Promoting Data Science

Volume 6 2026 Number 1

Journal of Behavioral Data Science V6N1 (2026)

https://isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

ISBN: 2575-8306 (Print) 2574-1284 (Online)

https://jbds.isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

No Publication Charge and Open Access

jbds@isdsa.org

List of Articles

Journal of Behavioral Data Science, 2026, *6* (1), 1-40.
DOI: https://doi.org/10.35566/jbds/choiwu

Zero-Frequency Cell Correction Strategies in Tetrachoric Correlation Estimation: Expanded Strategies and Multivariate Implications

Jeongwon Choi[1 [0000−0001−6087−2124]] and Hao Wu[1 [0000−0001−6471−1774]]

Vanderbilt University, Nashville, TN 37203, USA
jeongwon.choi@vanderbilt.edu, hao.wu.1@vanderbilt.edu

Abstract. Zero-frequency cells pose a challenge for tetrachoric correlation estimation, but investigation of correction strategies remains limited. This study evaluates several zero-cell correction strategies, including different values to add, different ways to add the value, and the use of unadjusted versus adjusted thresholds in the second stage in the two-stage procedure. These strategies are examined across different correlation sizes and thresholds, to estimate a single tetrachoric correlation and extended to multivariate applications involving a tetrachoric correlation matrix and a confirmatory factor analysis model for binary data. Using multiple evaluation criteria, we show how these strategies perform differently across correlation sizes and the pattern of thresholds. This study also introduces ways to improve computational efficiency for tetrachoric correlation simulation studies that leverage the discrete structure to reduce redundant computations.

Keywords: Tetrachoric Correlation · Zero-frequency Cells · Binary Data · Tetrachoric Correlation Matrix · Confirmatory Factor Analysis

1 Introduction

1.1 Polychoric and Tetrachoric Correlations

Psychological research often produces ordered categorical data. In this situation, such variables are assumed to have arisen from discretizing multivariate-normally distributed underlying responses by thresholds. The correlations in this multivariate normal distribution are polychoric correlations. Their estimation has wide applications in structural equation modeling and item factor analysis. A special case of the polychoric correlation is the tetrachoric correlation between two binary variables.

The two-stage procedure (Olsson, 1979) is the most widely used approach for estimating polychoric correlations. It first estimates the thresholds for each variable using its marginal category proportions and then for each pair of variables

maximizes the likelihood of the proportions in their two-way contingency table as a function of the correlation, treating the thresholds as fixed. For the special case of estimating a single tetrachoric correlation, the two-stage procedure is equivalent to maximum likelihood where the two thresholds and the single correlation are estimated jointly in a single stage to maximize the likelihood of the observed proportions, because both procedures would produce estimates that can perfectly reproduce the observed proportions.

1.2 The Zero-Frequency Issue in Tetrachoric Correlation Estimation

One major issue in estimating a polychoric correlation is the presence of zero-frequency cells in the contingency table. Zero-frequency cells are common when the sample size is small (e.g., less than 200), when the estimated thresholds are extreme, or when the underlying correlation of the variable is high (Savalei, 2011).

The zero-frequency issue in 2×2 tables is distinctive because a bivariate normal distribution with a perfect correlation can exactly reproduce the observed proportions in a 2×2 table that contains one zero cell. This means that even a single zero cell pushes the best-fitting correlation to the boundary (± 1) , whereas larger tables do not have this property.

Figure 1 illustrates this point. The left panel shows the space of two latent responses that produce a 2×2 table with one zero. In this case, a degenerate bivariate normal distribution whose support is on the slanted gray line can perfectly reproduce the observed proportions once discretized by thresholds (represented by the dotted lines) calculated through the observed marginal proportions. This means the correlation of 1, being able to perfectly reproduce the observed proportion, must be the maximizer of the likelihood function, yielding a correlation estimate on the boundary of the parameter space. In contrast, for tables larger than 2×2, a single zero-frequency does not yield a boundary solution, because a bivariate normal distribution with a perfect correlation necessarily results in at least two zero cells. As shown in the right panel of Figure 1, the linear subspace (represented by the gray line) on which a degenerate bivariate normal distribution is supported can go through at most four of the six regions defined by the thresholds, leaving at least two zero-probability cells in the 2×3 table. Because a zero probability cell cannot produce a nonzero count, the analysis above means a boundary correlation of ± 1 would produce a likelihood function value of 0 for bivariate data with only one zero in a contingency table bigger than 2×2 and therefore cannot be an estimate. This difference highlights why the zero-frequency problem is uniquely severe in 2×2 tables. Motivated by this distinctiveness, in this paper we focus on the estimation of tetrachoric correlations among binary variables. Prior research (Savalei, 2011) also found that zero-frequency cell treatment is most relevant for binary data.

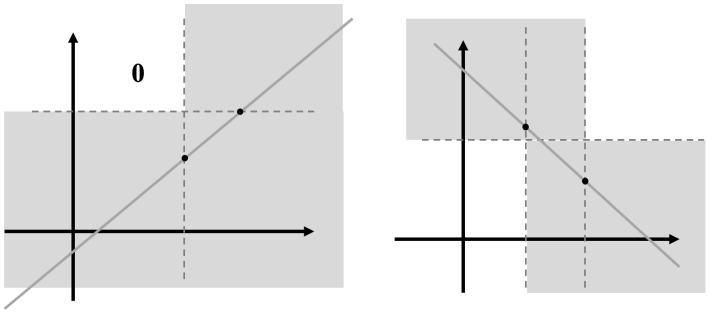


Figure 1: A degenerate bivariate normal distribution. Left: 2×2 table with one zero cell. Right: 2×3 table with two possible zero-cell locations under perfect correlation. The solid line represents a bivariate distribution with perfect correlation; dotted lines indicate thresholds.

1.3 Correction Strategies for Zero-Frequency Cells

The most widely used approach to address the problem of boundary solutions caused by zero-frequency is to add a small value to the zero-frequency cell in the contingency table. This idea was initially introduced by Brown and Benedetti (1977) for tetrachoric correlation based on the idea of Yates (1934)'s correction for continuity. This modification is available in several R (R Core Team, 2022) packages, such as the `PolychoricRM` function in the `Turbofuns` package (Zhang, Trichtinger, Lee, & Jiang, 2022), `polychoric.matrix` function in the `EGAnet` package (Golino & Christensen, 2025), the `lavCor` function in the `lavaan` package (Rosseel, 2012), and the `polychoric` function in the `psych` package (Revelle, 2023). These software programs make it easy for researchers to implement the correction method of adding a small value.

Although the general idea across software packages is to add a small constant, the specific way this correction is applied differs. First, different values can be added. While 0.5 is the most frequently used as a value to add to zero-frequency cells, other values such as 0.1 or the inverse of the total number of cells in the bivariate contingency table (0.25 for calculating a tetrachoric correlation) are also being used. For example, `Turbofuns` offers the option to use values of 0.1, 0.5, and the inverse of the number of cells. It remains unclear which value is optimal. Secondly, one can add the small number only to the zero-frequency (e.g., `lavaan`, `psych`) or to all cells (e.g., `Turbofuns`, `EGAnet`). One can even choose to add to all cells even when zero-frequency is not present within the contingency table. There is also the choice to maintain the marginal proportions (e.g., `lavaan`), which aligns with Brown and Benedetti (1977)'s idea that the remaining cells should also be adjusted according to the substitution of zero cell to maintain the marginal. However, this option is limited to 2×2 tables and is not available for larger contingency tables, which makes it especially relevant to examine in the 2×2 case. Lastly, additional inconsistencies arise in the use of

thresholds in the second stage of the two-stage procedure when zero-frequency cells are present in the contingency table. Mplus (Muthén & Muthén, 2017) and `lavaan` employ thresholds calculated from the uncorrected table (referred to as unadjusted thresholds), whereas several R programs, including `psych` and `Turbofuns`, use thresholds calculated from the table after the zero-frequency cell correction (referred to as adjusted thresholds).

1.4 Prior Research on Zero-Frequency Cell Corrections

While the zero-frequency cell issue is critical in tetrachoric correlation estimation, there has been relatively limited research on these correction strategies. Initial work on this topic carried out by Savalei (2011) focused primarily on comparing the impact of adding 0.5 with that of making no modification for zero-frequency cells. For binary data, their simulation conditions included correlations of 0, 0.3, 0.5, 0.7 and 0.9 combined with various threshold values. Their research concluded with a tentative recommendation of adding 0.5 to the zero cell over making no modification. In particular, their results showed that adding 0.5 to a zero cell yielded more converged replications, a bell shaped sampling distribution for the estimated correlation, smaller empirical standard error, less biased estimated standard error, and, for the more typical situations of small correlations, a better coverage of confidence intervals, while it was also noted that with opposite-signed extreme thresholds, neither adding 0.5 nor not adding performed well.

Yang and Weng (2024) extended Savalei (2011) by incorporating smaller thresholds when comparing two approaches: adding 0.5 only to the zero cell versus leaving zero cells uncorrected. They considered the threshold sets used in Savalei (2011) and added smaller values to represent milder skewness of the observed data. For the larger-threshold conditions (the same as in Savalei, 2011), they found that zero-cell correction was most beneficial, especially under strong correlations with thresholds in the same direction. This result was consistent with Savalei's earlier findings. In these scenarios, adding 0.5 generally produced smaller empirical standard errors, particularly when correlations were not too high (< 0.7). They also conducted a supplementary simulation study evaluating different methods for adding a number, except for smaller thresholds conditions. They reported that different correction techniques, such as adding 0.5 to all cells, adding 0.5 and keeping the marginal counts, and adding the reciprocal of the number of cells, yielded results similar to adding 0.5 only to zero cells. Meanwhile, adding the reciprocal of the number of cells to all cells resulted in lower bias but higher empirical standard error compared to adding 0.5 only to zero cells.

For smaller threshold conditions (i.e., smaller than those in Savalei, 2011), Yang and Weng (2024) recommended leaving zero cells uncorrected. However, this conclusion may reflect a natural limitation of their simulation design. As shown in their Table 3, the number of replications with zero-frequency cells is extremely small under milder threshold conditions. For example, with small and same-direction thresholds, the average number of zero-frequency cells across

datasets was zero. With opposite-direction thresholds, replications with zero-frequency cells also did not occur unless correlations were very high (0.7 or 0.9).

1.5 The Purpose and Overview of the Present Study

Despite these existing works, significant gaps remain, and this study aims to address them. First, there are additional strategy dimensions that prior work did not explore. For example, it did not consider adding 0.1 or adding a small value to all cells regardless of the presence of a zero cell. Nor did it compare the use of adjusted and unadjusted thresholds in the second stage of estimation after the adjusting the table. These strategies will be examined in this study.

Second, the behavior of standard errors has not yet been studied across different choices of the added constant for zero cells. Yang and Weng (2024) examined several zero-cell corrections but reported only empirical standard errors, not estimated standard errors. Therefore, we extend this work by assessing both point estimates and the properties of Wald confidence intervals across correction strategies.

Third, this study includes a sample size of 50 to fill the gap in research on smaller sample sizes, specifically those less than 100. It is particularly important because smaller sample sizes are more prone to the occurrence of zero-frequency cells, which can intensify the issues associated with these occurrences.

Finally, we extend the evaluation of these strategies from estimating a single correlation to settings with multiple variables, which are more relevant to practical applications. In practice, tetrachoric correlations are usually computed as a preliminary step to construct a full correlation matrix for further statistical analyses, such as structural equation modeling or item factor analysis. When zero-frequency cells are present and tetrachoric correlations are estimated pairwise, the resulting matrix can easily have non-positive eigenvalues (Deng, Yang, & Marcoulides, 2018; Yuan, Wu, & Bentler, 2011), which leads to difficulties for model fitting. For this reason, we evaluate correction strategies in multivariate settings, examining their impact on the entire correlation matrix, and, in a confirmatory factor analysis (CFA) application, on how such corrections affect model estimation results.

In addition to expanding the scope of the existing studies on zero-frequency cell correction, we also propose ways to improve the computational efficiency and accuracy of simulation studies for discrete problems. First, we reduce redundant estimation by exploiting the discrete nature of the problem and the symmetry of the model. Traditional simulations randomly generate a large number of datasets from a distribution and run competing statistical procedures on each of them to produce outcomes. For a discrete problem, this tends to generate the same dataset (i.e., 2×2 table) multiple times, leading to repeated calculations and computational inefficiency. Our approach exploits this discrete nature and certain symmetry of the problem and avoids unnecessary estimations, saving computational time. Second, we reduce simulation error by obtaining the theoretical sampling distribution instead of relying on a limited number of random replications (e.g., $1,000$ or $10,000$) under each condition. Further details

are provided in the Methods section of Study 1 under "Strategies for Efficient Simulation."

This paper presents three studies. The first study is a simulation comparing strategies for estimating a single correlation by varying both the value added and specific ways to add this number. The second extends the comparison to multivariate settings, and the third applies these strategies in a confirmatory factor analysis for binary data. We then synthesize the findings across the three studies and discuss their implications in the conclusion and discussion section.

2 Study 1: Bivariate Analysis

The first simulation study was conducted to evaluate various methods with different added values and different manners to add values for zero-cell correction to obtain a single correlation estimate.

2.1 Methods

Simulation Conditions In our simulation, we considered three factors: sample size (50, 100, 200), the magnitude of the underlying correlation (0.3, 0.5, 0.7, 0.9), and thresholds for each variable. The sets of thresholds were created with -1.5, -1.0, -0.8, 0.8, 1.0, and 1.5 based on Savalei (2011). We did not include the additional smaller thresholds considered by Yang and Weng (2024), since milder thresholds rarely generate zero-frequency cells. Instead, we focused on the threshold sets from Savalei to better represent scenarios where the zero-frequency problem is more likely to occur. Thus, the correlation and threshold values were consistent with Savalei (2011). We also introduced a sample size of 50 to reflect a situation where more zero cells are likely to exist. The lack of this sample size was mentioned as a limitation of Yang and Weng (2024), highlighting the need to analyze a sample size of 50 because empirical studies often have a smaller sample size.

We considered different zero-cell correction options used in practice and software. In addition to the default of not making correction, 12 different correction methods for zero-frequency cells were considered, involving two dimensions of corrections: which value to add (0.1, 0.25, 0.5) and how a value is added (keeping the marginal counts, only adding to the zero, adding to every cell when a zero cell is present, or always adding to all cells).

In the second stage of the two-stage procedure, the options of using unadjusted and adjusted thresholds were both considered whenever the zero-cell correction changes the marginal proportions. Adjusted thresholds are the thresholds calculated using the table corrected for zero-frequency cells, whereas unadjusted thresholds are the thresholds derived from the raw table without such correction.

Strategies for Efficient Simulation The purpose of a simulation study is typically to obtain the sampling distribution of an outcome of a statistical procedure. In this study, we minimized simulation error arising from the randomness

of the Monte Carlo procedure by calculating the sampling distribution directly. Note that due to the discrete nature of the problem, there are a large but limited number of different 2×2 contingency tables for each given sample size. There are 23,426 tables for the sample size of 50, there are 176,851 tables for the sample size of 100, and there are 1,373,701 tables for the sample size of 200.

Given each population correlation and threshold of a simulation condition, we calculated the theoretical probability for each contingency table to be sampled. Specifically, the theoretical probabilities for each of the four cells in the 2×2 table can first be computed based on discretizing the bivariate normal distribution, and then the probability of each contingency table can be computed from a multinomial distribution with the given cell probabilities and observed counts.

In theory, these probabilities of all 2×2 contingency tables define the theoretical sampling distribution of the observed contingency tables. Once tetrachoric correlation is estimated from each contingency table, these probabilities also define the theoretical sampling distribution of the tetrachoric correlations. For a sample size of 50, all possible tables except for those with two or more zero cells[1] were estimated to construct the sampling distribution. This involved a total of 23,128 tables. [2]

However, for larger sample sizes of 100 and 200, only tables with a probability of at least 10^{-5} in at least one condition were considered to exclude rare tables, reducing the total number of tables estimated. This resulted in a total of 15,829 tables (15,732 tables when those with two or more zeros were further excluded) for a sample size of 100 and 58,397 tables (58,394 tables when those with two or more zero cells were further excluded) for a sample size of 200.

The number of estimated tables can further be reduced by identifying prototype tables. This strategy was applied to sample sizes of 100 and 200. For two binary variables, because switching the two categories of either variable or switching the two variables results in predictable changes (e.g., a flip in sign) in the thresholds or correlation, there is no need to analyze every distinct 2×2 table. Rather, only one "prototype" table needs to be analyzed. Consider bi-

[1] We excluded tables with two zero-frequency cells, whether on the diagonal or on the same row or column, because either the two variables are perfectly related to each other and only one of them is retained in practice, or one variable is degenerated and needs to be removed in practice. The removed tables account for a small fraction in terms of both count and probability.

[2] For a fixed sample size N, a 2×2 contingency table can be represented by nonnegative integer cell counts (a, b, c, d) satisfying $a + b + c + d = N$. The number of such tables is $\binom{N+3}{3}$ (equivalently, ${}_4H_N$), which is the same as the number of ways to choose three balls from a sequence of 53 balls to determine the number of balls between and beyond the three chosen balls as the desired partition of 50. For $N = 50$, this gives ${}_4H_{50} = \binom{53}{3} = 23{,}426$ possible tables. We then excluded tables with two or more zero cells. The number of tables with exactly two zero cells is $\binom{4}{2}(N-1)$: there are $\binom{4}{2} = 6$ ways to choose which two cells are zero, and the remaining two positive counts must sum to N, which yields $N - 1$ possibilities. The number of tables with exactly three zero cells is 4 (all observations fall in a single cell). Thus, the number excluded is $6(N - 1) + 4$, which equals 298 when $N = 50$, leaving $23{,}426 - 298 = 23{,}128$ tables.

variate contingency tables with counts n_{00}, n_{01}, n_{10}, and n_{11}, where the two subscripts indicate the category labels for the first and the second variables. Every table can be turned into a prototype table that satisfies $n_{00} \geq n_{01} \geq n_{10}$ and $n_{00} \geq n_{11}$ by switching the categories or the two variables. Once the prototype table is analyzed, the parameter estimates can be modified (e.g., through sign changes or flips of the thresholds) to obtain the estimates for the other seven related tables.

We also simplified the number of conditions by reducing the number of correlations and thresholds considered in the study. Without loss of generality, we only included positive correlations, because changing the order of responses can account for negative correlations. With this approach, half of the possible correlations can be removed from the analysis. For thresholds, we only considered the situation where the first threshold is positive and no less than the absolute value of the second threshold. Consequently, with six different threshold values, the total number of threshold pairs was reduced to 12 from 36. Specifically, there are $6 \times 6 = 36$ ordered pairs. By imposing an ordering (threshold1 $\geq$ threshold2; with threshold1 > 0 to avoid symmetric duplicates), (threshold1, threshold2) and (threshold2, threshold1) are treated as the same, leaving $6 \times 5/2 = 15$ unique pairs. Excluding the three same-threshold cases leaves 12. When fully crossing these three factors, a total of 144 conditions (3 sample sizes $\times$ 4 correlations $\times$ 12 thresholds) were obtained. This is significantly smaller than the number of conditions without reduction, which would have been 864 (3 sample sizes $\times$ 8 correlations $\times$ 36 thresholds).

Computation After generating a table, we computed the correlation estimate ($\hat{\rho}$) and its standard error for each prototype table through the two-stage procedure using our modified version of the function `polychor` in the `polycor` package (Fox, 2022). Moreover, when a non-excluded table contained a zero cell, we did not estimate the uncorrected correlation; instead, we set it to 1 or -1 (depending on the location of the zero cell), because these are the theoretical boundary values that maximize the likelihood function at the second stage (see Figure 1). The use of iterative estimation for such tables typically results in a value close to 1 or -1 that depends on the optimization package used. These approaches were also used in the later simulations in Study 2 and Study 3.

The loss function minimized in the second stage is defined as L below. In the formula, N is the total sample size, n_{ij} is the count in nonzero cells where i and j correspond to response categories for the first and second variables, respectively, and p_{ij} is the expected proportion of each cell:

$$L = -\frac{1}{N} \sum_{j=0}^{1} \sum_{i=0}^{1} n_{ij} \cdot \log \left(\frac{p_{ij}}{n_{ij}/N} \right). \tag{1}$$

We used the Nelder–Mead algorithm in the `optim` function in R to estimate $\hat{\rho}$ by minimizing the loss function L, which is defined on $[-1, 1]$ but set as undefined beyond -1 and 1. The R code used in our computations is available in the OSF repository: `https://osf.io/hmk2e/`.

