, Y.; Ram, O.; Magar, I.; Abend, O.; Karpas, E.; Shashua, A.; Leyton-Brown, K.; Shoham, Y. Parallel Context Windows for Large Language Models. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023; pp. 6383–6402. https://doi.org/10.18653/v1/2023.acl-long.352.
82. He, Z.; Jiang, H.; Wang, Z.; Yang, Y.; Qiu, L.K.; Qiu, L. Position engineering: Boosting large language models through positional information manipulation. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, 2024; pp. 7333–7345.
83. Egressy, B.; Stühmer, J. Set-LLM: A Permutation-Invariant LLM, 2025, [arXiv:cs.LG/2505.15433]. arXiv preprint; arXiv comments indicate acceptance to ACL 2025 (main), https://doi.org/10.48550/arXiv.2505.15433.
84. Yoon, S.; Ahn, D.; Lee, Y.; Jung, M.; Jang, H.; Hwang, S.w. RoToR: Towards More Reliable Responses for Order-Invariant Inputs. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 18739–18760. https://doi.org/10.18653/v1/2025.acl-long.918.
85. Kinder, L.; Edman, L.; Fraser, A.; Käfer, T. Positional Overload: Positional Debiasing and Context Window Extension for Large Language Models using Set Encoding. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 3896–3908. https://doi.org/10.18653/v1/2025.acl-long.197.
86. Liusie, A.; Fathullah, Y.; Gales, M. Teacher-student training for debiasing: General permutation debiasing for large language models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 2024; pp. 1376–1387.
87. Meade, N.; Poole-Dayan, E.; Reddy, S. An Empirical Survey of the Effectiveness of Debiasing Techniques for Pre-trained Language Models. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 2022; pp. 1878–1898. https://doi.org/10.18653/v1/2022.acl-long.132.
88. Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; Hou, L. Instruction-Following Evaluation for Large Language Models, 2023, [arXiv:cs.CL/2311.07911].
89. Zhu, L.; Wang, X.; Wang, X. JudgeLM: Fine-Tuned Large Language Models are Scalable Judges, 2023, [arXiv:cs.CL/2310.17631].
90. Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; Zhu, C. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 2023; pp. 2511–2522. https://doi.org/10.18653/v1/2023.emnlp-main.153.
91. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring Massive Multitask Language Understanding, 2020, [arXiv:cs.CY/2009.03300].
92. Gema, A.P.; Leang, J.O.J.; Hong, G.; Saxena, R.; Devoto, A.; Mancino, A.C.M.; He, X.; Zhao, Y.; Du, X.; Madani, M.R.G.; et al. Are We Done with MMLU?, 2024, [arXiv:cs.CL/2406.04127].
93. Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; Tafjord, O. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge, 2018, [arXiv:cs.AI/1803.05457].
94. Liang, Y.; Li, J.; Yin, J. A New Multi-choice Reading Comprehension Dataset for Curriculum Learning. In Proceedings of the Proceedings of The Eleventh Asian Conference on Machine Learning, Nagoya, Japan, 2019; Vol. 101, *Proceedings of Machine Learning Research*, pp. 742–757.
95. Talmor, A.; Herzig, J.; Lourie, N.; Berant, J. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2019; pp. 4149–4158.
96. Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; Choi, Y. HellaSwag: Can a Machine Really Finish Your Sentence? In Proceedings of the Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019; pp. 4791–4800.
97. Sakaguchi, K.; Bras, R.L.; Bhagavatula, C.; Choi, Y. WinoGrande: An Adversarial Winograd Schema Challenge at Scale, 2020, [arXiv:cs.CL/1907.10641].