Evaluation Criteria We used three different criteria to evaluate point estimates of a single tetrachoric correlation: the root mean square error (RMSE) and the mean absolute error (MAE) for correlation estimates, and the MAE of Fisher's z-transformed correlation estimates. These measures were calculated using the probabilities of the tables being sampled as weights.

Fisher's z-transformation is particularly relevant for the boundary-solution problem, because it amplifies the penalty for near-perfect correlations. This makes Fisher's z an especially important complement to RMSE and MAE, which evaluate errors uniformly across the correlation range without giving extra weight to boundary cases. Fisher's z-transformation for the correlation is defined as follows.

$$z = \frac{1}{2} \ln \left(\frac{1+\rho}{1-\rho} \right) \tag{2}$$

After applying this transformation, the transformed values were used in place of the raw correlation coefficient $(\hat{\rho}, \rho)$ when computing error. Fisher's z-transformation cannot be used with a perfect correlation, so it was not applied when no modification was used for zero counts.

The estimated standard error (SE) is typically used to form a Wald confidence interval (CI) as an interval estimate, so to evaluate the SE we computed the noncoverage rates of the 95% Wald CI of the correlation. Because when a zero-frequency cell is present but no correction is made, the estimate must be 1 or -1 and the SE cannot be properly estimated, the noncoverage rates were only calculated when a nonzero added value was used.[3]

2.2 Results

In this section we present results for the sample size of 50. The sample sizes of 100 and 200 led to similar patterns in the results, and their difference from the sample size of 50, if present, will be noted below in footnotes. The tables for all sample sizes and figures for the two larger sample sizes are provided in the Appendix. In the figures, rows are ordered by increasing correlation size from top to bottom, while columns are ordered by the distance between thresholds, with thresholds becoming closer from left to right. In the far-right three columns with identical thresholds, the order is based on the size of the threshold, with the more extreme threshold on the left.

The Number of Zero-Frequency Cells Table A1 in the Appendix presents the composition of the sampling distribution of 2×2 tables based on the number of zero-frequency cells. This includes scenarios with no zero-frequency cells, one zero-frequency cell, and two or three zero-frequency cells. The sampling distribution primarily consists of tables with no zero-frequency cell or just one, while

[3] When the population correlation is 1 (or -1), there must be at least one zero cell in the contingency table and the estimate must be 1 (or -1), so the theoretical SE is zero.

probabilities of two or more zero-frequency cells are relatively rare, and such tables were removed from the analysis. Specifically, in 52.08% of conditions (25 out of 48), the probability of the presence of exactly one zero is greater than 0.5, while only in 8.33% of conditions (4 out of 48), this probability is less than 0.05.[4] Zero-frequency cells appear more often in cases with high correlations or extreme thresholds. For example, high correlations like 0.7 or 0.9 combined with thresholds of opposite signs lead to higher probabilities of one zero cell. These results indicate our choice of simulation conditions has properly captured the scenarios with zero-frequencies.

Root Mean Square Error (RMSE) The RMSE values of the correlation estimates were calculated for each combination of threshold sets, correlation sizes, and different modifications for zero-frequencies. Modifications included which value to add, how to add it, and whether to use adjusted or unadjusted thresholds in the second stage of the two-stage procedure. Figure 2 presents the RMSE for point estimates of the correlation when the sample size is 50.

The overall pattern in Figure 2 indicates that the added value plays the primary role in determining the results, and that, given the optimal added value, the manner of addition has only a minor influence. The choice of unadjusted or adjusted thresholds makes little difference. For the added value, there is a general trend that as thresholds become far apart and correlation becomes greater, adding a smaller number tends to produce the best result. In particular, when thresholds have the same sign (i.e., the right block of Figure 2), the use of 0.5 as the added value appears to be optimal in all panels, with possible exceptions for the highest correlation and most distant thresholds, for which smaller added values may be optimal. For opposite-signed thresholds and the highest correlation (i.e., last row of the left block), not adding a number to the zero cells produces the lowest RMSE. In the remaining panels of this figure (i.e., first three rows of the left block), the optimal added value increases from 0 to 0.5 from the lower left to the upper right.

The mean absolute errors (MAE) show a similar pattern to RMSE. The relevant figures can be found in the Appendix (Figures A1, A4, A5).

Mean Absolute Error of Fisher's Z-Transformed Correlation Figure 3 presents the MAE of Fisher's z-transformed correlation estimates for a sample size of 50. It shows that the MAE for Fisher's z-transformed correlation estimates exhibits a very similar pattern, but no modification is no longer an option because it would produce an estimate of 1 or -1 and an infinite transformed value. Although the added value drives the general pattern, under some conditions the

[4] For a sample size of 100, 45.8% (22 out of 48 conditions) have a probability of exactly one zero greater than 0.5, while in 22.9% (11 out of 48) this probability falls below 0.05. For a sample size of 200, in 37.5% (18 out of 48) conditions this probability exceeds 0.5, and in 47.9% (23 out of 48) conditions it falls below 0.05 (see Tables A2 and A3 in the Appendix).

specific way of adding the value leads to notable differences within the same added value. For instance, for correlations of 0.3 and 0.5, and thresholds of 1.5 and 1, keeping the marginals and adding to all cells regardless result in much lower MAE within an added value of 0.5 compared to other methods. When thresholds have opposite signs, the optimal added value decreases from 0.5 to 0.1 as the thresholds become farther apart and the correlation becomes greater. With the optimal added value, the way to add does not matter much. When thresholds are of the same sign, in most cases (with the possible exception of a correlation of 0.9 combined with thresholds 1.5 and 0.8), adding 0.5 produces the smallest MAE. With this optimal added value, either adding it consistently to all cells even when no zero is present or adding it while keeping the marginals is among the best ways to add.

Noncoverage Rates The noncoverage rates for the 95% Wald CI are provided in Figure 4. We compared different correction methods against the nominal level of 0.05. When there are zero cells in the table, standard errors cannot be computed without zero-cell correction due to the boundary estimate of ± 1. Therefore, coverage rates were not calculated for no correction. Because adjusted and unadjusted thresholds produced very similar point estimates, only adjusted thresholds were considered in the calculation of SE and CI.

In scenarios with positive thresholds, adding 0.5 leads to the lowest noncoverage rates in almost all conditions. Specifically, when thresholds are not extreme, adding 0.5 to all cells regardless of the presence of zero cells tends to result in the lowest noncoverage rates, which are also the rates closest to the nominal level of 0.05. In cases with at least one extreme threshold of 1.5, adding to zero cells while maintaining the marginals tends to give the lowest noncoverage rates, which are also the rates closest to the nominal level for small and moderate correlations; for higher correlations, a smaller added value with marginals maintained may lead to the closest noncoverage to the nominal value of 0.05.

When thresholds have mixed signs, adding different values generally produces very similar results, which are mostly below 0.05, except for some combinations of high correlation or extreme thresholds. For small correlations and closer thresholds, adding a larger value (0.5) is a better strategy compared to adding smaller values. However, when a high correlation or two distant thresholds are present, a smaller added value such as 0.1 or 0.25 becomes the better strategy.

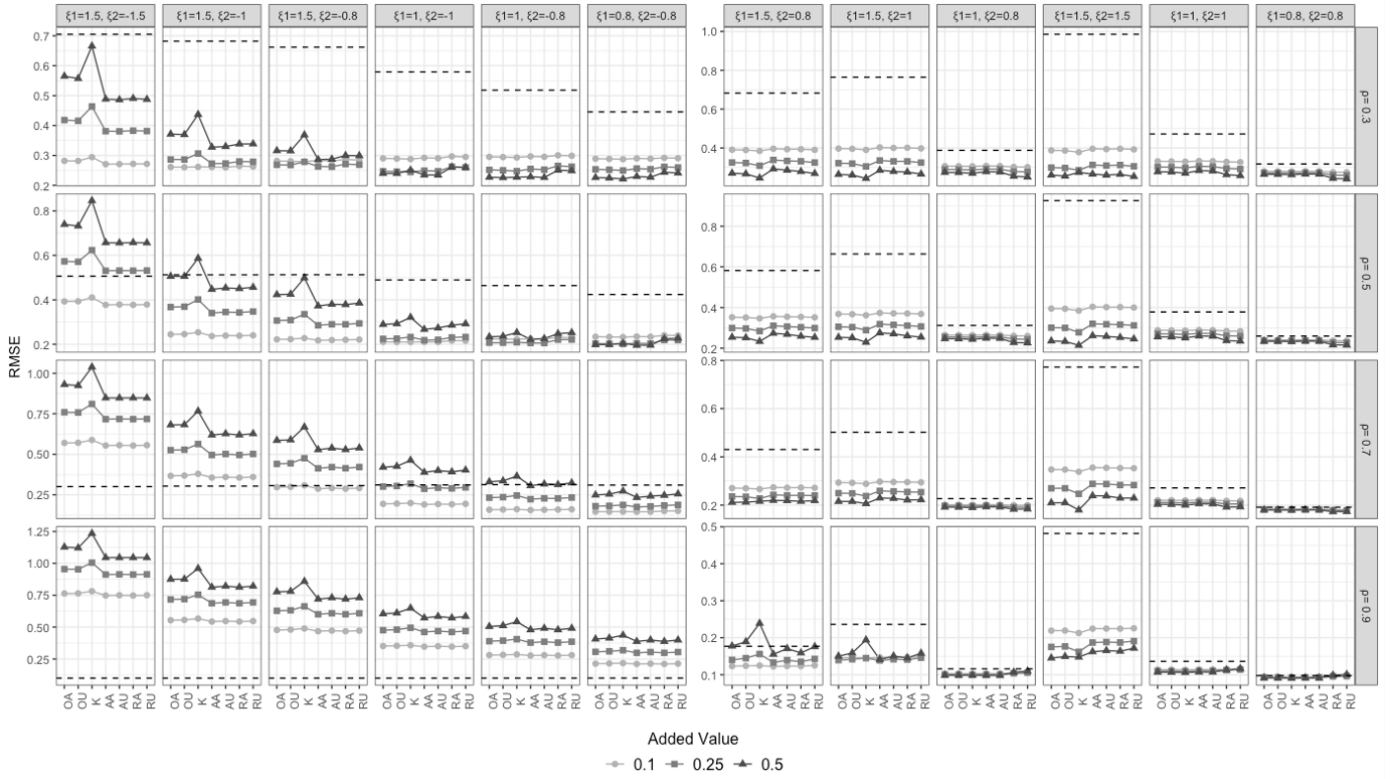


Figure 2: Root Mean Square Error (RMSE) for point estimates of the correlation ($N = 50$).

Note. Abbreviations for modifications: OA/OU = Only add to the zero cell and use adjusted/unadjusted thresholds in the second stage of estimation; K = Keep the marginal, for which the thresholds stay the same; AA/AU = Add to all cells when a zero is present in the table, and use adjusted/unadjusted thresholds in the second stage; RA/RU = Add to all cells regardless of the presence of zero, and use adjusted/unadjusted thresholds in the second stage. The dotted horizontal bar in each panel represents no correction. Y-scales vary with correlation sizes and threshold signs to better show differences within each panel.

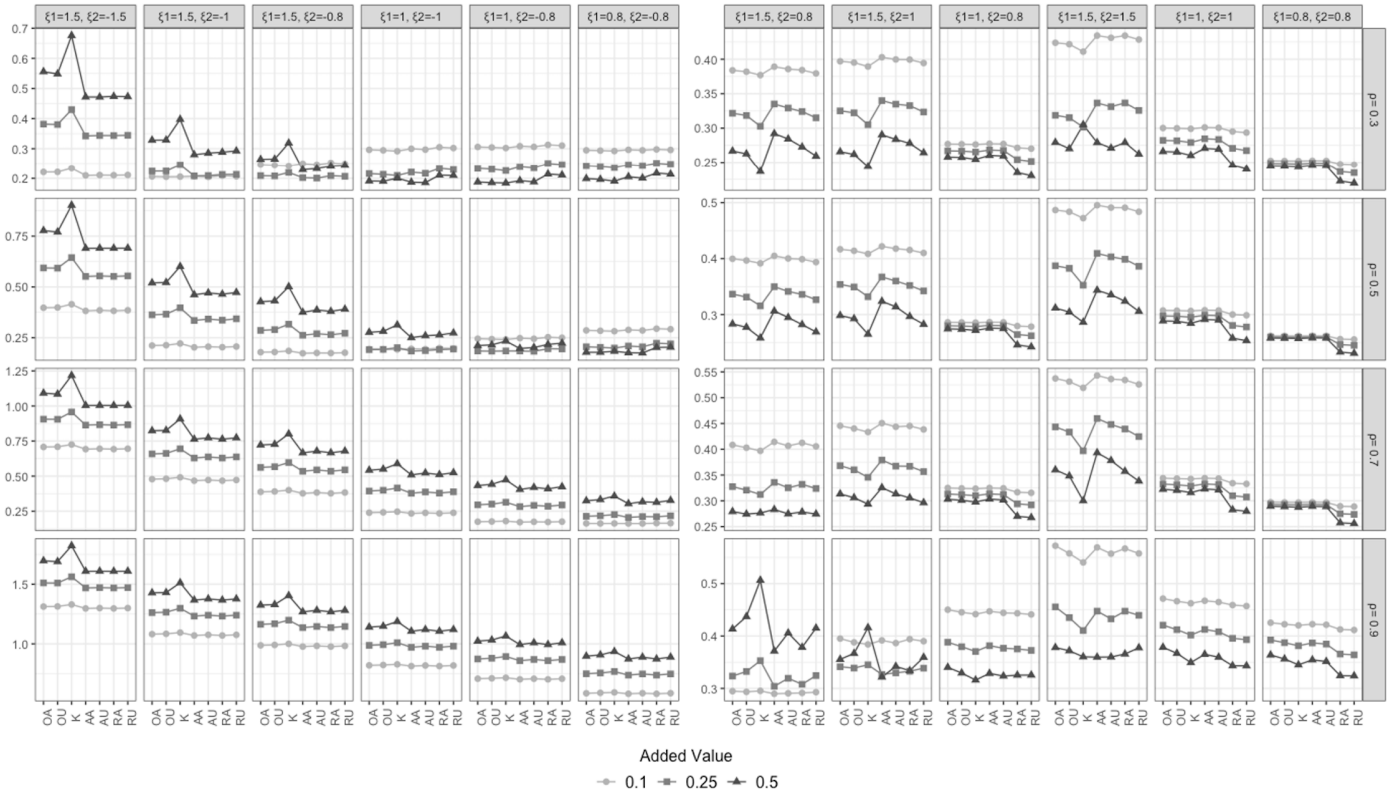


Figure 3: Mean Absolute Error (MAE) for point estimates of the correlation after Fisher's z-transformation ($N = 50$).

Note. The structure of this figure is the same as Figure 2. However, in this evaluation, no correction (i.e., added value 0) was not included.

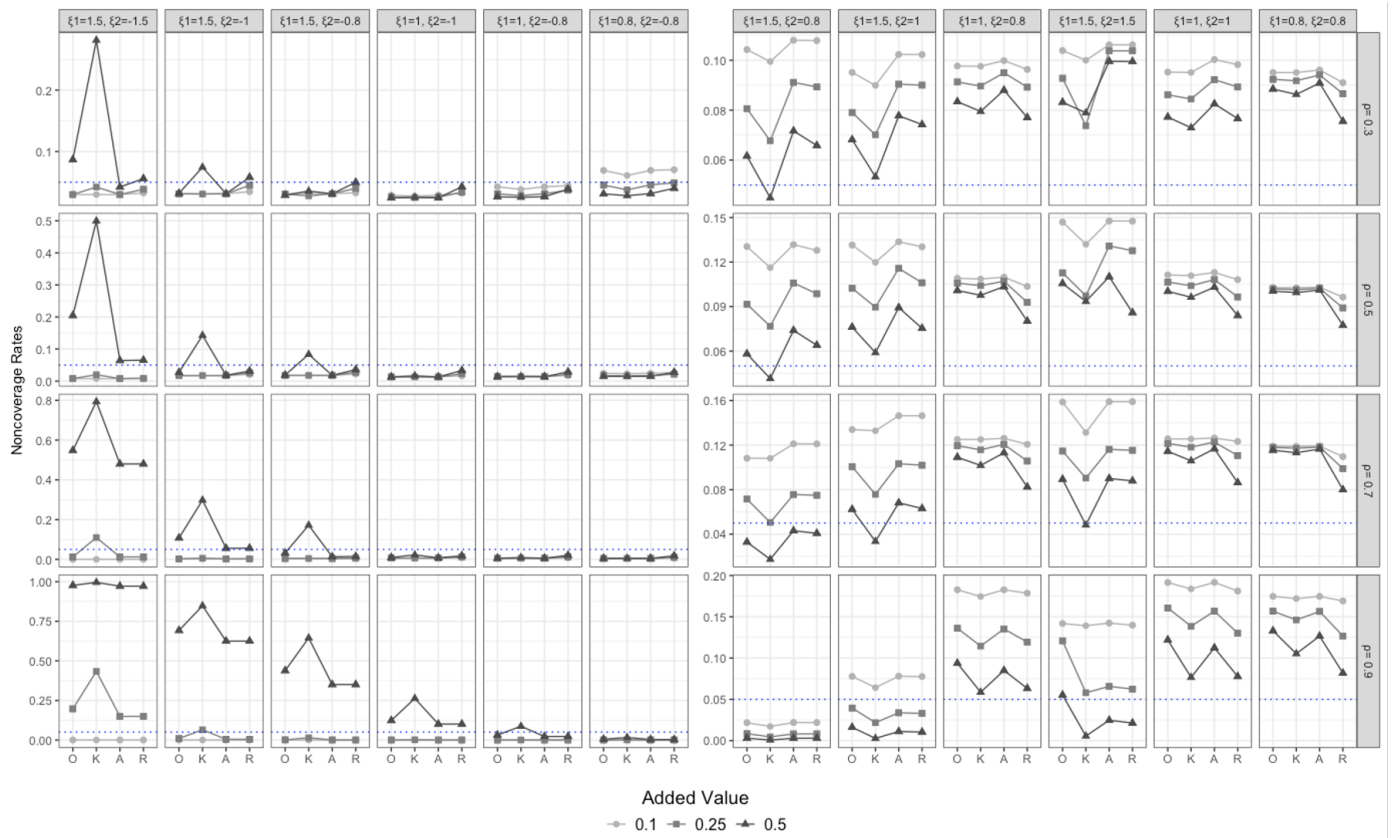


Figure 4: Noncoverage rate of the 95% Wald confidence interval of the correlation ($N = 50$).

Note. Abbreviations for modifications: O = Only add to the zero cell; K = Keep the marginal; A = Add to all cells when zero is present in the table; R = Add to all cells regardless of the presence of zero. The dotted horizontal bar in each panel indicates 0.05. Y-scales vary with correlation sizes and threshold signs.

3 Study 2: Multivariate Analysis

Given the common practice of using tetrachoric correlations in modeling with multiple variables rather than a single correlation, the second study was conducted to assess the different strategies' performance in estimating a correlation matrix.

3.1 Methods

Simulation Conditions The second simulation study used six binary variables. Data were first generated from a 6-variate normal distribution, and then discretized using thresholds. The sample size was set at 50, which produced the most notable results in the bivariate analysis.

To simplify the four correlations used in the bivariate simulation (0.3, 0.5, 0.7, and 0.9), we selected two representative values: 0.4 as the midpoint of 0.3 and 0.5 to represent moderate correlations, and 0.8 as the midpoint of 0.7 and 0.9 to represent high correlations. Based on these values, three correlation structures were considered: uniformly 0.4, uniformly 0.8, or mixed with two 3×3 diagonal blocks of 0.8 and an off-diagonal block of 0.4. Thresholds were also simplified using values from the bivariate simulation. Two sets of thresholds were considered: positively signed $(1.5, 1.0, 0.8, 1.5, 1.0, 0.8)$ and mixed signed $(-1.5, -1.0, -0.8, 1.5, 1.0, 0.8)$.

Crossing three sets of correlations and two sets of thresholds resulted in six generation conditions, each replicated 30,000 times. After the data generation, contingency tables were formed for each pair of variables, which were then used for estimating correlations.

Efficient Estimation of Correlations We evaluated case of no correction alongside 12 correction strategies, consistent with the bivariate simulation. However, we exclusively used adjusted thresholds in the second stage of the estimation, as their impact was found to be minimal in the bivariate simulation. Each of these correction strategies was applied to every contingency table for each pair of variables.

Unlike a traditional simulation where each replication from each simulation condition is analyzed, we only needed to estimate distinct tables among all replications across all conditions. Especially, given results from Study 1, the estimate of each distinct table was directly matched from our estimated results in Study 1 with a sample size of 50, to avoid redundant estimation and reduce computational cost. Then correlation matrices were constructed using these matched correlations. Because we had six variables in the analysis, we had a sample 6×6 correlation matrix including 15 different correlation estimates $\big(6 \times 5/2\big)$, along with diagonal elements being 1.

Evaluation Criteria We evaluated different strategies for handling zero cells based on two evaluation criteria: the percentage of positive definite correlation

matrices and the accuracy of estimation as measured by the average weighted squared error loss (a quadratic form loss).

First, the number of replications that yielded positive definite tetrachoric correlation matrices was counted. A positive definite matrix is necessary for many analyses, but having one does not mean the estimates are accurate. For example, adding larger constants to zero cells increases the chance of positive definiteness, but this can also move the estimates further away from the true matrix.

Second, we measured estimation error using the quadratic form loss. Let the sample correlation matrix be $\mathbf{R}$ and the population (true) correlation matrix be $\mathbf{P}$, then the quadratic form loss can be calculated as follows:

$$\mathrm{tr}\{(\mathbf{R}-\mathbf{P})\mathbf{P}^{-1}(\mathbf{R}-\mathbf{P})\mathbf{P}^{-1}\}. \tag{3}$$

Note that the matrix $\mathbf{P}$ is always positive definite, but the estimated matrix $\mathbf{R}$ may or may not be so. This criterion is the multivariate analogue of mean squared error. It can be seen as a variant of Stein's loss for covariance matrices a second-order Taylor expansion of Stein's loss around $\mathbf{P}$ yields the expression above. Unlike Stein's loss, however, it only requires $\mathbf{P}$ to be positive definite, not $\mathbf{R}$, which is useful here because pairwise tetrachoric estimates can produce a nonpositive definite $\mathbf{R}$.

3.2 Results

The Number of Zero-Frequency Cells We first conducted an analysis on the composition of generated tables across 30,000 replications. Each replication was examined for the presence of zero-frequency cells within the 15 pairwise contingency tables. Replications were classified based on the zero cells in these tables: if none of the 15 tables had any zero cells, the replication was categorized as "Without Zero-Frequency Cells." If there was at least one table with one zero cell, but none with two zero cells, it was categorized as "With a Zero-Frequency Cell." Finally, if at least one table included two or more zero cells, it was classified under "With Two or More Zero Cells (Removed)," and such replications were excluded from the analysis.

Table A4 in the Appendix shows that conditions with mixed correlations tend to have more instances of single zero-frequency cell than those with correlations consistently set at 0.4 or 0.8. With mixed thresholds, more replications within the condition include at least one table with zero-frequency cells, regardless of the correlation size, compared to positive thresholds. For each condition, the vast majority of the replications contain at least one table with a single zero-frequency cell. This suggests that our choice of simulation conditions is suitable for the evaluation of strategies to handle zero-frequency cells.

Positive Definiteness Results on positive definiteness are provided in Table A5 in the Appendix and Figure 5. The number of positive-definite matrices varies from 0 (0%) to 25,534 (85.11%), depending on the generation condition

and the correction method used. In this multivariate simulation, the added value also plays a major influence on the results. Positive definiteness is rare when no correction is performed on zero-frequency cells. More positive-definite matrices are observed in mixed threshold conditions compared to positive threshold conditions. Additionally, uniformly 0.4 and 0.8 correlations tend to produce more positive-definite matrices compared to mixed correlation conditions. Across all conditions, adding larger values tends to increase the count of positive-definite matrices. Specifically, adding an optimal value of 0.5 to the zero cell while keeping the marginal results in the highest count of positive-definite matrices.

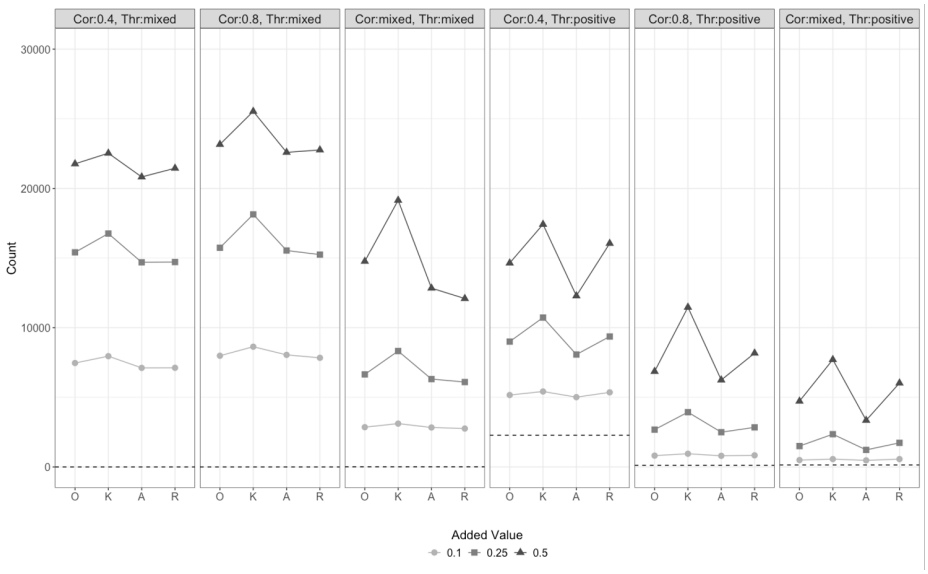


Figure 5: Number of positive definite matrices

Note. Abbreviations for modifications: O = Only add to the zero cell; K = Keep the marginal; A = Add to all cells when zero is present in the table; R = Add to all cells regardless of the presence of zero. The dotted horizontal bar in each panel represents no correction.

As the results suggest, adding 0.5 can make it more likely that the correlation matrix is positive definite, but a correction that achieves this by adding a larger constant may lead to estimates that deviate substantially from the population values. Thus, positive definiteness alone is not an adequate criterion for evaluating correction methods. It is also necessary to consider measures of deviation, such as quadratic form loss, which we introduce in the next section to assess how far the corrected estimates are from the true correlation structure.

Quadratic Form Loss Results regarding the quadratic form loss are shown in Table A6 in the Appendix and Figure 6. These results show that correlations being consistently 0.4 lead to smaller deviations compared to the other correlation conditions.

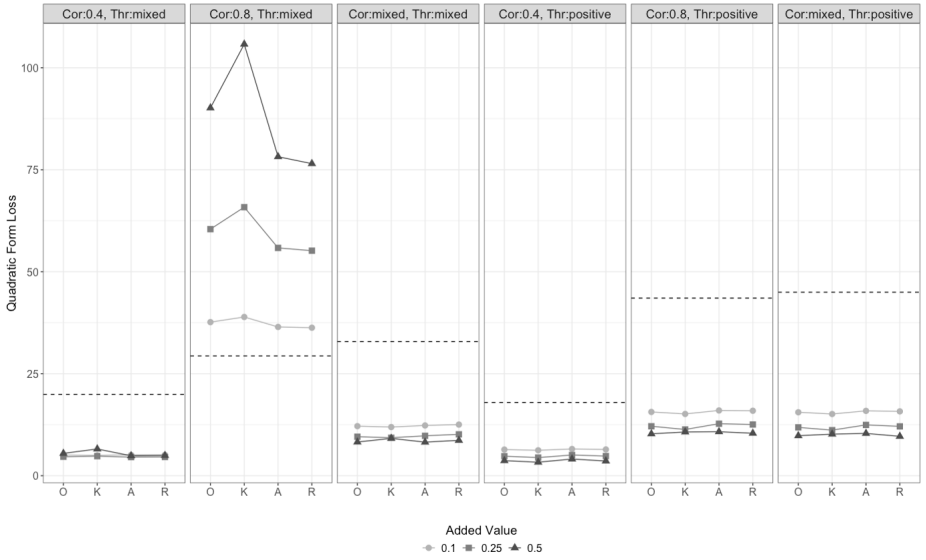


Figure 6: Quadratic form loss.

Note. This figure follows the same structure as Figure 5.

When thresholds are all positive, the quadratic form loss generally decreases as larger values are added, and not correcting for the zero cells results in notably higher deviations. With the optimal added value of 0.5, the manner in which it is added makes minimal differences.

When thresholds have mixed signs, the pattern differs by the size of correlations: when the correlations are 0.4, the quadratic form loss is lowest with the added value of 0.25; when the correlations are mixed between 0.4 and 0.8, adding 0.5 is the optimal added value; when the correlations are 0.8, the lowest loss occurs without zero-cell correction. In this third condition, adding larger values leads to higher deviations and the deviations are larger than those in the other panels. This is likely due to occasional sign flips with the correction: when the estimated correlation becomes negative, the distance from the true value (0.8) increases substantially, resulting in larger deviations. This will be discussed in further detail, with illustrative examples, in the Conclusion and Discussion section.

Implications of Multivariate Results and Their Relation to Bivariate Patterns Taken together, the positive definiteness and quadratic form loss results indicate that larger additions increase the likelihood of positive definiteness but move estimates farther from the true matrix. Thus, positive definiteness alone is therefore not an adequate criterion and it should be considered alongside accuracy measures such as the quadratic form loss. In our simulations, the strategy that produced the most positive definite matrices was not always the one that minimized quadratic form loss: larger corrections tended to favor positive definiteness, whereas smaller corrections often performed better when correlations were high and thresholds had mixed signs. This suggests that both criteria need to be considered together when evaluating correction strategies.

Overall, the multivariate results are generally consistent with the bivariate results. In both settings, the choice of added value primarily determines performance, while the manner of addition has little impact once the value chosen well. As in the bivariate patterns, higher correlations tend to favor smaller added values. With all-positive thresholds, adding 0.5 generally performs best, and leaving zero cells uncorrected performs poorly. With mixed-sign thresholds, at high correlation (e.g., 0.9 in the bivariate case or 0.8 in the multivariate case), no correction is preferable at high correlations (about 0.9 in the bivariate case and 0.8 in the multivariate case), whereas at lower correlations (around 0.4 in the multivariate case) smaller additions such as 0.1 or 0.25 work better.

4 Study 3: Confirmatory Factor Analysis

Since tetrachoric correlations are often used in fitting factor analysis models, in this simulation, different correction strategies for treating zero cells were evaluated according to their performance in estimating a confirmatory factor analysis (CFA) model.

4.1 Methods

Simulation Design This simulation involved four binary variables within a single-factor model. Data were generated from a 4-variate normal distribution with a sample size of 50 and then discretized using specific thresholds. Population loadings were consistently set at either 0.4 or 0.7. The threshold conditions for the four variables included either all positive thresholds or mixed-signed thresholds, with two positive and two negative values, and absolute values of 1.5, 1.0, and 0.8. This resulted in six distinct threshold conditions: all thresholds set to 1.5, all thresholds set to 1.0, all thresholds set to 0.8, mixed thresholds of 1.5 and -1.5, mixed thresholds of 1.0 and -1.0, and mixed thresholds of 0.8 and -0.8. By crossing the two loadings with the six sets of thresholds, twelve generation conditions were obtained, each replicated 1,000 times.

Computation With the generated data, contingency tables were created for each pair of variables. Consistent with Study 2, correlation and standard error

estimates for these tables were matched from the bivariate simulation results, and a 4×4 tetrachoric correlation matrix was produced for each replication. The resultant tetrachoric correlation matrix was further used to estimate the one-factor model with four variables through diagonally weighted least squares (DWLS). Specifically, we minimized the sum of squared differences between the estimated tetrachoric correlations and their model-implied values, with each difference weighted by the inverse of the estimated SE of the tetrachoric correlation. In the factor model, the factor variance was fixed at 1, the unique variances were constrained so that the latent continuous responses would have unit variances, and the factor loadings were constrained to lie between -1 and 1 to avoid Heywood cases. The `cfa` function from the `lavaan` package (Rosseel, 2012) was used.

Evaluation Criteria To evaluate the estimated loadings, we compared the mean of the four estimated loadings to the population loading. RMSE was used as the evaluation criterion and was computed after aligning the estimates. Before the estimated loadings could be properly evaluated, they were aligned across replications to address the potential sign indeterminacy in factor analysis. Note that in this CFA model all loadings can take the reversed signs to produce a statistically equivalent solution. To align the solutions across replications, for each replication, we computed the sum of squared differences between the estimated and true loadings, as well as for the true loadings with flipped signs. If the sum of squared differences was smaller for the sign-flipped true loadings, we flipped the signs of all estimated loadings for that replication.

4.2 Results

Figure 7 and Table A7 in the Appendix also show that the size of the RMSE is influenced by the added value. For all positive thresholds, adding larger values generally results in lower RMSE. Specifically, with the largest added value of 0.5, either keeping the marginals or adding to all cells regardless of zero cells often leads to the lowest RMSE. For mixed-sign thresholds, adding smaller values generally results in lower RMSE. With extreme thresholds of 1.5 regardless of the size of loadings, and thresholds of 1.0 with a loading of 0.7, the smallest added value of 0.1 results in the lowest RMSE. For thresholds of 1.0 with a loading of 0.4 and thresholds of 0.8, adding a medium (0.25) or large (0.5) value produces the lowest RMSE.

Relation to Bivariate Patterns These results are also generally consistent with the bivariate results. When thresholds are all positive, both show that larger added values lead to better performance, whether evaluated by RMSE of estimates or mean loadings. For mixed-sign thresholds, smaller added values work better in most cases. In less extreme situations, such as with smaller loadings or less extreme thresholds, added values like 0.25 or 0.5 give better results. These patterns are consistent with the bivariate results.

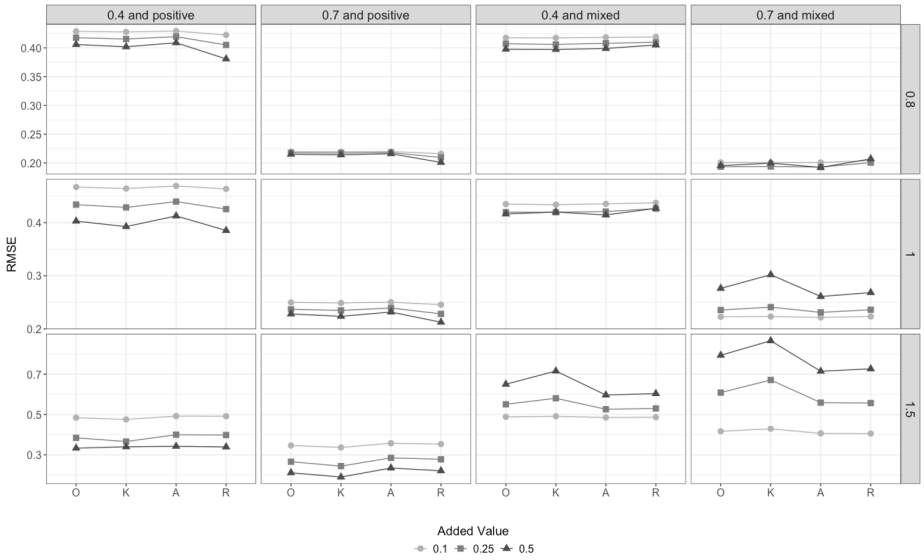


Figure 7: Root mean square error (RMSE) in the mean of loadings in the CFA simulation.

Note. This figure follows the same structure as Figure 6.

5 Conclusion & Discussion

5.1 Summary and Discussion of Results

This research addresses an important gap in the literature on tetrachoric correlation estimation by evaluating different strategies for treating zero-frequency cells. Practical approaches to zero-frequency cells are crucial for researchers estimating tetrachoric correlations. Despite available correction methods in statistical software for tetrachoric correlations, there are limited systematic evaluations on this issue, especially in multivariate contexts. Our simulation study makes a comprehensive effort to explore various correction methods of zero-frequency cell correction.

In the bivariate analysis, the choice of the added value had the largest impact on performance measures; for the optimal added value, the four different strategies made much less difference and the use of adjusted or unadjusted thresholds following correction had minimal effects. This optimal added value tended to be greater for a smaller correlation and similarly located thresholds but smaller for a more extreme correlation and more distantly located thresholds. Specifically, for most conditions with same-signed thresholds, adding a larger number, 0.5, while keeping the marginal produced the best results. For opposite-signed thresholds, the optimal added value remained at 0.5 for a small correlation and less distant thresholds but it decreased as the correlation and thresholds became more extreme; the best strategy became making no correction (or adding 0.1 if making

no correction was not feasible for the evaluation criterion) for a correlation of 0.9. Table 1 summarizes the main patterns in the bivariate analysis.

Table 1: Summary of optimal added-value patterns in bivariate simulation

Scenario	Pattern	Best added value
Overall (across conditions)	Choice of added value had the largest impact on performance. When the added value was near-optimal, differences among strategies were small, and using adjusted versus unadjusted thresholds after correction had minimal effects.	–
Larger optimal added value	More likely with smaller correlations and thresholds that are closer (similarly located).	Higher
Smaller optimal added value	More likely with more extreme correlations and thresholds that are farther apart (more distantly located).	Lower
Same-signed thresholds (most conditions)	Adding a larger number while keeping marginals fixed produced the best results.	0.5
Opposite-signed thresholds (small correlation, less distant thresholds)	Optimal added value remained high.	0.5
Opposite-signed thresholds (more extreme correlation and/or more distant thresholds)	Optimal added value decreased as correlation and thresholds became more extreme.	Decreasing trend
Opposite-signed thresholds (very high correlation)	Best became making no correction; if not feasible for the evaluation criterion (fisher's z), use a minimal correction.	0 (or 0.1 if needed)

Smaller optimal added values under extreme thresholds and/or large correlations likely reflect how sensitive tetrachoric estimates are to small corrections when zero cells occur. In these settings, adding a value to a zero cell can noticeably shift the estimate, and in some cases even reverse its sign. Figure 8 illustrates two examples of this sign-flip behavior. In the top panel of Figure 8, under a simulation condition with population correlation 0.3 and thresholds ± 1.5, an example table with observed counts $(47, 2, 1, 0)$ contains a zero cell because one theoretical cell probability is extremely small. For this table, the MLE without correction is 1, although software may report a value close to 1 due to imper-

fect convergence. When 0.5 is added only to the zero cell, the estimate becomes -0.5667. This example shows that, with opposite-signed thresholds, a correction can substantially change the estimate and even flip its sign, which explains why the optimal added value tends to be smaller in more extreme settings.

Negative-correlation cases were not included in our simulation condition because they can be re-expressed within our design by flipping the threshold sign pattern: a negative correlation with same-signed thresholds corresponds to a positive correlation with mixed-signed thresholds, and a negative correlation with mixed-signed thresholds corresponds to a positive correlation with same-signed thresholds. For illustration, the bottom panel of Figure 8 shows the corresponding case with both thresholds negative and demonstrates that the sign-flip behavior can also occur in the opposite direction: in the top panel the correction induces a sign flip, whereas in the bottom panel the uncorrected estimate flips sign and the correction yields an estimate with the same sign as the population value.

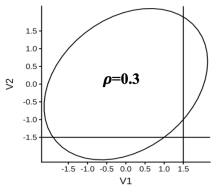
Given that a simulation study can only consider a limited number of added values, it is highly likely that the actual optimal added value lies between 0 and 0.1 in this situation. Our findings show that in situations where adding a number is advantageous, such as with opposite-signed thresholds found in Savalei (2011) and Yang and Weng (2024), the optimal value to add can vary. This adds more detailed information compared to the previous studies.

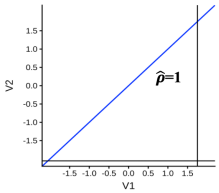
In multivariate simulation, the combination of correlations and thresholds led to different outcomes. Adding a larger number often yielded better results, while not correcting for the zero cell consistently resulted in poor results. Within these added values, keeping the marginals was generally the most effective strategy for achieving positive definiteness. However, for high correlations with mixed-signed thresholds, not correcting for zero cells or adding a smaller value often yielded better results in terms of the quadratic form loss. The results show that methods improving positive definiteness do not always give the most accurate estimates, so both aspects should be considered when choosing a correction strategy.

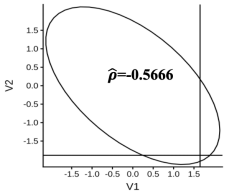
In CFA simulation, the optimal added value depended on correlation and threshold sizes. Smaller added values produced better results with extreme mixed-signed thresholds, while larger values were more effective for positive thresholds.