 doi:10.20944/preprints202604.2234.v1

98. Mihaylov, T.; Clark, P.; Khot, T.; Sabharwal, A. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In Proceedings of the Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018; pp. 2381–2391.
99. Bisk, Y.; Zellers, R.; Le Bras, R.; Gao, J.; Choi, Y. PIQA: Reasoning about Physical Commonsense in Natural Language, 2020, [arXiv:cs.CL/1911.11641].
100. Sap, M.; Rashkin, H.; Chen, D.; Le Bras, R.; Choi, Y. Social IQa: Commonsense Reasoning about Social Interactions. In Proceedings of the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019; pp. 4463–4473.
101. Huang, L.; Le Bras, R.; Bhagavatula, C.; Choi, Y. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In Proceedings of the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019; pp. 2391–2401.
102. Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; Hovy, E. RACE: Large-scale ReAding Comprehension Dataset From Examinations, 2017, [arXiv:cs.CL/1704.04683].
103. Yu, W.; Jiang, Z.; Dong, Y.; Feng, J. ReClor: A Reading Comprehension Dataset Requiring Logical Reasoning. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 2020.
104. Srivastava, A.; Rastogi, A.; Rao, A.; Shoeb, A.A.M.; et al. Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models, 2022, [arXiv:cs.CL/2206.04615].
105. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, 2013; pp. 1631–1642.
106. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium, 2018; pp. 353–355.
107. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems, 2019, [arXiv:cs.CL/1905.00537].
108. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Proceedings of the Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018; pp. 2369–2380.
109. Hsieh, C.P.; Sun, S.; Kriman, S.; Acharya, S.; Rekesh, D.; Jia, F.; Zhang, Y.; Ginsburg, B. RULER: What's the Real Context Size of Your Long-Context Language Models?, 2024, [arXiv:cs.CL/2404.06654].
110. Bai, Y.; Lv, X.; Zhang, J.; Lyu, H.; Tang, J.; Huang, Z.; Du, Z.; Liu, X.; Zeng, A.; Hou, L.; et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 2024; pp. 3119–3137.
111. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 2020; Vol. 33, pp. 9459–9474.
112. Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; Chang, M.W. REALM: Retrieval-Augmented Language Model Pre-Training. In Proceedings of the Proceedings of the 37th International Conference on Machine Learning, Virtual, 2020; Vol. 119, *Proceedings of Machine Learning Research*, pp. 3929–3938.
113. Izacard, G.; Grave, E. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering, 2021, [arXiv:cs.CL/2007.01282].
114. Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* **2019**, *7*, 453–466.

 doi:10.20944/preprints202604.2234.v1

115. Saito, K.; Lee, C.Y.; Sohn, K.; Ushiku, Y. Where is the answer? An empirical study of positional bias for parametric knowledge extraction in language model. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Albuquerque, New Mexico, 2025; pp. 1252–1269. https://doi.org/10.18653/v1/2025.naacl-long.58.
116. Lambert, N.; Pyatkin, V.; Morrison, J.; Miranda, L.J.V.; Lin, B.Y.; Chandu, K.; Dziri, N.; Kumar, S.; Zick, T.; Choi, Y.; et al. RewardBench: Evaluating Reward Models for Language Modeling. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, 2025; pp. 1755–1797.
117. Kim, S.; Suk, J.; Longpre, S.; Lin, B.Y.; Shin, J.; Welleck, S.; Neubig, G.; Lee, M.; Lee, K.; Seo, M. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, 2024; pp. 4334–4353.
118. Huang, Z.; Qiu, Z.; Wang, Z.; Ponti, E.M.; Titov, I. Post-hoc Reward Calibration: A Case Study on Length Bias, 2024, [arXiv:cs.CL/2409.17407].
119. Dubois, Y.; Li, C.X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.S.; Hashimoto, T.B. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems* **2023**, *36*, 30039–30069.
120. Dubois, Y.; Galambosi, B.; Liang, P.; Hashimoto, T.B. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators, 2024, [arXiv:cs.CL/2404.04475].
121. Fabbri, A.R.; Kryściński, W.; McCann, B.; Xiong, C.; Socher, R.; Radev, D. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* **2021**, *9*, 391–409.
122. Wang, Y.; Yu, Z.; Zeng, Z.; Yang, L.; Wang, C.; Chen, H.; Jiang, C.; Xie, R.; Wang, J.; Xie, X.; et al. PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization, 2023, [arXiv:cs.CL/2306.05087].
123. Du, M.; He, F.; Zou, N.; Tao, D.; Hu, X. Shortcut learning of large language models in natural language understanding. *Communications of the ACM* **2024**, *67*, 110–120.
124. Clark, C.; Lee, K.; Chang, M.W.; Kwiatkowski, T.; Collins, M.; Toutanova, K. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2019; pp. 2924–2936. https://doi.org/10.18653/v1/N19-1300.
125. Ma, M.D.; Kao, J.Y.; Gupta, A.; Lin, Y.H.; Zhao, W.; Chung, T.; Wang, W.; Chang, K.W.; Peng, N. Mitigating Bias for Question Answering Models by Tracking Bias Influence. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Mexico City, Mexico, 2024; pp. 4592–4610. https://doi.org/10.18653/v1/2024.naacl-long.257.
126. Li, H.; Xu, T.; Yang, K.; Chu, Y.; Chen, Y.; Song, Y.; Wen, Q.; Liu, H. Ask-Before-Detection: Identifying and Mitigating Conformity Bias in LLM-Powered Error Detector for Math Word Problem Solutions. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 1597–1609. https://doi.org/10.18653/v1/2025.acl-long.80.
127. Li, X.; Wang, W.; Li, M.; Guo, J.; Zhang, Y.; Feng, F. Evaluating Mathematical Reasoning of Large Language Models: A Focus on Error Identification and Correction, 2024, [arXiv:cs.CL/2406.00755].
128. Zhou, Z.; Liu, S.; Ning, M.; Liu, W.; Wang, J.; Wong, D.F.; Huang, X.; Wang, Q.; Huang, K. Is Your Model Really A Good Math Reasoner? Evaluating Mathematical Reasoning with Checklist, 2024, [arXiv:cs.CL/2407.08733].
129. Wan, D.; Vig, J.; Bansal, M.; Joty, S. On Positional Bias of Faithfulness for Long-Form Summarization. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Albuquerque, New Mexico, 2025; pp. 8791–8810.
130. An, S.; Ma, Z.; Lin, Z.; Zheng, N.; Lou, J.G.; Chen, W. Make your llm fully utilize the context. *Advances in Neural Information Processing Systems* **2024**, *37*, 62160–62188.