Although no single approach performed best across all studies, several consistent patterns emerged across the bivariate, multivariate, and CFA scenarios. First, when thresholds are of the same sign, adding 0.5 tends to work well across settings and can be a reasonable choice in many situations. Second, when thresholds have opposite signs, smaller additions such as 0.1 or 0.25 may work better, while in rare cases of very high correlations leaving the zero cells uncorrected can perform similarly. Third, the specific manner the value is added appears less important than the size of the addition itself. Finally, the effect of correction should be evaluated from multiple perspectives: larger corrections increase the likelihood of obtaining a positive definite matrix but do not necessarily improve estimation accuracy, as positive definiteness becomes more likely as a consequence of larger corrections.

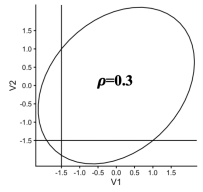
True ρ						$\widehat{\rho}$ (No Modification)						$\widehat{\rho}$ (With Modification)					
Probabilities			Expected Table			Probabilities			Observed Table			Probabilities			Modified Table		
	0	1		0	1		0	1		0	1		0	1		0	1
1	0.867	0.066	1	43.35	3.30	1	0.94	0.04	1	47	2	1	0.93	0.04	1	47	2
0	0.066	0.001	0	3.30	0.05	0	0.02	0	0	1	0	0	0.02	0.01	0	1	0.5

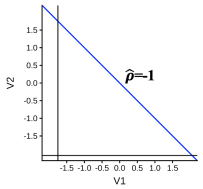






True ρ						$\widehat{\rho}$ (No Modification)						$\widehat{\rho}$ (With Modification)					
Probabilities			Expected Table			Probabilities			Observed Table			Probabilities			Modified Table		
	0	1		0	1		0	1		0	1		0	1		0	1
1	0.06	0.88	1	2.77	43.89	1	0.04	0.94	1	2	47	1	0.04	0.93	1	2	47
0	0.01	0.06	0	0.57	2.77	0	0.00	0.02	0	0	1	0	0.01	0.02	0	0.5	1





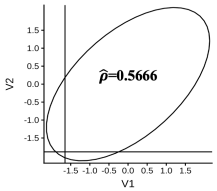


Figure 8: Examples of sign flips in the estimated tetrachoric correlation with and without correction

Note. Each panel shows the bivariate normal cell probabilities computed with the corresponding correlation and threshold values, together with an example 2×2 table and the resulting tetrachoric correlation estimates. For the estimate without correction (with a zero cell), $\widehat{\rho} = \pm 1$; however, different software packages may give slightly different results due to lack of perfect convergence.

We would like to note that our simulation focused on the estimation of tetrachoric correlations, whereas polychoric correlations involve more than two categories and can behave differently as discussed in the Introduction. Accordingly, our findings may not generalize to polychoric settings, consistent with Savalei's (2011) conclusions.

5.2 Contributions

The current study makes several contributions to the literature on estimating tetrachoric correlations in the presence of zero-frequency cells. Firstly, we provided a more comprehensive understanding of the zero-cell corrections in the tetrachoric correlation context. Prior work, such as Savalei (2011), primarily compared adding 0.5 to not adding it. Our study included more correction methods and revealed a potential optimal value to add between 0 and 0.5. We revealed unexplored distinctions in different ways of adding the number, including the issue of using adjusted and unadjusted thresholds in the two-stage procedure.

Secondly, these different correction methods were examined under more realistic data conditions and practical scenarios. Specifically, we included a sample size of 50 to better depict smaller sample situations where zero-frequency cells are more prevalent. Additionally, we considered multivariate and confirmatory factor analysis scenarios, reflecting the practical use of tetrachoric correlations in constructing correlation matrices for various analyses. In this way, we illustrate how different correction strategies affect not only individual correlation estimates but entire models.

Thirdly, this study provided a more efficient way of conducting simulation studies so that researchers can significantly reduce the number of estimations. Tetrachoric correlations are known to be computationally intensive (Zhang et al., 2022). To alleviate this problem, we took advantage of 2×2 discrete nature of tetrachoric correlations. We did this by simplifying the sets of the simulation condition and reducing the number of estimations by setting the prototype of similar tables. As this computation issue can get worse in simulation studies that involve many replications, our approach to make the simulation efficient can serve as a good example of simulations with tetrachoric correlations. It also has potential to be extended as a more general framework for discrete data simulations in broader contexts.

Lastly, we reduced simulation error by deriving the sampling distribution from all possible tables under each condition, not relying on a limited number of replications. This framework avoids error introduced by random sampling and produces more precise simulation results. The influence of sampling error is especially evident in discrete data problems such as 2×2 tables, where the limited number of replications can exaggerate variability. Therefore, this approach is useful for generating more accurate and reliable simulation results and helps to overcome fundamental limitations of simulation studies involving discrete data.

5.3 Limitations and Future Directions

Despite these contributions, there is a need for additional exploration in future research. Firstly, our analysis is limited in assessing potentially effective ways to correct zero cells due to simulation design constraints. Future research could consider a wider range of values such as those smaller than 0.1. For multivariate simulations, applying different correction methods to each pair of variables could potentially provide useful insights, while our study applied the same correction across the variables to see the general trend. Secondly, exploring varied correction methods for each variable in multivariate simulations and investigating alternative approaches, such as collapsing categories (DiStefano, Shi, & Morgan, 2021) instead of adding a small number, are also possible directions. We anticipate future research to investigate more meticulous approaches for addressing zero-frequency cells.

Author Notes

Parts of the results of this paper were presented at the 2024 Annual Meeting of the Society of Multivariate Experimental Psychology (Choi & Wu, 2025).

References

Brown, M. B., & Benedetti, J. K. (1977). On the mean and variance of the tetrachoric correlation coefficient. *Psychometrika*, *42*(3), 347–355. doi: https://doi.org/10.1007/bf02293655

Choi, J., & Wu, H. (2025). On zero-count correction strategies in tetrachoric correlation estimation (abstract). *Multivariate Behavioral Research*, *60*(1), 3–4. doi: https://doi.org/10.1080/00273171.2024.2442249

Deng, L., Yang, M., & Marcoulides, K. M. (2018). Structural equation modeling with many variables: A systematic review of issues and developments. *Frontiers in Psychology*, *9*, 580. doi: https://doi.org/10.3389/fpsyg.2018.00580

DiStefano, C., Shi, D., & Morgan, G. B. (2021). Collapsing categories is often more advantageous than modeling sparse data: Investigations in the cfa framework. *Structural Equation Modeling: A Multidisciplinary Journal*, *28*(2), 237–249. doi: https://doi.org/10.1080/10705511.2020.1803073

Fox, J. (2022). *polycor: Polychoric and polyserial correlations.* Retrieved from `https://CRAN.R-project.org/package=polycor` (R package version 0.8-1)

Golino, H., & Christensen, A. P. (2025). *Eganet: Exploratory graph analysis – a framework for estimating the number of dimensions in multivariate data using network psychometrics.* Retrieved from `https://r-ega.net` (R package version 2.0.3)

Muthén, L. K., & Muthén, B. O. (2017). *Mplus.* (Version 8)

Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, *44*(4), 443–460. doi: https://doi.org/10.1007/bf02296207

R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. Retrieved from `https://www.R-project.org` (Version 4.2.1)

Revelle, W. (2023). *psych: Procedures for psychological, psychometric, and personality research.* Retrieved from `https://CRAN.R-project.org/package=psych` (R package version 2.3.9)

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. doi: https://doi.org/10.18637/jss.v048.i02

Savalei, V. (2011). What to do about zero frequency cells when estimating polychoric correlations. *Structural Equation Modeling: A Multidisciplinary Journal*, *18*(2), 253–273. doi: https://doi.org/10.1080/10705511.2011.557339

Yang, T.-R., & Weng, L.-J. (2024). Revisiting Savalei's (2011) research on remediating zero-frequency cells in estimating polychoric correlations: A data distribution perspective. *Structural Equation Modeling: A Multidisciplinary Journal*, *31*(1), 81–96. doi: https://doi.org/10.1080/10705511.2023.2220919

Yates, F. (1934). Contingency tables involving small numbers and the χ^2 test. *Supplement to the Journal of the Royal Statistical Society*, *1*(2), 217–235. doi: https://doi.org/10.2307/2983604

Yuan, K.-H., Wu, R., & Bentler, P. M. (2011). Ridge structural equation modeling with correlation matrices for ordinal and continuous data. *The British Journal of Mathematical and Statistical Psychology*, *64*(1), 107-133. doi: https://doi.org/10.1348/000711010x497442

Zhang, G., Trichtinger, L. A., Lee, D., & Jiang, G. (2022). PolychoricRM: A computationally efficient R function for estimating polychoric correlations and their asymptotic covariance matrix. *Structural Equation Modeling: A Multidisciplinary Journal*, *29*(2), 310–320. doi: https://doi.org/10.1080/10705511.2021.1929996

Appendix

Table A1: The Composition of Sampling Distribution of 2×2 Tables for $N = 50$ (Study 1)

ξ_1		1.5	1.5	1.5	1	1	0.8	1.5	1.5	1	1.5	1	0.8
ξ_2		-1.5	-1	-0.8	-1	-0.8	-0.8	0.8	1	0.8	1.5	1	0.8
ρ	Zero Cells	Probabilities											
0.3	0	0.046	0.147	0.213	0.404	0.535	0.674	0.648	0.603	0.943	0.383	0.898	0.976
	1	0.892	0.821	0.756	0.595	0.465	0.326	0.320	0.365	0.057	0.554	0.102	0.024
	2 or 3	0.062	0.032	0.032	0.000	0.000	0.000	0.032	0.032	0.000	0.063	0.000	0.000
0.5	0	0.008	0.041	0.070	0.173	0.271	0.403	0.654	0.664	0.967	0.504	0.948	0.989
	1	0.930	0.928	0.898	0.827	0.729	0.597	0.315	0.304	0.033	0.432	0.052	0.011
	2 or 3	0.062	0.032	0.032	0.000	0.000	0.000	0.032	0.032	0.000	0.065	0.000	0.000
0.7	0	0.000	0.003	0.006	0.025	0.054	0.111	0.504	0.595	0.944	0.559	0.947	0.981
	1	0.938	0.966	0.962	0.974	0.945	0.889	0.465	0.373	0.056	0.368	0.052	0.019
	2 or 3	0.062	0.032	0.032	0.000	0.000	0.000	0.032	0.032	0.000	0.073	0.001	0.000
0.9	0	0.000	0.000	0.000	0.000	0.000	0.000	0.105	0.224	0.713	0.415	0.789	0.864
	1	0.938	0.968	0.968	1.000	1.000	1.000	0.863	0.741	0.282	0.444	0.200	0.131
	2 or 3	0.062	0.032	0.032	0.000	0.000	0.000	0.032	0.036	0.004	0.141	0.011	0.004

Note. ξ_1 and ξ_2 are thresholds for the first and second variables, respectively. Probabilities were rounded to the third decimal place.

Table A2: The Composition of Sampling Distribution of 2×2 Tables for $N = 100$ (Study 1)

ξ_1		1.5	1.5	1.5	1	1	0.8	1.5	1.5	1	1.5	1	0.8
ξ_2		-1.5	-1	-0.8	-1	-0.8	-0.8	0.8	1	0.8	1.5	1	0.8
ρ	Zero Cells	Probabilities											
0.3	0	0.095	0.281	0.393	0.641	0.778	0.886	0.917	0.881	0.988	0.674	0.983	0.988
	1	0.902	0.715	0.603	0.354	0.215	0.105	0.077	0.114	0.002	0.322	0.009	0.000
	2 or 3	0.002	0.001	0.001	0.000	0.000	0.000	0.001	0.001	0.000	0.002	0.000	0.000
	Removed	0.001	0.003	0.003	0.005	0.007	0.009	0.005	0.004	0.010	0.002	0.008	0.012
0.5	0	0.017	0.082	0.139	0.313	0.465	0.639	0.921	0.931	0.990	0.829	0.990	0.988
	1	0.981	0.916	0.858	0.684	0.531	0.355	0.073	0.065	0.000	0.167	0.001	0.000
	2 or 3	0.002	0.001	0.001	0.000	0.000	0.000	0.001	0.001	0.000	0.002	0.000	0.000
	Removed	0.000	0.001	0.002	0.003	0.004	0.006	0.005	0.003	0.010	0.002	0.009	0.012
0.7	0	0.000	0.005	0.012	0.048	0.104	0.206	0.785	0.875	0.989	0.904	0.992	0.989
	1	0.997	0.993	0.985	0.949	0.893	0.790	0.210	0.121	0.002	0.091	0.000	0.000
	2 or 3	0.002	0.001	0.001	0.000	0.000	0.000	0.001	0.001	0.000	0.002	0.000	0.000
	Removed	0.001	0.001	0.002	0.003	0.003	0.004	0.004	0.003	0.009	0.003	0.008	0.011
0.9	0	0.000	0.000	0.000	0.000	0.000	0.000	0.204	0.413	0.920	0.801	0.972	0.984
	1	0.998	0.999	0.998	0.999	0.999	0.998	0.793	0.584	0.074	0.187	0.023	0.009
	2 or 3	0.002	0.001	0.001	0.000	0.000	0.000	0.001	0.001	0.000	0.011	0.000	0.000
	Removed	0.000	0.000	0.001	0.001	0.001	0.002	0.002	0.002	0.006	0.001	0.005	0.007

Note. The structure of this table is the same as Table A1. "Removed" indicates tables excluded from analyses due to their small probability in the sampling distribution. "2 or 3" was also removed from the analysis due to the number of zero cells.

Table A3: The Composition of Sampling Distribution of 2×2 Tables for $N = 200$ (Study 1)

ξ_1		1.5	1.5	1.5	1	1	0.8	1.5	1.5	1	1.5	1	0.8
ξ_2		-1.5	-1	-0.8	-1	-0.8	-0.8	0.8	1	0.8	1.5	1	0.8
ρ	Zero Cells	Probabilities											
0.3	0	0.180	0.482	0.628	0.863	0.937	0.967	0.983	0.978	0.972	0.892	0.976	0.966
	1	0.817	0.512	0.364	0.124	0.045	0.009	0.003	0.011	0.000	0.102	0.000	0.000
	2 or 3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Removed	0.003	0.006	0.008	0.013	0.018	0.024	0.014	0.011	0.028	0.006	0.024	0.034
0.5	0	0.033	0.156	0.258	0.524	0.707	0.859	0.984	0.987	0.972	0.969	0.976	0.967
	1	0.965	0.840	0.737	0.467	0.281	0.125	0.002	0.001	0.000	0.024	0.000	0.000
	2 or 3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Removed	0.002	0.004	0.005	0.009	0.012	0.016	0.014	0.012	0.028	0.007	0.024	0.033
0.7	0	0.000	0.009	0.024	0.094	0.196	0.367	0.947	0.977	0.975	0.990	0.978	0.970
	1	0.998	0.988	0.973	0.901	0.797	0.624	0.042	0.012	0.000	0.003	0.000	0.000
	2 or 3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Removed	0.002	0.003	0.003	0.005	0.007	0.009	0.011	0.011	0.025	0.007	0.022	0.030
0.9	0	0.000	0.000	0.000	0.000	0.000	0.000	0.366	0.653	0.979	0.976	0.985	0.979
	1	0.999	0.999	0.999	0.998	0.998	0.996	0.629	0.341	0.004	0.019	0.000	0.000
	2 or 3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Removed	0.001	0.001	0.001	0.002	0.002	0.004	0.005	0.006	0.017	0.005	0.015	0.021

Note. The structure of this table is the same as Table A2. "Removed" indicates tables excluded from analyses due to their small probability in the sampling distribution. "2 or 3" was also removed from the analysis due to the number of zero cells. The removed cases are those with very small probabilities. As sample size increases, most replications concentrate on 2×2 tables close to the expected table, so many other possible tables become unlikely. As a result, more tables fall into the very low probability range and are removed, which is why the larger-sample condition shown in this table ends up with a larger total probability of removed tables.

Table A4: The Composition of the Simulated Dataset (Study 2)

Correlation	Thresholds	No Zero Cells	One Zero Cell	Two or More Zero Cells (Removed)
0.4	positive	4207	23866	1927
	mixed	0	28239	1761
0.8	positive	409	26683	2908
	mixed	0	27947	2053
mixed	positive	715	27215	2070
	mixed	19	28004	1977

Note. The total number of replications is 30,000 for each condition.

Table A5: The Proportion of Positive-Definite Matrices (Study 2)

Added Value	Way of Adding	Cor 0.4 & Mixed Thresholds	Cor 0.8 & Mixed Thresholds	Cor Mixed & Mixed Thresholds	Cor 0.4 & Positive Thresholds	Cor 0.8 & Positive Thresholds	Cor Mixed & Positive Thresholds
0.00	–	0.000	0.000	0.000	0.076	0.004	0.005
0.10	Only to the Zero Cell	0.249	0.266	0.095	0.172	0.027	0.017
	Keep Marginals	0.265	0.288	0.104	0.181	0.032	0.019
	Add to All	0.237	0.268	0.094	0.167	0.027	0.016
	Add to All Regardless	0.237	0.261	0.092	0.179	0.028	0.019
0.25	Only to the Zero Cell	0.514	0.525	0.222	0.300	0.089	0.050
	Keep Marginals	0.559	0.605	0.277	0.358	0.131	0.078
	Add to All	0.490	0.518	0.210	0.269	0.083	0.041
	Add to All Regardless	0.490	0.508	0.203	0.312	0.095	0.058
0.50	Only to the Zero Cell	0.726	0.772	0.492	0.488	0.229	0.157
	Keep Marginals	0.751	0.851	0.638	0.581	0.382	0.257
	Add to All	0.694	0.753	0.428	0.410	0.208	0.112
	Add to All Regardless	0.715	0.759	0.403	0.535	0.272	0.201

Table A6: Quadratic Form Loss (Study 2)

Added Value	Way of Adding	Cor 0.4 & Mixed Thresholds	Cor 0.8 & Mixed Thresholds	Cor Mixed & Mixed Thresholds	Cor 0.4 & Positive Thresholds	Cor 0.8 & Positive Thresholds	Cor Mixed & Positive Thresholds
0.00	–	19.944	29.372	32.886	17.955	43.544	44.983
0.10	Only to the Zero Cell	5.047	37.650	12.150	6.402	15.629	15.550
	Keep Marginals	4.987	38.910	11.922	6.233	15.136	15.121
	Add to All	5.095	36.470	12.311	6.541	15.992	15.891
	Add to All Regardless	5.107	36.280	12.530	6.430	15.920	15.765
0.25	Only to the Zero Cell	4.646	60.434	9.554	4.787	12.121	11.843
	Keep Marginals	4.793	65.822	9.301	4.439	11.327	11.161
	Add to All	4.557	55.836	9.772	5.073	12.743	12.448
	Add to All Regardless	4.592	55.154	10.144	4.807	12.551	12.092
0.50	Only to the Zero Cell	5.472	90.158	8.249	3.687	10.260	9.799
	Keep Marginals	6.553	105.776	9.148	3.306	10.715	10.187
	Add to All	4.876	78.200	8.221	4.088	10.780	10.365
	Add to All Regardless	4.954	76.497	8.667	3.605	10.401	9.648

Note. Values are rounded to the third decimal place.

Table A7: Root Mean Square Error (RMSE) in the Mean of Loadings (Study 3)

		Loadings of 0.4 & Positive Thresholds			Loadings of 0.7 & Positive Thresholds			Loadings of 0.4 & Mixed Thresholds			Loadings of 0.7 & Mixed Thresholds		
Added Value	Way of Adding	0.8	1	1.5	0.8	1	1.5	0.8	1	1.5	0.8	1	1.5
0.1	Only to the zero cell	0.429	0.467	0.484	0.220	0.250	0.346	0.418	0.435	0.488	0.201	0.223	0.417
	Keep marginals	0.428	0.464	0.475	0.220	0.249	0.337	0.417	0.434	0.491	0.201	0.223	0.429
	Add to all	0.429	0.469	0.492	0.220	0.250	0.357	0.418	0.435	0.485	0.201	0.222	0.406
	Add to all regardless	0.423	0.464	0.492	0.216	0.246	0.353	0.419	0.437	0.486	0.204	0.223	0.405
0.25	Only to the zero cell	0.418	0.434	0.384	0.218	0.237	0.266	0.407	0.420	0.551	0.194	0.236	0.609
	Keep marginals	0.415	0.429	0.366	0.217	0.235	0.244	0.406	0.420	0.581	0.194	0.241	0.671
	Add to all	0.420	0.440	0.400	0.218	0.239	0.285	0.408	0.421	0.526	0.193	0.231	0.559
	Add to all regardless	0.405	0.426	0.398	0.210	0.228	0.278	0.410	0.427	0.530	0.201	0.236	0.557
0.5	Only to the zero cell	0.406	0.403	0.333	0.215	0.228	0.211	0.398	0.416	0.650	0.195	0.276	0.794
	Keep marginals	0.402	0.393	0.340	0.214	0.224	0.190	0.397	0.420	0.716	0.200	0.302	0.866
	Add to all	0.409	0.413	0.342	0.216	0.232	0.234	0.399	0.415	0.597	0.192	0.261	0.715
	Add to all regardless	0.381	0.385	0.339	0.201	0.213	0.221	0.405	0.427	0.604	0.207	0.268	0.727

Note. Values are rounded to the third decimal place.

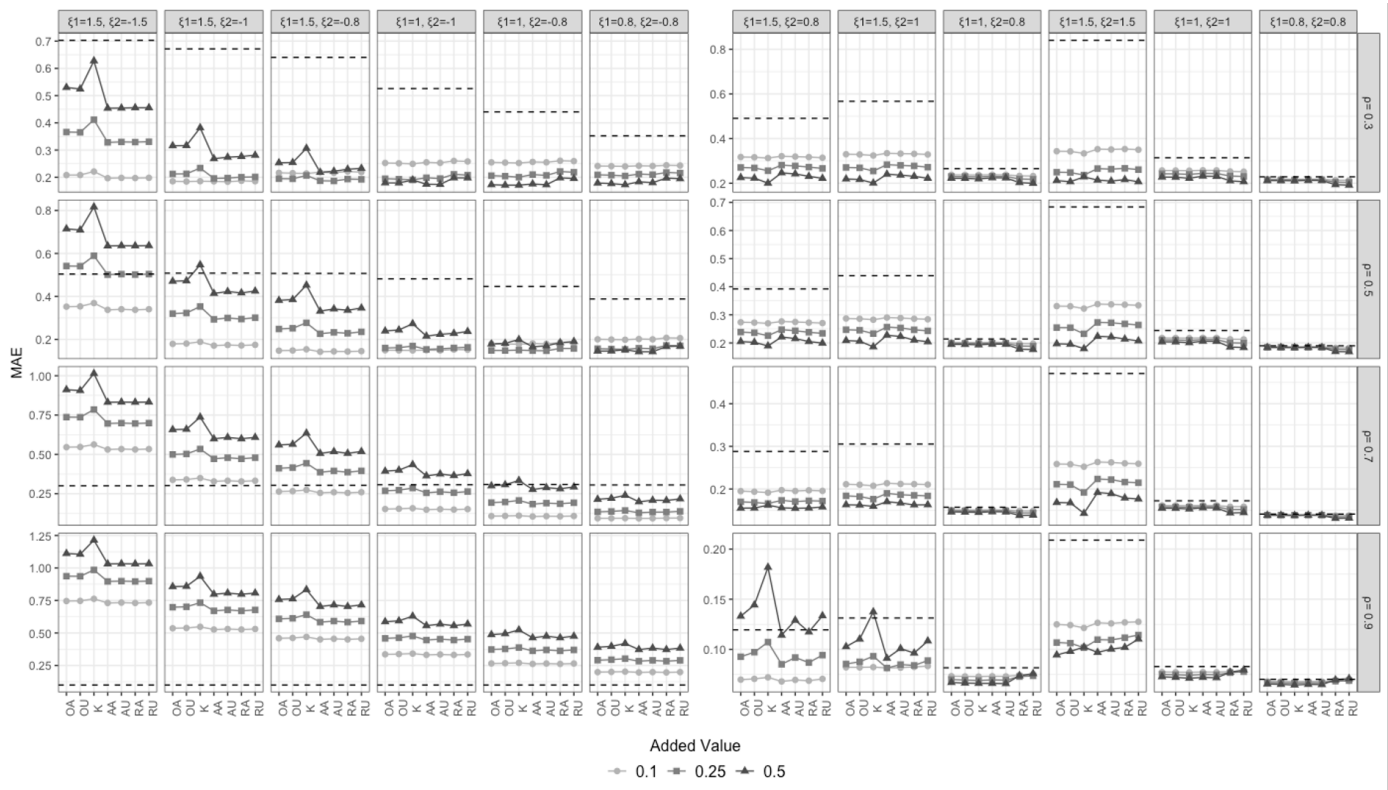


Figure A1: Mean Absolute Error (MAE) for Point Estimates (N=50)

Note. The structure of this figure is the same as Figure 2 in the paper.

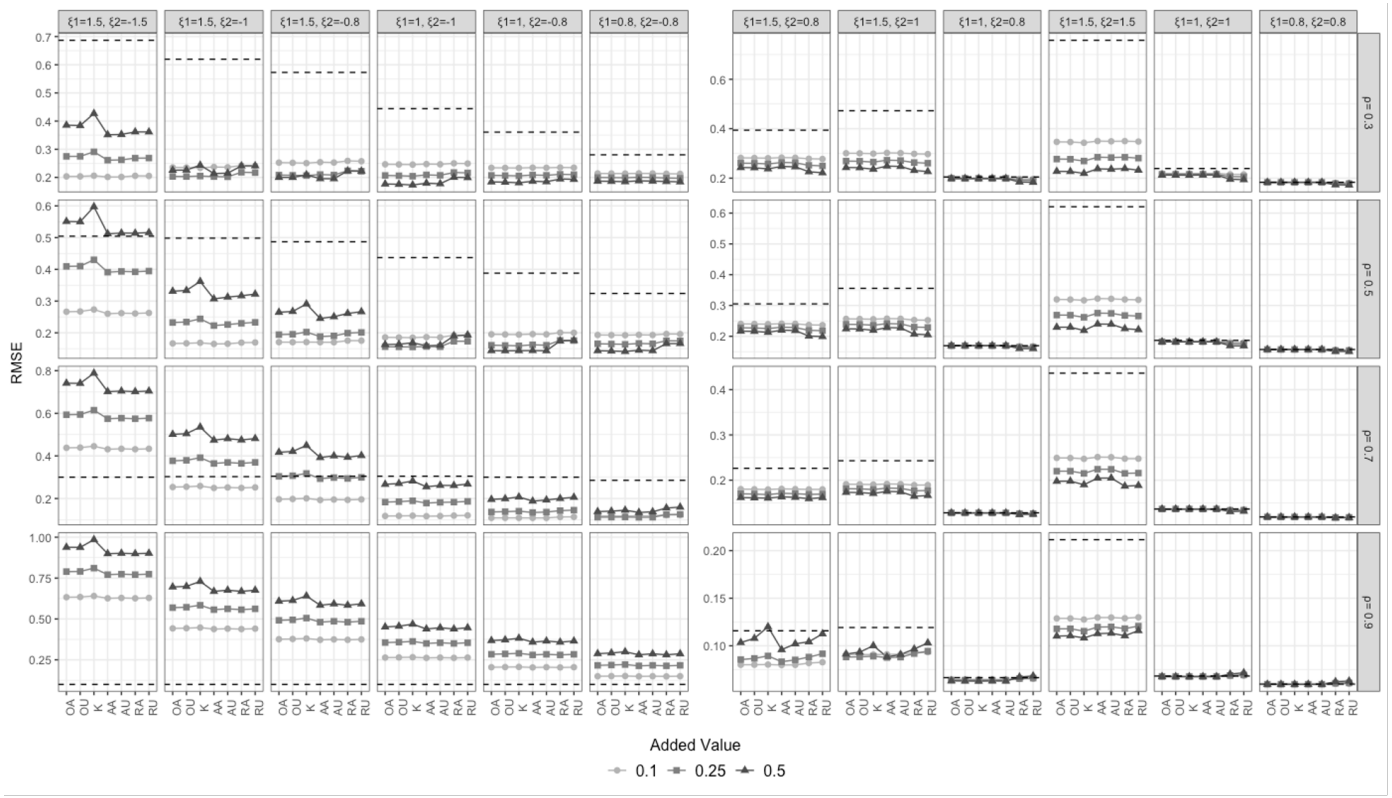


Figure A2: Root Mean Square Error (RMSE) for Point Estimates (N=100)

Note. The structure of this figure is the same as Figure 2 in the paper.

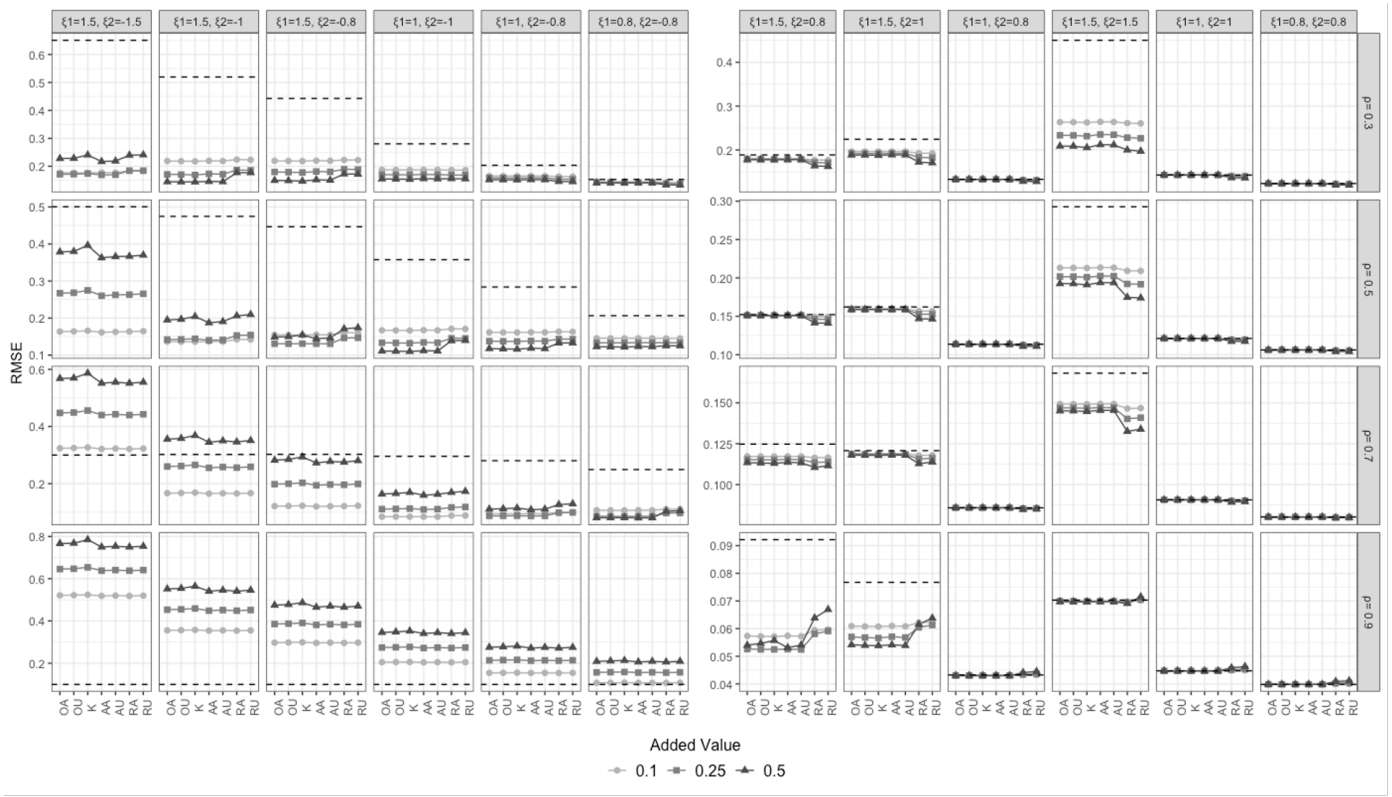


Figure A3: Root Mean Square Error (RMSE) for Point Estimates (N=200)

Note. The structure of this figure is the same as Figure 2 in the paper.

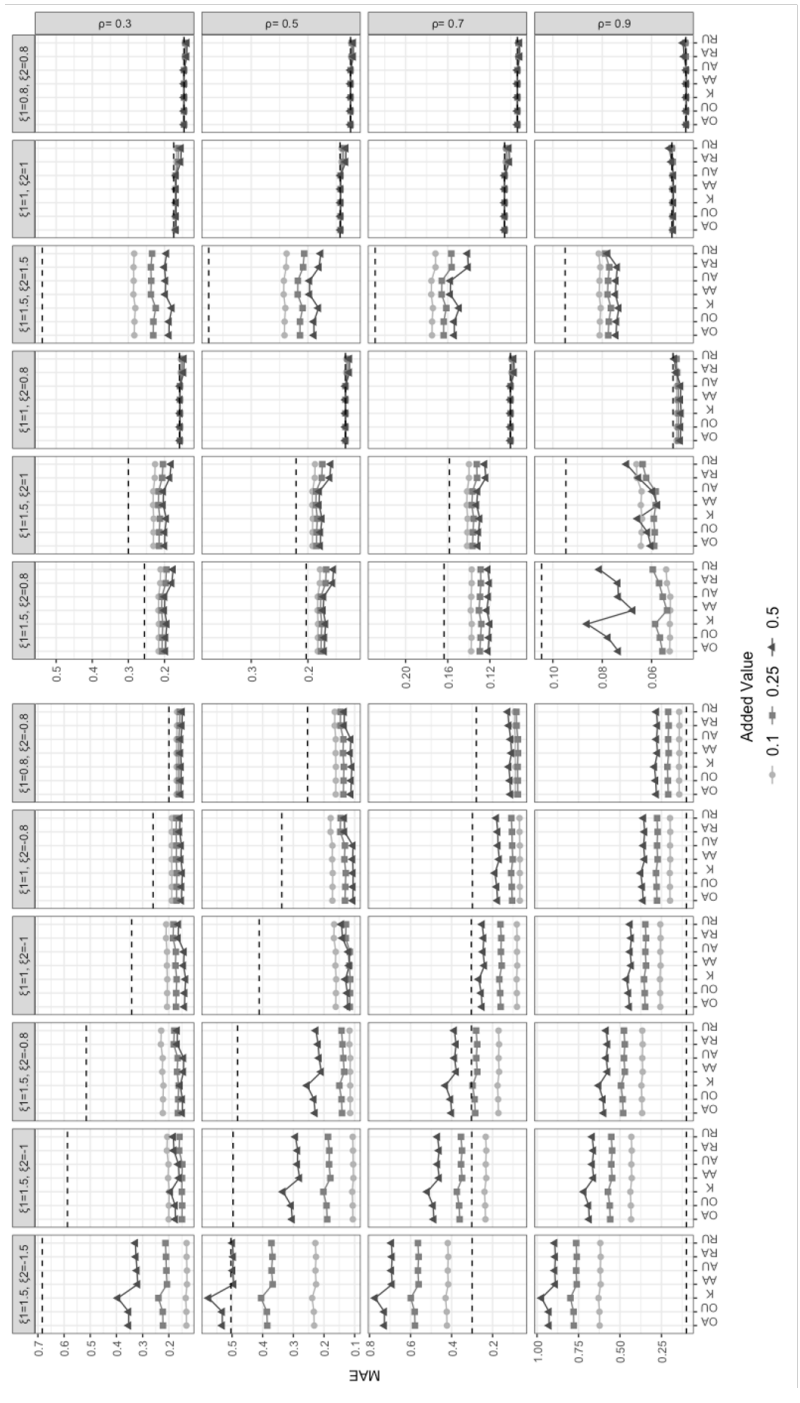


Figure A4: Mean Absolute Error (MAE) for Point Estimates (N=100)

Note. The structure of this figure is the same as Figure 2 in the paper.

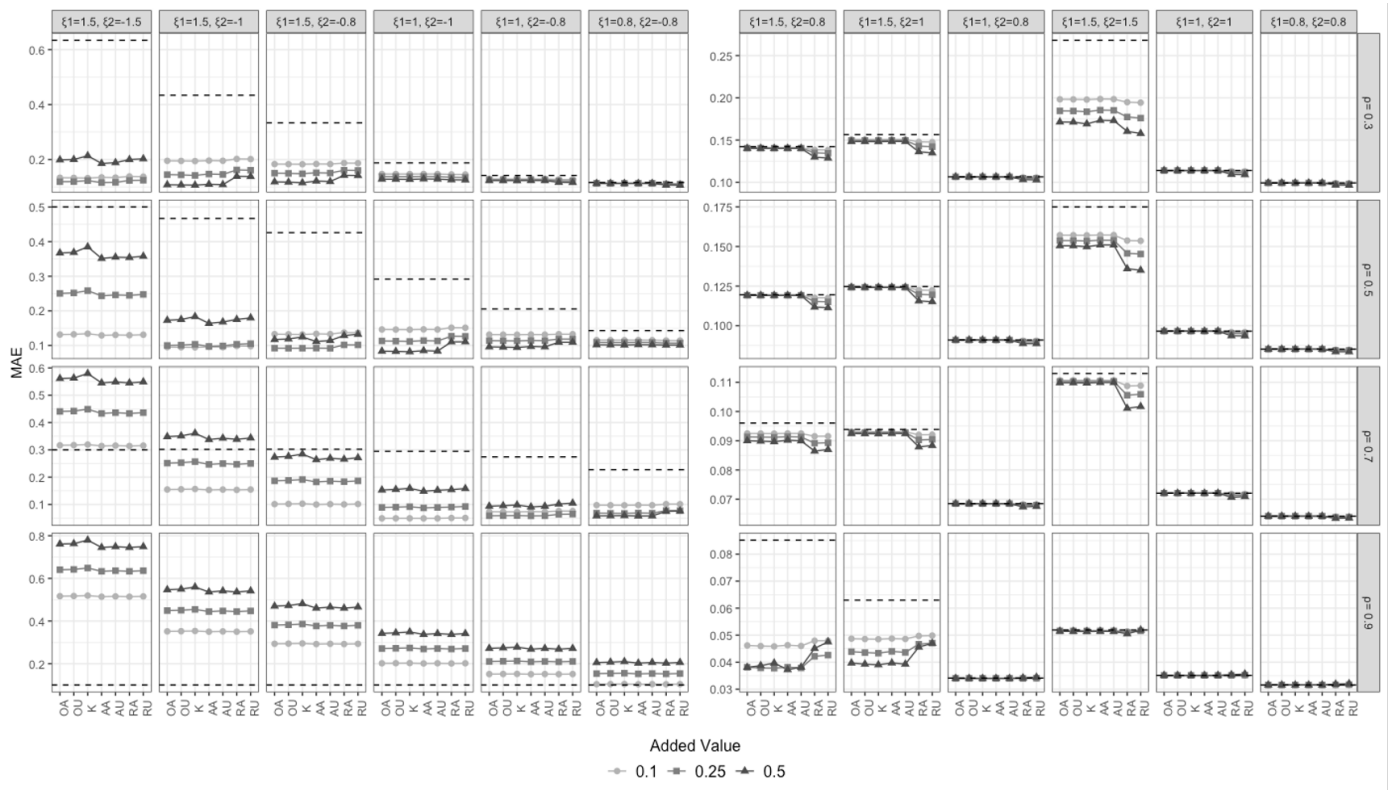


Figure A5: Mean Absolute Error (MAE) for Point Estimates (N=200)

Note. The structure of this figure is the same as Figure 2 in the paper.