 doi:10.20944/preprints202604.2234.v1

131. Wu, P.; Li, H.; Deng, Y.; Hu, W.; Dai, Q.; Dong, Z.; Sun, J.; Zhang, R.; Zhou, X.H. On the Opportunity of Causal Learning in Recommendation Systems: Foundation, Estimation, Prediction and Challenges. In Proceedings of the Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, Vienna, Austria, 2022; pp. 5646–5653. Survey Track, https://doi.org/10.24963/ijcai.2022/787.
132. Chen, J.; Dong, H.; Wang, X.; Feng, F.; Wang, M.; He, X. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* **2023**, *41*, 1–39. https://doi.org/10.1145/3564284.
133. Liu, Y.; Wei, W.; Liu, J.; Mao, X.; Fang, R.; Chen, D. Improving Personality Consistency in Conversation by Persona Extending. In Proceedings of the Proceedings of the 31st ACM International Conference on Information and Knowledge Management, Atlanta, GA, USA, 2022; pp. 1350–1359. https://doi.org/10.1145/3511808.3557359.
134. Lu, Z.; Wei, W.; Qu, X.; Mao, X.L.; Chen, D.; Chen, J. Miracle: Towards Personalized Dialogue Generation with Latent-Space Multiple Personal Attribute Control. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 2023; pp. 5933–5957. https://doi.org/10.18653/v1/2023.findings-emnlp.395.
135. Chen, Y.; Lv, A.; Lin, T.E.; Chen, C.; Wu, Y.; Huang, F.; Li, Y.; Yan, R. Fortify the Shortest Stave in Attention: Enhancing Context Awareness of Large Language Models for Effective Tool Use. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 2024; pp. 11160–11174. https://doi.org/10.18653/v1/2024.acl-long.601.
136. Li, B.; Wu, W.; Tang, Z.; Shi, L.; Yang, J.; Li, J.; Yao, S.; Qian, C.; Hui, B.; Zhang, Q.; et al. DevBench: A Comprehensive Benchmark for Software Development, 2024, [arXiv:cs.SE/2403.08604].
137. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models, 2023, [arXiv:cs.CL/2210.03629].
138. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools, 2023, [arXiv:cs.CL/2302.04761].
139. Zhang, Z.; Zhang, S.; Zhan, Y.; Luo, Y.; Wen, Y.; Tao, D. Confronting Reward Overoptimization for Diffusion Models: A Perspective of Inductive and Primacy Biases, 2024, [arXiv:cs.LG/2402.08552].
140. An, H.; Acquaye, C.; Wang, C.; Li, Z.; Rudinger, R. Do Large Language Models Discriminate in Hiring Decisions on the Basis of Race, Ethnicity, and Gender?, 2024, [arXiv:cs.CY/2406.10486].
141. Mina, M.; Ruiz-Fernández, V.; Falcão, J.; Vasquez-Reina, L.; Gonzalez-Agirre, A. Cognitive Biases, Task Complexity, and Result Interpretability in Large Language Models. In Proceedings of the Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, UAE, 2025; pp. 1767–1784.
142. Parrish, A.; Chen, A.; Nangia, N.; Padmakumar, V.; Phang, J.; Thompson, J.; Htut, P.M.; Bowman, S. BBQ: A hand-built bias benchmark for question answering. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 2022; pp. 2086–2105. https://doi.org/10.18653/v1/2022.findings-acl.165.
143. Nadeem, M.; Bethke, A.; Reddy, S. StereoSet: Measuring Stereotypical Bias in Pretrained Language Models, 2021, [arXiv:cs.CL/2004.09456].
144. Nangia, N.; Vania, C.; Bhalerao, R.; Bowman, S.R. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 2020; pp. 1953–1967. https://doi.org/10.18653/v1/2020.emnlp-main.154.