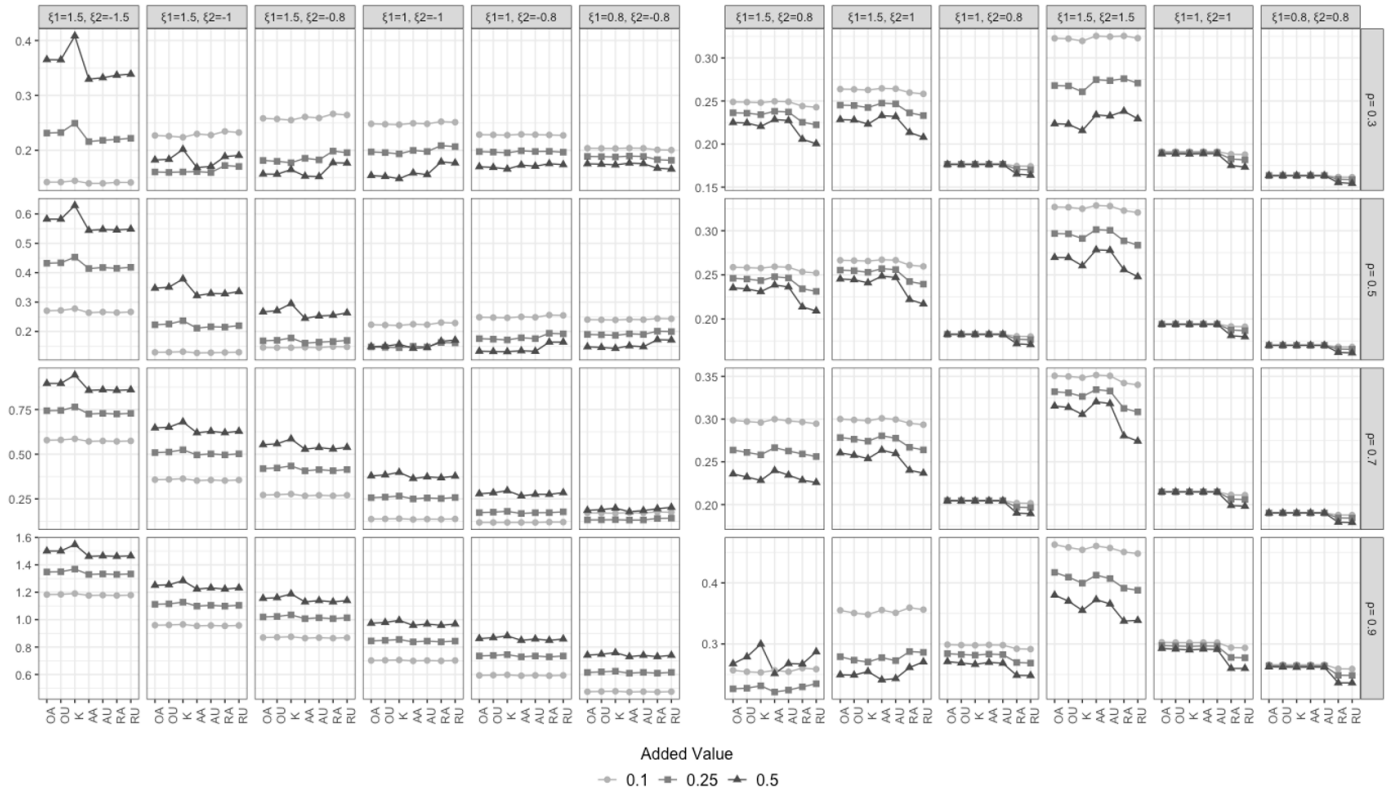


Figure A6: Mean Absolute Error (MAE) for Point Estimates after Fisher's Z-Transformation (N=100)

Note. The structure of this figure is the same as Figure 3 in the paper.

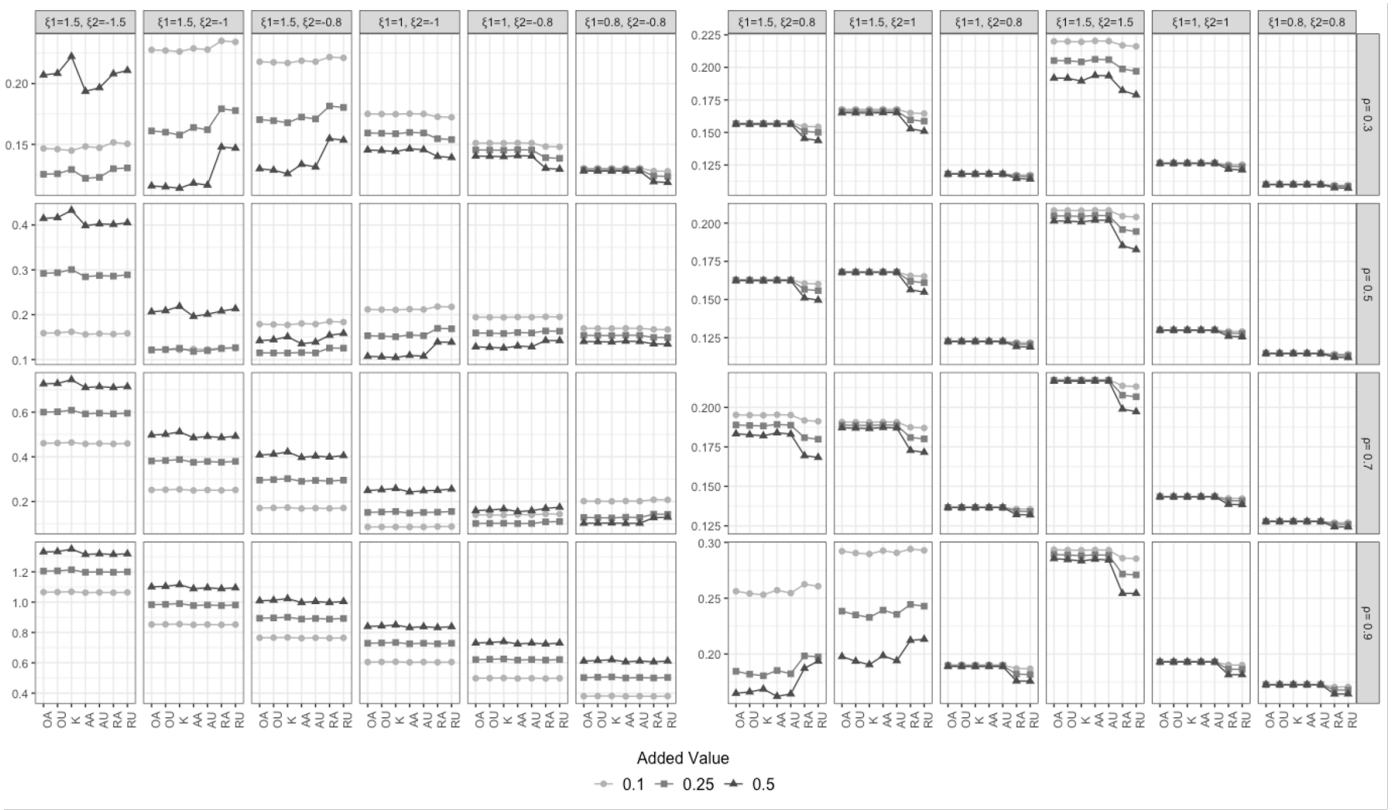


Figure A7: Mean Absolute Error (MAE) for Point Estimates after Fisher's Z-Transformation (N=200)

Note. The structure of this figure is the same as Figure 3 in the paper.

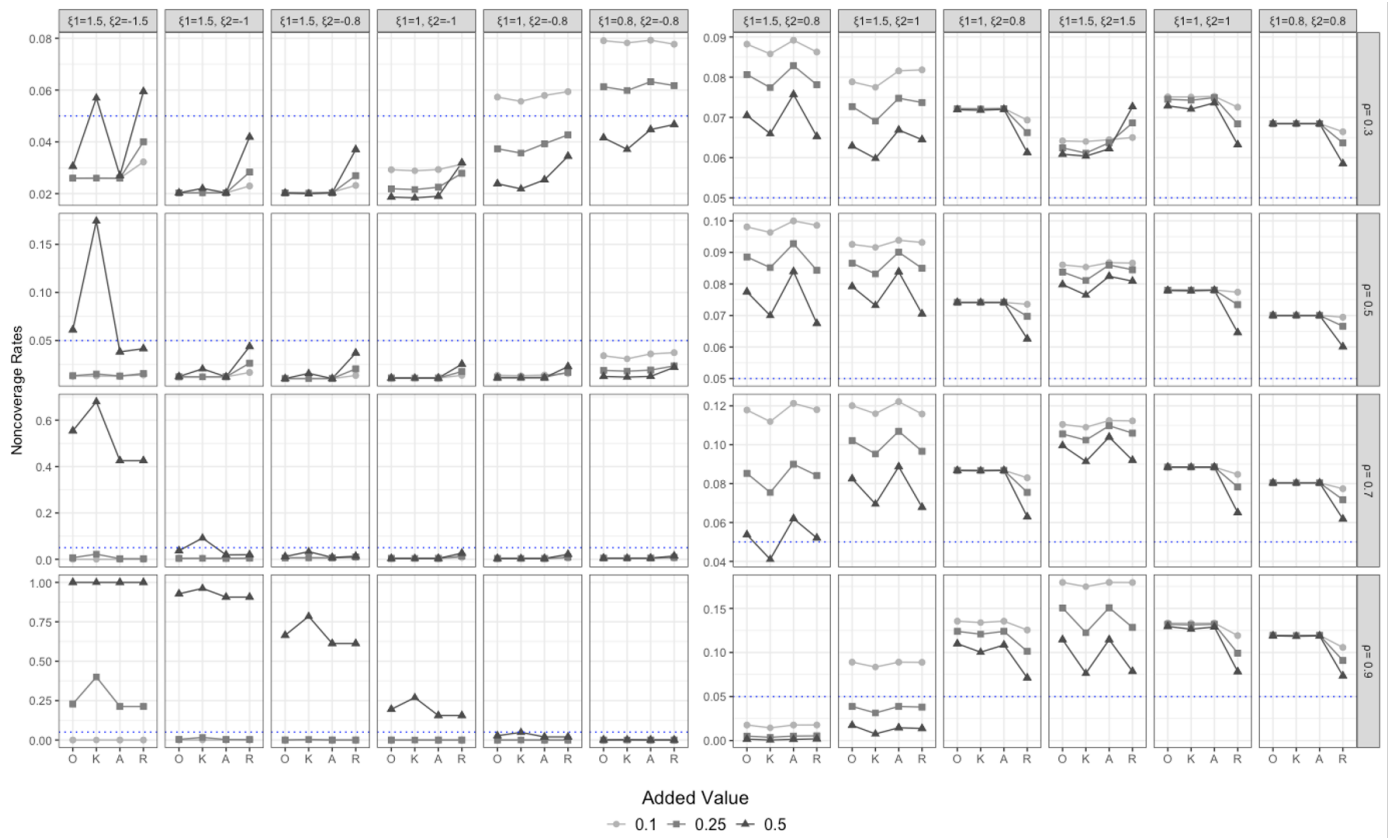


Figure A8: Noncoverage Rate of the 95% Wald Confidence Interval of the Correlation (N=100)

Note. The structure of this figure is the same as Figure 4 in the paper.

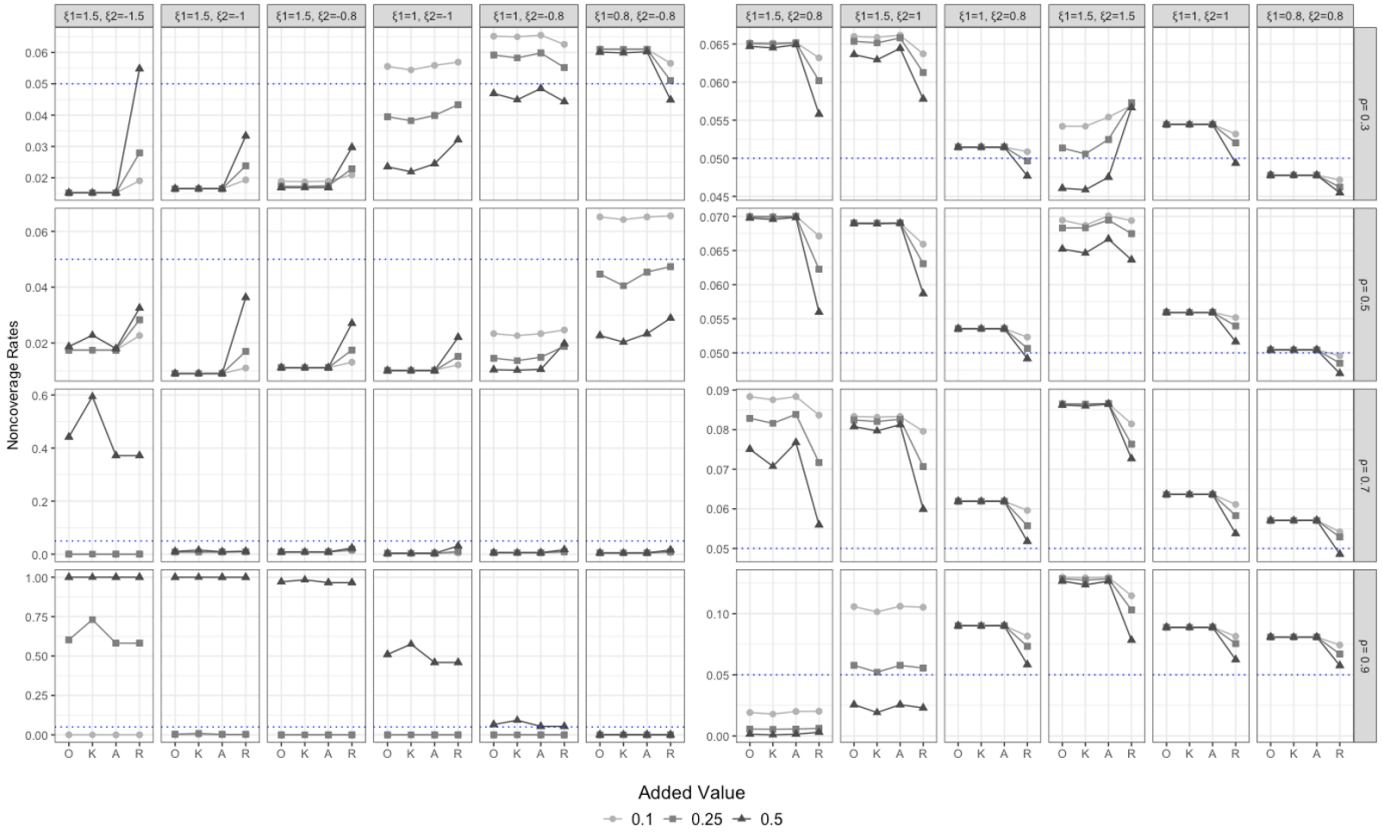


Figure A9: Noncoverage Rate of the 95% Wald Confidence Interval of the Correlation (N=200)

Note. The structure of this figure is the same as Figure 4 in the paper.

Journal of Behavioral Data Science, 2026, *6* (1), 41-67.
DOI: https://doi.org/10.35566/jbds/luchman

Determining Relative Importance with Independent Variable Groups: An Alternative Dominance Analysis Method

Joseph N. Luchman[1] [0000−0002−8886−9717]

Fors Marsh
jluchman@gmail.com; jluchman@forsmarsh.com

Abstract. Grouping independent variables (IVs) in relative importance analyses like dominance analysis (DA) can reduce computation time and improve analysis feasibility. A side effect of grouping IVs is that determining the importance of individual IVs within groups is not possible. This work proposes an extension of DA where the researcher can group IVs yet still compare individual IVs. The proposed extension extends from Owen values, a variant of the Shapley value solution concept from Cooperative Game Theory. Owen values are used to generate within-group DA statistics and designations which can be used for relative importance analysis of individual IVs in IV groups. This manuscript provides an analytic example with data in the R statistical computing environment that shows how within-group DA statistics and designations are determined. The manuscript concludes by discussing how to group IVs into IV groups and potential extensions to the method to simplify the IV grouping process.

Keywords: Relative Importance Analysis · Dominance Analysis · Shapley Value Decomposition · Owen Value Decomposition

Comparing independent variables (IVs) in terms of how each contributes to predicting a dependent variable is a common practice when evaluating statistical models like linear regression and is known as relative importance analysis (e.g., Tonidandelx& LeBreton, 2011). Historically, many research applications of relative importance analysis have used simple approaches to evaluating IV contributions to prediction such as comparing standardized regression coefficients, the change in the R^2 or ΔR^2 when each IV when included first in the model, or the ΔR^2 when an IV is included last after all other IVs are included (Grömping, 2007; Johnsonx& LeBreton, 2004). Relative importance analysis has, however, been defined as a method that evaluates contributions that an IV makes when alone as well as the contribution it makes when included along with other IVs (Johnsonx& LeBreton, 2004). The definition of relative importance analysis has led statisticians to recommend methods that can accommodate the contributions

IVs make to prediction in the context of different subsets of IVs simultaneously like *Dominance Analysis* (DA; Budescux& Azen, 2004). DA is one of the most widely used relative importance methods in the literature and has been applied to research in multiple behavioral science domains including understanding response styles on surveys (e.g., Miller, Kirby,x& Stevens, 2025), communication patterns in educational settings (e.g., Yinx& Zhou, 2025), and longitudinal injury risks (e.g., McLaurin, West,x& Thomson, 2025).

DA differs from other recommended relative importance methods (e.g., relative weights, Pratt's method/geometrical decomposition; Johnsonx& LeBreton, 2004; Thomas, Zumbo, Kwan,x& Schweitzer, 2014) in that it generates a hierarchy of pairwise dominance designations that describe the relative importance of two IVs. Across all pairs of IVs, the hierarchy of dominance designations results in a rank ordering of the IVs from which importance can be determined. The three dominance designations, in order of the strength of evidence they provide about the relative importance of two IVs, are: *complete*, *conditional*, and *general* (Azenx& Budescu, 2003). The three strength of evidence levels reported by DA provide the researcher much detail about the predictive utility of each IV relative to each other IV in the model. However detailed, the hierarchy of designations generated by DA is only possible because the method uses ΔR^2 values for all possible combinations of IVs included in the model. Computing ΔR^2 values for all possible IV combinations is a computationally expensive methodology and, as the number of IVs gets large, DA grows computationally intractable.

Restructuring DA by making inseparable groups of IVs can mitigate the effect of having many IVs in a model as the ΔR^2 values required will derive from all combinations of IV groups instead of all combinations of IVs (e.g., Bittmann, 2024; Gu, 2023; Luchman, 2021). This is because all members of an IV group are included simultaneously and the predictive usefulness/ΔR^2 value of the group as a whole is used in the dominance statistics and designations. Grouping IVs is not a full solution to the issue of having many IVs as it results in a strong constraint on the determinations from the DA. The constraint imposed by grouping IVs is that a researcher will not be able to determine the importance of individual IVs within an IV group as all IVs in an IV group are inseparable.

The purpose of this manuscript is to develop a within-group extension of DA that uses IV groups yet will allow relative importance determinations for IVs within an IV group. As I will show, the proposed within-group extension of DA combines aspects of the DA method considering all IVs in the model separately, referred to from this point on as the traditional DA method, with aspects of the DA method where IVs are grouped into inseparable IV groups, referred to from this point on as the grouped DA method. The primary benefit of the proposed within-group method is that it retains the traditional DA method's ability to compare all IVs to one another while also substantially reducing the number of required combinations for obtaining dominance designations similar to the grouped DA method. For example, I show later in this manuscript that a statistical model with 20 IVs could be grouped such that the within-group DA method requires only .01% of the combinations required for the traditional DA

method yet still would allow each IV to be compared to each other IV. As a result, the proposed within-group DA method offers researchers a way to apply relative importance analysis to statistical models with a large number of IVs—so long as those IVs can be grouped together.

I discuss the proposed within-group DA method in a series of five sections. First, I review computational details of the traditional DA method. Extending from the review of the traditional DA methodology, I discuss how the traditional DA method's general dominance statistics derive from the Shapley value solution concept in cooperative game theory (Shapley, 1953). Third, I discuss the grouped DA/Shapley value method where IVs are bundled into inseparable IV groups. The discussion of grouped DA/Shapley values leads directly to introducing Owen values (Owen, 1977), an extension of Shapley values where the computation is broken into multiple stages that depend on the structure of the IV groups. Finally, I show how Owen values can be translated back into DA computations and define the within-group DA methodology.

Following the five conceptual sections, this manuscript provides an analytic example which includes a detailed account of computing traditional, grouped, and within-group DA statistics and designations using the `ability.cov` data from the `datasets` package in the R statistical computing environment.

1 Computational Details of Dominance Analysis

1.1 Sequences of IVs

Imagine I am using variables α, β, ξ, and ζ, as IVs in a linear regression model and that I am seeking to determine the relative importance of these four IVs in predicting some dependent variable. As I mentioned in the introduction, approaches to determining relative importance have historically focused on computing ΔR^2 for IVs when they are included in the model and I begin by taking that approach. As opposed to the approaches discussed in the introduction (i.e., comparing IVs when included first or last), another way I could determine the relative importance of all four IVs is to use ΔR^2 values for each IV as included in a specific sequence. If I were to use the sequence of the four IVs described above, I would then obtain four values:

1. R^2 value for α,
2. ΔR^2 for α and β over α,
3. ΔR^2 for α, β, and ξ over α and β,
4. ΔR^2 for α, β, ξ, and ζ over α, β, and ξ.

This sequence of R^2 and ΔR^2 values are comparable to one another as as relative importance results and will result in a full decomposition of the model's R^2 that can be ascribed to each IV (e.g., Kruskal, 1987). Therefore, an added benefit of this method is that each value is simple to interpret as each value is a part of the whole model R^2 and can be described as a percentage of the R^2.

The sequential method described in the previous paragraph requires a researcher to predetermine an *inclusion precedence sequence* for the IVs in their model. I define an inclusion precedence sequence to be a determination by the researcher as to the order in which the IVs will be included in the model to compute, and compare, R^2 or ΔR^2 values. In addition, I define the *inclusion precedence position* of an IV as an IV's sequential position of inclusion in a specific inclusion precedence sequence. IVs with a higher inclusion precedence position precede, and are thus included before, IVs with a lower IV inclusion precedence position.

Predetermining a single IV inclusion precedence sequence is a simple and compelling, but infrequently used, method for determining IV relative importance. This method is infrequently used as it is often difficult to determine a best, most plausible inclusion precedence sequence for a set of IVs. That is, there may be reasonable disagreements that different researchers would have about the inclusion precedence sequencing for the IVs in a model. When IVs are correlated with one another, such disagreements over inclusion precedence sequencing can lead to different conclusions about IV relative importance as IVs that are positioned earlier in the sequence are ascribed components of the R^2 that overlap among multiple IVs. Hence, if α, β, ξ, and ζ, are correlated, their positions in the sequence can strongly affect their relative importance determinations.

1.2 Dominance Analysis: Use of all Sequences

The DA methodology derives directly from the sequential/predetermined inclusion precedence sequence approach by averaging across all possible IV inclusion precedence sequences (Budescu, 1993). DA then needs no predetermined inclusion precedence sequence as all sequences are incorporated into the dominance designations used to determine IV relative importance.

All three dominance designations (i.e., complete, conditional, and general) evaluate the inclusion precedence sequences of a set of IVs, $\mathbf{X}$, with a comparison methodology for any given pair of IVs that follows Equation equation1.

$$D_{\alpha > \beta} = \sum_{i=1}^{K} \frac{\begin{cases} 1, \ V_i^{\alpha} > V_i^{\beta} \\ 0, \ Otherwise \end{cases}}{K} \tag{1}$$

where $\alpha \in \mathbf{X}$ (i.e., α is a member of $\mathbf{X}$), $\beta \in \mathbf{X}$, K is a number of comparisons, V_i^{α} is a value (e.g., a ΔR^2) that is associated with α given comparison i, and V_i^{β} is a value that is associated with β given the same comparison i. Equation equation1 shows that each pairwise dominance designation is built on the proportion of comparisons between α and β that favor α. A dominance designation is achieved by α when, for that level of dominance, the proportion produced by Equation equation1 is a value of 1. Thus, dominance designations are only achieved when all the comparisons favor α over β.

Recall that each dominance designation differs in terms of the strength of evidence it offers about a pair of IVs. The key difference between the dominance designations is the number of comparisons, K, that are made between the two IVs. Complete dominance has the most comparisons (i.e., $2^{|\mathbf{X}|-2}$; where $|\mathbf{X}|$ is the cardinality, or number of members in, $\mathbf{X}$) which is the reason why it is the strongest designation as it is the hardest to achieve. General dominance has the fewest comparisons (i.e., 1) which is the reason why it is the weakest designation as it is the easiest to achieve. Conditional dominance falls in-between complete and general in terms of the number of comparisons (i.e., $|\mathbf{X}|$) and in terms of its strength of evidence.

The V_i^{α} and V_i^{β} values in Equation equation1 also differ across dominance designations and incorporate the inclusion precedence sequences differently. For instance, complete dominance uses ΔR^2 values directly which produces a larger K value as many specific comparisons, i, are required. By contrast, general dominance averages all the ΔR^2 values for an IV into a single statistic. In the sections to come, I provide details on the computation of the V_i^{α} and V_i^{β} values for each of the dominance designations beginning with complete dominance. Note that following the five conceptual sections, section six provides computational details for most of the dominance designations discussed in the coming three subsections.

1.3 Complete Dominance

Complete dominance seeks to determine whether α produces larger ΔR^2 values than β across all possible inclusion precedence sequences in which the other $|\mathbf{X}| - 2$ IVs would have higher inclusion precedence positions. Computationally, the focus of complete dominance is on combinations of other IVs as opposed to sequences of other IVs. This is because the R^2 is agnostic about the order in which IVs are included. For example, imagine I'm seeking to compare the ΔR^2 of α and β beyond ξ and ζ. Because the R^2 value when ξ and ζ are included prior to α or β does not change depending on whether ξ or ζ is included as the first IV in the sequence, the specific inclusion precedence positions of ξ and ζ for complete dominance designations is irrelevant. Thus, the number of comparisons for complete dominance focuses on unique ΔR^2 values which will be all combinations of other IVs that have a higher inclusion precedence position, and precede, α and β.

I will refer to the comparison IV subsets that include all other IVs as $\mathbf{X}_i^{\setminus\{\alpha,\beta\}}$ where $\mathbf{X}_i^{\setminus\{\alpha,\beta\}} \in \mathcal{P}(\mathbf{X} \setminus \{\alpha, \beta\})$. This means that IV subset $\mathbf{X}_i^{\setminus\{\alpha,\beta\}}$ is one member, i, of the power set (i.e., $\mathcal{P}()$) of $\mathbf{X}$ omitting α and β. The power set is a set that contains all possible subsets of members of a set. Thus, $\mathbf{X}_i^{\setminus\{\alpha,\beta\}}$ refers to one IV subset of the all the possible ways of combining the IVs in $\mathbf{X} \setminus \{\alpha, \beta\}$ into IV subsets.

The V_i^{α} values used in Equation equation1 for complete dominance are computed as $R^2_{\mathbf{X}_i^{\setminus\{\alpha,\beta\}}\cup\alpha} - R^2_{\mathbf{X}_i^{\setminus\{\alpha,\beta\}}} = \Delta R^2_{\mathbf{X}_i^{\setminus\{\alpha,\beta\}}\cup\alpha}$. The value $\Delta R^2_{\mathbf{X}_i^{\setminus\{\alpha,\beta\}}\cup\alpha}$ is then the increment to the R^2 that α makes when it is included with the set of

IVs in $\mathbf{X}_i^{\setminus\{\alpha,\beta\}}$. The V_i^{β} values are computed as $\Delta R^2_{\mathbf{X}_i^{\setminus\{\alpha,\beta\}}\cup\beta}$ where β is substituted for α. Complete dominance is then determined by comparing V_i^{α} and V_i^{β} across all $K = 2^{(|\mathbf{X}|-2)} = |\mathcal{P}(\mathbf{X} \setminus \{\alpha, \beta\})|$ comparisons of the different members of $\mathcal{P}(\mathbf{X} \setminus \{\alpha, \beta\})$.

Because the complete dominance designation requires every α versus β ΔR^2 comparison to favor α, complete dominance is a comprehensive, non-compensatory designation and is difficult to meet for most IV pairs. When IV relative importance cannot be determined using complete dominance, it may still be possible to determine IV relative importance using conditional dominance.

1.4 Conditional Dominance

Conditional dominance seeks to determine whether α, on average, produces larger ΔR^2 values at a specific inclusion precedence position, P, than does β across all possible $|\mathbf{X}|$ inclusion precedence positions.

The V_i^{α} values used in Equation equation1 for conditional dominance are known as conditional dominance statistics and are computed as in Equation equation2.

$$C_{\alpha}^{P} = \sum_{i=1}^{B^P} \frac{\Delta R^2_{\mathbf{X}_i^{\setminus\alpha:P}\cup\alpha}}{B^P} \tag{2}$$

where $\mathbf{X}_i^{\setminus\alpha:P} \in \mathcal{P}(\mathbf{X} \setminus \alpha) : |\mathbf{X}_i^{\setminus\alpha:P}| = P - 1$ and $B^P = \frac{(|\mathbf{X}|-1)!}{(P-1)!([|\mathbf{X}|-1]-[P-1])!}$.

The set $\mathbf{X}_i^{\setminus\alpha:P}$ is then defined as one member, i, of the power set of $\mathbf{X}$ excluding α with the constraint that it must have $P - 1$ members. Similar to complete dominance, conditional dominance then also considers combinations of other IVs that could precede α given the agnosticism of the R^2 statistic related to inclusion precedence. As compared to complete dominance, conditional dominance incorporates all inclusion precedence sequences of IVs more directly as the value of B^P. This is because B^P is composed of all possible sequences of IVs given α is at a fixed inclusion precedence position (i.e., the numerator value '$(|\mathbf{X}| - 1)!$)' adjusting for redundant R^2 values given the number of IVs that would precede α (i.e., the '$(P-1)!$' component of the denominator) and the number of IVs that would succeed α (i.e., the '$([|\mathbf{X}| - 1] - [P - 1])!$' component of the denominator). Moreover, $B^P = |(\mathcal{P}(\mathbf{X} \setminus \alpha) : |\mathbf{X}_i^{\setminus\alpha:P}| = P - 1)|$. The value of B^P is then equivalent to the number of combinations of the IVs in $\mathbf{X} \setminus \alpha$ such that they have $P - 1$ members.

The V_i^{β} values are computed using C_{β}^{P} which substitutes β for α in Equation equation2 and conditional dominance is determined by comparing V_i^{α} and V_i^{β} across all $K = |\mathbf{X}|$ inclusion precedence positions.

Conditional dominance is a more compensatory relative importance determination than complete dominance in that ΔR^2 values for α at a specific inclusion precedence position need not always be larger than those for β, they just need to be on average larger. Hence, at a specific inclusion precedence position, one

or more ΔR^2 value(s) for α can be smaller than those for β and yet α could still be determined to conditionally dominate β. When relative importance for a pair of IVs cannot be determined using complete or conditional dominance, it may still be possible to determine relative importance for the pair using general dominance.

1.5 General Dominance

General dominance seeks to determine whether α produces larger average conditional dominance statistics than does β. The V_i^{α} value used in Equation equation1 for general dominance is known as the general dominance statistic and it is computed as in Equation equation3:

$$C_{\alpha} = \sum_{P=1}^{|\mathbf{X}|} \frac{C_{\alpha}^{P}}{|\mathbf{X}|}. \tag{3}$$

The V_i^{β} value is computed using as C_{β} where β is substituted for α in Equation equation3. General dominance is determined by comparing each IV's general dominance statistic which means that $K = 1$ or that there is only one comparison necessary to determine general dominance.

General dominance statistics are equivalent to Shapley values that have been used as a method to produce an additive decomposition of the R^2 which ascribes components of its value to the IVs included in the model (e.g., Grömping, 2007). In the section below, I show how general dominance statistics translate into Shapley values.

2 Dominance Analysis and its Relationship with Shapley Values

General dominance statistics for α are the arithmetic average of α's conditional dominance statistics and, if I were to combine Equations equation2 and equation3 and expand on the B^P term, I would obtain:

$$C_{\alpha} = \sum_{P=1}^{|\mathbf{X}|} \sum_{i=1}^{B^P} \frac{(P-1)!([|\mathbf{X}|-1]-[P-1])!}{|\mathbf{X}|!} \Delta R^2_{\mathbf{X}_i^{\setminus \alpha:P} \cup \alpha}. \tag{4}$$

The result in Equation equation4 is implied by taking the multiplicative inverse of B^P and considering that $(|\mathbf{X}|-1)! \cdot |\mathbf{X}| = |\mathbf{X}|!$. Equation equation4 also shows that C_{α} statistics are a weighted average of the ΔR^2 values associated with including α beyond each possible combination of the other $|\mathbf{X}|-1$ IVs. Additionally, Equation equation4 shows that the weight applied to each ΔR^2 is proportional to the number of times α appears at a specific inclusion precedence position given all possible inclusion precedence sequences, or permutations, of

IVs in $\mathbf{X}$. As was discussed in the section on conditional dominance, this is because there will be $(P-1)!$ ways that the IVs with a higher inclusion precedence (i.e., the IVs in $\mathbf{X}_i^{\setminus\alpha:P}$) will precede α. There will also be $([|\mathbf{X}|-1]-[P-1])!$ ways that IVs with a lower inclusion precedence (i.e., the IVs in $\mathbf{X} \setminus \mathbf{X}_i^{\setminus\alpha:P}$) will succeed α.

As opposed to expressing C_α as a weighted average of all combinations of IVs in $\mathbf{X}$, C_α can also be expressed as an average of all inclusion precedence sequences of IVs in $\mathbf{X}$ as in Equation equation5:

$$C_\alpha = \sum_{i=1}^{|\mathbf{X}|!} \frac{\Delta R^2_{\tilde{\mathbf{X}}_i}}{|\mathbf{X}|!}, \tag{5}$$

where $\tilde{\mathbf{X}}_i = (\tilde{\mathbf{X}}_>^{\setminus\alpha}, \alpha, \tilde{\mathbf{X}}_<^{\setminus\alpha})_i \in \mathsf{Sym}(\mathbf{X})$ and $\mathsf{Sym}()$ is the symmetric group function which creates all possible permutations of the elements of a set.

The term $\tilde{\mathbf{X}}_i$ is then a complete inclusion precedence sequence, i, of all of the IVs in $\mathbf{X}$ where $\tilde{\mathbf{X}}_>^{\setminus\alpha}$ is an ordered set, or sub-sequence, of IVs that have higher inclusion precedence positions than, and thus precede, α in sequence i and $\tilde{\mathbf{X}}_<^{\setminus\alpha}$ is a sub-sequence of the IVs that have lower inclusion precedence positions than, and thus succeed, α in sequence i. In addition, the value of $\Delta R^2_{\tilde{\mathbf{X}}_i}$ is constructed as $R^2_{(\mathbf{X}_>^{\setminus\alpha},\alpha)_i} - R^2_{(\mathbf{X}_>^{\setminus\alpha})_i}$.

Equation equation5 is also a simplified formulation for Shapley values that more clearly illustrates its conceptual origins in ordered sequences. Indeed, the primary difference between Equations equation4 and equation5 is that the numerator of the weight in Equation equation4 is translated into additional ΔR^2 values to be averaged in Equation equation5.

The Shapley value solution concept was developed in cooperative game theory and ascribes contributions to a payoff value earned by a set of players in a game to the individual players. As is implied by Equation equation5, Shapley values ascribe values to players by computing the incremental contribution to the payoff each player makes in all possible permutations of sequences for including players in the game (Shapley, 1953). Specifically, players are added to the game sequentially and, each time a new player is added, the change to the payoff is computed and ascribed to that player. Across all permutations of player inclusion precedence sequences, the incremental payoffs for a player are averaged and this average constitutes the Shapley value for that player. General dominance statistics map to Shapley values by considering IVs as players, the linear regression as the game they play, and the ΔR^2 as the payoff. Equation equation5 is also equivalent to the metric proposed by Lindeman, Marenda, and Gold (1980; as cited in Budescu, 1993) from which the general dominance statistic was originally derived.

As was mentioned in the introduction, one problem with the use of Shapley values and DA for relative importance is that these methods tend to require large numbers of ΔR^2 values corresponding with different subsets of IVs. This is because C_α statistics require ΔR^2 values from all $|\mathcal{P}(\mathbf{X})| = 2^{|\mathbf{X}|}$ combinations

of IVs in $\mathbf{X}$. I also mentioned in the introduction that a practical solution to this problem is to use the grouped DA method and put the IVs in $\mathbf{X}$ into IV groups where members of an IV group are inseparable.

3 Grouping Independent Variables

IVs can be put into inseparable groups prior to generating ΔR^2 values for use in Shapley values and DA. In some cases, IV groups are formed from collections of IVs with a conceptual similarity that the researcher believes are more valuable to discuss as a conceptual category than as individual IVs (see Gu, 2023, for an example). In other cases, IV groups are formed to reduce the number of IV combinations that are required to compute Shapley values or DA statistics (e.g., Bittmann, 2024; Luchman, 2021).

I define a set of IV groups as $\mathbf{G}$ which include all IVs in $\mathbf{X}$ such that $|\mathbf{G}| > 1$ or there must be at least two IV groups. In addition, $\alpha \in \Gamma_\alpha \in \mathbf{G}$. This means that IV α is a member of IV group Γ_α which is also a member of the IV groups in $\mathbf{G}$. $\mathbf{G}$ can then be substituted for $\mathbf{X}$ and Γ_α can be substituted for α in all places in Equation equation5 to compute general dominance statistics/Shapley values on the IV groups as in Equation equation6:

$$C_{\Gamma_\alpha} = \sum_{j=1}^{|\mathbf{G}|!} \frac{\Delta R^2_{\tilde{\mathbf{G}}_j}}{|\mathbf{G}|!}. \tag{6}$$

where $\tilde{\mathbf{G}}_j = (\tilde{\mathbf{G}}_>^{\setminus\Gamma_\alpha}, \alpha, \tilde{\mathbf{G}}_<^{\setminus\Gamma_\alpha})_j \in \mathsf{Sym}(\mathbf{G})$. The sub-sequence $\tilde{\mathbf{G}}_>^{\setminus\Gamma_\alpha}$ is a permutation of the IVs groups in $\mathbf{G} \setminus \Gamma_\alpha$ that have higher inclusion precedence positions than Γ_α in sequence j. Similarly, the sub-sequence $\tilde{\mathbf{G}}_<^{\setminus\Gamma_\alpha}$ is a permutation of the IVs groups in $\mathbf{G} \setminus \Gamma_\alpha$ that have lower inclusion precedence positions than Γ_α in sequence j. The value of $\Delta R^2_{\tilde{\mathbf{G}}_j}$ is computed in a way identical to that of $\Delta R^2_{\tilde{\mathbf{X}}_i}$ by focusing on the increment that Γ_α makes when included in the model following inclusion of all IV groups in $\tilde{\mathbf{G}}_>^{\setminus\Gamma_\alpha}$.

I noted in the introduction that the grouped DA methodology imposes the strong constraint that it cannot distinguish between individual IVs within IV groups. This is because each IV within an IV group has the same inclusion precedence position as all the other members of their IV group. As a result, distinctions between IVs within an IV group are not possible as individual IV contributions to the ΔR^2 are pooled. Although grouped DA statistics/Shapley values cannot distinguish IV contributions within an IV group, there exist other methods which can disentangle contributions an IV makes from their IV group.

4 Owen Values

Owen values are an extension of Shapley values in that allow players to "union-ize" and pool their impact in terms of how they are ascribed components of

the payoff. Owen values then allow players to join together when being ascribed parts of a payoff across player unions yet still obtain individual, within-union payoff values (Owen, 1977). The method to ascribe components of the payoff to unions/groups, and then to individuals within those unions/groups, results in a two-step approach that extends on Equation equation6.

Owen values begin by computing the Shapley values for all IV groups in $\mathbf{G}$ but add an additional pseudo-Shapley value like approach within an IV group, Γ_α. This second pseudo-Shapley value step holds the inclusion precedence positions of other IV groups constant but considers all inclusion precedence positions of IVs within Γ_α. Using this two step-approach, Owen values are able to first ascribe a component of the R^2 to all the IV groups in $\mathbf{G}$ and then, subsequently, are able to ascribe a sub-component of the R^2 associated with an IV group to each of the individual IVs within the IV group.

Owen values, W_α, are computed as in Equation equation7:

$$W_\alpha = \sum_{i=1}^{grp\_perm(\mathbf{G})} \frac{\Delta R^2_{\tilde{\mathbf{S}}_i}}{grp\_perm(\mathbf{G})}, \tag{7}$$

where $\tilde{\mathbf{S}}_i = (\tilde{\mathbf{G}}_{>}^{\setminus\Gamma_\alpha}, \tilde{\Gamma}_{\alpha>}^{\setminus\alpha}, \alpha, \tilde{\Gamma}_{\alpha<}^{\setminus\alpha}, \tilde{\mathbf{G}}_{<}^{\setminus\Gamma_\alpha})_i \in \mathsf{Sym}(\mathbf{X}) : \mathbf{G}$. A sequence $\tilde{\mathbf{S}}_i$ will then include all the IVs in $\mathbf{X}$ but will require that IVs in the same IV group are contiguous. The sub-sequence $\tilde{\Gamma}_{\alpha>}^{\setminus\alpha}$ includes the IVs in $\Gamma_\alpha \setminus \alpha$ with higher inclusion precedence positions than α in sequence i. The sub-sequence $\tilde{\Gamma}_{\alpha<}^{\setminus\alpha}$ includes the IVs in $\Gamma_\alpha \setminus \alpha$ with lower inclusion precedence positions than α in sequence i. In addition, $|\mathsf{Sym}(\mathbf{X}) : \mathbf{G}| = grp\_perm(\mathbf{G}) = |\mathbf{G}|! \cdot \prod_{l=1}^{|\mathbf{G}|} |\Gamma_l|!$ where $\Gamma_l \in \mathbf{G}$. The $grp\_perm()$ function then takes a set of IV groups and computes the number of possible permutations of the IVs in $\mathbf{X}$ such that it respects the grouping structure of $\mathbf{G}$. Hence, $grp\_perm(\mathbf{G})$ will compute permutations based on the number of IV groups (i.e., $|\mathbf{G}|!$) and the composition of the individual IV groups (i.e., $\prod_{l=1}^{|\mathbf{G}|} |\Gamma_l|!$). Finally, the value of $\Delta R^2_{\tilde{\mathbf{S}}_i}$ represents the increment to the R^2 that α makes beyond the IV groups in $\tilde{\mathbf{G}}_{>}^{\setminus\Gamma_\alpha}$ and the other IVs in $\tilde{\Gamma}_{\alpha>}^{\setminus\alpha}$.

Note the similarity in the formulation of Owen values in Equation equation7 and Shapley values in Equation equation5. Owen values differ from Shapley values only in that they focus on $\mathsf{Sym}(\mathbf{X}) : \mathbf{G}$ instead of $\mathsf{Sym}(\mathbf{X})$ and thus include only a subset of the possible inclusion precedence sequences of IVs in $\mathbf{X}$. The inclusion precedence sequences used by Owen values require that all members of an IV group are contiguous in terms of their inclusion precedence positions.

An implication of the contiguity requirement imposed by Owen values is that all members of one IV group must all be included before members of another IV group are included in the sequence. This leads directly to another implication of the grouping structure; that $|\mathsf{Sym}(\mathbf{X})| > |\mathsf{Sym}(\mathbf{X}) : \mathbf{G}| > |\mathsf{Sym}(\mathbf{G})|$ which is another way of saying that the number of inclusion precedence sequences required for Owen values will always be between the number of inclusion precedence sequences required by traditional Shapley values and grouped Shapley values.

Again, Owen values only use a subset of the possible inclusion precedence sequences used by Shapley values.

Owen values can translate back into DA statistics and this is the focus of the next section. I also note before moving on that an example of Owen decomposition is included in the analytic example section focusing on within-group DA statistics for interested readers.

5 Within-group Dominance Analysis

The statistics and designations for the traditional DA method are derived in a way that extends conceptually from Shapely values and, similarly, the statistics and designations for the *within-group DA method* will be defined in a way that extends conceptually from Owen values. Therefore, a goal of the definition of the within-group DA method will be to preserve the relationships that the statistics and designations of traditional DA have with one another while ensuring that they are conceptually aligned with Owen values.

5.1 Within-group Complete Dominance

The intention of complete dominance is to compare two IVs, α and β, across all possible combinations of subsets of the other IVs to determine whether α or β always produces a larger ΔR^2. I intend to ensure that a definition of within-group complete dominance derives from this requirement yet also respects the IV grouping structure imposed by $\mathbf{G}$ and the contiguity in inclusion precedence sequences requirement that extends from it in Owen values. There is, however, no way to define a version of within-group complete dominance such that both of these constraints are met.

The reason that it is not possible to define a version of within-group complete dominance is that the IV grouping structure could result in two IVs, α and β, being in the same, or different, IV groups. IVs being in the same or in different IV groups results in strong constraints on inclusion precedence sequencing given the Owen value contiguity requirements.

When α and β are in the same IV group, α and β can precede or succeed the other $|\mathbf{G}|-1$ IV groups or the other $|\Gamma_\alpha|-2$ IVs within their group in any given inclusion precedence sequence. It is then possible to compare α and β across any subset of the other $|\mathbf{G}|-1$ IV groups or the other $|\Gamma_\alpha|-2$ IVs in their IV group. Hence IVs in the same IV group could be compared using complete dominance in a way that derives from the traditional method yet respects the contiguity requirement of Owen values.

On the other hand, when α and β are in different IV groups, α and β are never to appear in an inclusion precedence sequence without one of them also appearing with all other members of their respective IV group. In such cases, α and β could precede or succeed the other $|\mathbf{G}|-2$ IV groups, but there are no cases where β could precede or succeed subsets of members of Γ_α as β is only allowed to appear in a sequence with all members of Γ_α, never a subset. Thus,

the contiguity requirement of Owen values makes comparing IVs in different IV groups conceptually impossible in a way that derives from the traditional method.

As an illustration of the contiguity issue when comparing IVs in different IV groups, consider the four example IVs discussed above (i.e., α, β, ξ, and ζ). In the traditional DA methodology, α and β would be compared across the following four subsets of other IVs: $\emptyset$ (i.e., the empty set; compared directly to one another), ξ, ζ, and $\{\xi, \zeta\}$. Imagine I were to group the IVs such that $\Gamma_{\{\alpha,\xi\}} = \{\alpha, \xi\}$ and $\Gamma_{\{\beta,\zeta\}} = \{\beta, \zeta\}$. This grouping structure makes comparing ξ across α and β impossible as β cannot appear with ξ without also including α. Similarly, ζ cannot be compared across α and β as α cannot appear with ζ without also including β. This leaves only the following two subsets, $\emptyset$ and $\{\xi, \zeta\}$, across which α and β can be compared. By not being able to use ξ and ζ separately, the comparisons between α and β are confounded with ξ and ζ which defies the idea underlying complete dominance that α and β should be compared across all the other $|\mathbf{X}| - 2$ IVs in the model.

Given that a within-group complete dominance method does not extend to IVs in different IV groups in a way that respects the contiguity requirement of Owen values, all IVs cannot be compared to one another using complete dominance. As such, there is no within-group complete dominance method.

5.2 Within-group Conditional Dominance

The intention of conditional dominance is to compare averaged ΔR^2 values of α and β across all inclusion precedence positions to determine whether α or β always produces a larger averaged ΔR^2 value. Similar to complete dominance, I intend to ensure that a definition of within-group conditional dominance meets this requirement yet also respects the contiguity requirement of Owen values.

Inclusion Precedence A critical first step in defining within-group conditional dominance is to determine how the concept of inclusion precedence position translates from the traditional method into the within-group method. For the traditional method, inclusion precedence is straightforward in that it corresponds with the inclusion precedence of each IV in $\mathbf{X}$. When translating that idea into Owen values, the IV grouping structure complicates the translation process as the method could consider the inclusion precedence positions of IV groups in $\mathbf{G}$/values of Q or the inclusion precedence positions of individual IVs within an IV group Γ_α/values of $P^{\mathbf{G}}$.

I argue that inclusion precedence for within-group conditional dominance should be based on IV group inclusion precedence values, Q, as they extend in a more natural way from the traditional DA method. IV group inclusion precedence is advantageous as all IVs will be in an IV group and the size of each IV group is irrelevant for IV group inclusion precedence positions as all ΔR^2 values within an IV group inclusion precedence position would be averaged. Thus, there will always be the same number of comparisons, $|\mathbf{G}|$, for within-group conditional dominance.

By contrast, using inclusion precedence positions of IVs within an IV group would be exceedingly complicated due to the possibility that IV groups could be of different sizes. Moreover, it is not clear how the inclusion precedence of IV groups would be incorporated in a way that is conceptually reasonable and similar to the traditional DA method if the focus was on IVs within an IV group. I then again assert that it is advantageous to average over inclusion precedence positions of individual IVs within an IV group and to use IV group inclusion precedence positions as the basis for determining within-group conditional dominance.

Constructing Averages Next, I consider how to average ΔR^2 values by IV group inclusion precedence position in a way that respects the contiguity requirement of Owen values. Doing so requires that I first define a new combination of IV groups and IVs within an IV group, $\mathbf{T}^{\setminus\alpha}_{(Q,j,P^{\mathbf{G}},i)}$.

$\mathbf{T}^{\setminus\alpha}_{(Q,j,P^{\mathbf{G}},i)}$ is one combination of $Q-1$ IV groups, j, from $\mathbf{G} \setminus \Gamma_\alpha$ and one combination of $P^{\mathbf{G}}-1$ IVs, i, from $\Gamma_\alpha \setminus \alpha$. More formally, $\mathbf{T}^{\setminus\alpha}_{(Q,j,P^{\mathbf{G}},i)} = \{\mathbf{G}_j^{\setminus\Gamma_\alpha:Q}, {\Gamma_\alpha}_i^{\setminus\alpha:P^{\mathbf{G}}}\}$ where $\mathbf{G}_j^{\setminus\Gamma_\alpha:Q} \in \mathcal{P}(\mathbf{G}\setminus\Gamma_\alpha) : |\mathbf{G}_j^{\setminus\Gamma_\alpha:Q}| = Q-1$ and ${\Gamma_\alpha}_i^{\setminus\alpha:P^{\mathbf{G}}} \in \mathcal{P}(\Gamma_\alpha \setminus \alpha) : |{\Gamma_\alpha}_i^{\setminus\alpha:P^{\mathbf{G}}}| = P^{\mathbf{G}}-1$.

There will be a total of $B^Q = \frac{(|\mathbf{G}|-1)!}{[Q-1]!([|\mathbf{G}|-1]-[Q-1])!}$ different ways in which other IV groups could precede Γ_α at IV group inclusion precedence position Q. In addition, there will be a total of $|\Gamma_\alpha|$ IV inclusion precedence positions for IVs within Γ_α. Finally, there will be a total of $B^{P^{\mathbf{G}}} = \frac{(|\mathbf{\Gamma}_\alpha|-1)!}{[P^{\mathbf{G}}-1]!([|\mathbf{\Gamma}_\alpha|-1]-[P^{\mathbf{G}}-1])!}$ different ways in which other IVs could precede α at IV inclusion precedence position $P^{\mathbf{G}}$. These three sets of combinations determine a specific number of required ΔR^2 summations for a conditional dominance statistic but not the number of times a specific ΔR^2 value is repeated at a summation value that can be incorporated into a weight similar to the traditional method.

The weight at any given summation for within-group conditional dominance is defined as in Equation equation8:

$$wgt(Q, j, P^{\mathbf{G}}, i) = \frac{(Q-1)! \cdot (P^{\mathbf{G}}-1)! \cdot (|\Gamma_\alpha| - P^{\mathbf{G}})! \cdot (|\mathbf{G}| - Q)!}{(|\mathbf{G}|-1)! \cdot |\Gamma_\alpha|!}. \quad (8)$$

The value of $wgt(Q, j, P^{\mathbf{G}}, i)$ reflects the number of identical ΔR^2 values obtained for different permutations of IV group members of $\mathbf{G}_j^{\setminus\Gamma_\alpha:Q}$ that precede Γ_α (i.e., '$(Q-1)!$'), different permutations of the IV group members of $\mathbf{G}\setminus\mathbf{G}_j^{\setminus\Gamma_\alpha:Q}$ that succeed Γ_α (i.e., '$(|\mathbf{G}| - Q)!$'), different permutations of IV members of ${\Gamma_\alpha}_i^{\setminus\alpha:P^{\mathbf{G}}}$ that precede α (i.e., '$(P^{\mathbf{G}}-1)!$'), and different permutations of IV members of $\Gamma_\alpha \setminus {\Gamma_\alpha}_i^{\setminus\alpha:P^{\mathbf{G}}}$ that succeed α (i.e., '$(|\Gamma_\alpha| - P^{\mathbf{G}})!$').

In the section discussing Owen values, I mentioned that Owen values use $grp\_perm(\mathbf{G}) = |\mathsf{Sym}(\mathbf{X}) : \mathbf{G}|$ different inclusion precedence sequences total. It may come as a surprise then that the denominator of $wgt(Q, j, P^{\mathbf{G}}, i)$ includes

only the number of permutations of members of Γ_α times the number of permutations of the other $|\mathbf{G}| - 1$ IV groups. This result extends from the fact that, at a fixed IV group inclusion precedence position, Q, the number of possible permutations (i.e., the value in the denominator) is $|\mathsf{Sym}(\mathbf{X}) : \{\mathbf{G} : \Gamma_{\alpha Q}\}| = grp\_perm(\mathbf{G} \setminus \Gamma_\alpha) \cdot |\Gamma_\alpha|!$ where $\Gamma_{\alpha Q}$ indicates that IV group Γ_α is at inclusion precedence position Q. However, there would also be $grp\_perm(\mathbf{G}_j^{\setminus \Gamma_\alpha : Q})$ permutations of IV groups and their members that precede Γ_α and $grp\_perm(\mathbf{G} \setminus \mathbf{G}_j^{\setminus \Gamma_\alpha : Q})$ permutations of IV groups and their members that succeed Γ_α in the numerator. The repeated products in the numerator and denominator (i.e., those related to $|\Gamma_l|!$) then cancel, leaving only $(|\mathbf{G}| - 1)! \cdot |\Gamma_\alpha|!$ in the denominator as well as $|\mathbf{G}_j^{\setminus \Gamma_\alpha : Q}| = (Q - 1)!$ and $|\mathbf{G} \setminus \mathbf{G}_j^{\setminus \Gamma_\alpha : Q}| = (|\mathbf{G}| - Q)!$ in the numerator.

Defining the Conditional Dominance Statistic I can now define the *within-group conditional dominance statistic*, W_α^Q, as:

$$W_\alpha^Q = \sum_{j=1}^{B^Q} \sum_{P^{\mathbf{G}}=1}^{|\Gamma_\alpha|} \sum_{i=1}^{B^{P^{\mathbf{G}}}} wgt(Q, j, P^{\mathbf{G}}, i) \cdot R^2_{\mathbf{T}^{\setminus \alpha}_{(Q,j,P^{\mathbf{G}},i)} \cup \alpha}. \tag{9}$$

The within-group conditional dominance statistics determine conditional dominance with Equation equation1 using $K = |\mathbf{G}|$, $V_i^\alpha = W_\alpha^Q$, and $V_i^\beta = W_\beta^Q$.

The W_α^Q values show one interesting property that is worth noting; each value can be summed within its respective IV group by inclusion precedence position to obtain the value of $C_{\Gamma_\alpha}^Q$, or the grouped conditional dominance statistic value for Γ_α at inclusion precedence Q. This is because $C_{\Gamma_\alpha}^Q = \sum_{j=1}^{B^Q} \frac{\Delta R^2_{\mathbf{G}^{\setminus \Gamma_\alpha : Q} \cup \Gamma_\alpha}}{B^Q}$ which has a form similar to Equation equation9. In fact, Equation equation9 is an extension of these grouped conditional dominance statistics that includes additional averaging for IVs within an IV group (i.e., the two other summations and extensions given the $wgt()$ function and $\mathbf{T}$ combination). I show an example of this grouped to within-group DA decomposition property in the analytic example discussed in section six.

5.3 Within-group General Dominance

The intention of general dominance is to compare the averaged conditional dominance statistic values of α and β. I again intend to ensure that a definition of within-group general dominance meets this requirement yet also respects the contiguity requirement of Owen values.

Extending general dominance to adhere to Owen value computations is straightforward and, like the traditional method, involves merely averaging the within-group conditional dominance statistics. I then define the *within-group general dominance statistic*, W_α as:

$$W_\alpha = \sum_{Q=1}^{|\mathbf{G}|} \frac{W_\alpha^Q}{|\mathbf{G}|}. \tag{10}$$

The within-group general dominance statistics determine general dominance with Equation equation1 using $K = 1$, $V_i^{\alpha} = W_{\alpha}$, and $V_i^{\beta} = W_{\beta}$.

The W_{α} values also show an interesting property; each value can be summed within its respective IV group to obtain the value of $C_{\Gamma_{\alpha}}$, or the grouped general dominance statistic value for Γ_{α}. This property of W_{α} values extends directly from the W_{α}^{Q} values on which they are based; the W_{α}^{Q} values decompose $C_{\Gamma_{\alpha}}^{Q}$ and, when averaged, the W_{α} values decompose $C_{\Gamma_{\alpha}}$.

Recall that Owen values also produce a value of the payoff ascribed to a union of players and then subsequently ascribe components of that player union's value to individual players within the union. Hence, Owen values applied to the R^2 in a linear regression also decompose the values $C_{\Gamma_{\alpha}}$ and, as is suggested by their shared notation, the result in Equation equation10 is equivalent to that of Equation equation7. The within-group general dominance statistics are then tantamount to Owen values in the same way that traditional general dominance statistics are tantamount to Shapley values. I also show an example of this grouped to within-group DA decomposition property in the analytic example discussed in section six.

6 Analytic Example

Within-group DA extends on the traditional and grouped DA methodologies by combining aspects of both approaches which produces a set of relative importance determinations that focuses on individual IVs but incorporates information relevant to IV groupings.

This section provides an analytic example which applies the traditional, grouped, and within-group DA methodologies to data. The purpose of this section is to more concretely illustrate the differences between the methods in terms of the amount of information they provide about IVs and how the IV groups affect the information used by the different DA methods.

6.1 Analysis Context

The data used in this example were derived the `ability.cov` covariance matrix from the R package `datasets` (Antal, 2025). These data describe the relationships between a number of different ability and intelligence tests given the data from 112 different test takers. I used the variables *picture*, *blocks*, *reading*, and *vocab* in the example analyses. *picture* is described as a picture-completion test, *blocks* is described as a block design task, *reading* is a reading comprehension test, and *vocab* is a vocabulary test. The IVs were grouped such that *picture* and *blocks* formed one IV group, $\Gamma_{spatial}$, and that *reading* and *vocab* formed another, Γ_{verbal}. As the subscripts to the IV groups suggest, the spatial tests were grouped together and the verbal tests were also grouped together. Finally, the dependent variable was *general* which is described as a non-verbal measure of general intelligence using Cattell's culture-fair test. The covariance matrix provided in the data was transformed into a correlation matrix and the intercorrelations between

Table 1. Correlations between Variables

	general	*picture*	*blocks*	*reading*	*vocab*
general	1.0000	0.4663	0.5517	0.5765	0.5144
picture	0.4663	1.0000	0.5724	0.2629	0.2393
blocks	0.5517	0.5724	1.0000	0.3540	0.3565
reading	0.5765	0.2629	0.3540	1.0000	0.7914
vocab	0.5144	0.2393	0.3565	0.7914	1.0000

all study variables are reported below in Table table1. Consistent with the idea that they might assess similar content within a group, the members of $\Gamma_{spatial}$ correlated fairly strongly as did the members of Γ_{verbal}.

The results from the linear regressions of the 16 IV subsets, which form all possible combinations of the four IVs, is reported in Table table2. Table table2

Table 2. All Combinations of Subsets of Independent Variables and their R^2 Values

Subset Number	IV Subset	R^2
1	*picture blocks reading vocab*	0.4960
2	*blocks reading vocab*	0.4726
3	*picture reading vocab*	0.4448
4	*reading vocab*	0.3414
5	*picture blocks vocab*	0.4482
6	*blocks vocab*	0.4200
7	*picture vocab*	0.3895
8	*vocab*	0.2646
9	*picture blocks reading*	0.4935
10	*blocks reading*	0.4704
11	*picture reading*	0.4387
12	*reading*	0.3323
13	*picture blocks*	0.3380
14	*blocks*	0.3043
15	*picture*	0.2174
16	$\emptyset$	0.0000

shows that the overall model R^2 (i.e., subset 1) is .4960. In addition, the R^2 values associated with *reading* are among the highest and those associated with *picture* are among the lowest. This suggests the possibility that *reading* is the most and *picture* is the least important IV.

6.2 Conditional Dominance Results

Because complete dominance is not achievable using the within-group DA methodology, I began evaluating the relative importance of the IVs using conditional dominance. As an illustration of making conditional dominance designations across the methods, I considered the comparison of *picture* and *reading*. This

comparison was chosen to emphasize that the within-group DA method can compare IVs in different IV groups.

Traditional Method The four conditional dominance statistics for *picture* used the average ΔR^2 values where *picture*'s inclusion precedence position was first, second, third, and fourth in the model. There was only one subset where *picture* was included first, subset 15, and its conditional dominance statistic was the increment it made beyond subset 16. Applying Equation equation2 and representing the P values with ordinal positions (i.e., as the value 1^{st}), resulted in: $C_{picture}^{1^{st}} = \frac{(.2174-.0000)}{1} = .2174$.

The next computation focused on subsets where *picture* was included second. This included using the increment of subsets 7 over 8, 11 over 12, and 13 over 14. When included in Equation equation2 the result was: $C_{picture}^{2^{nd}} = \frac{(.3895-.2646)}{3} + \frac{(.4387-.3323)}{3} + \frac{(.3380-.3043)}{3} = .0883$.

The third computation focused on subsets where *picture* was included third which incorporated the increments of subsets 3 over 4, 5 over 6, and 9 over 10. The conditional dominance computation resulted in: $C_{picture}^{3^{rd}} = \frac{(.4448-.3414)}{3} + \frac{(.4482-.4200)}{3} + \frac{(.4935-.4704)}{3} = .0516$.

The fourth computation focused on subsets where *picture* was included last or fourth. Similar to when it was included first, there was only one increment of subset 1 over 2 which produced: $C_{picture}^{4^{th}} = \frac{(.4960-.4726)}{1} = .0234$.

The four conditional dominance statistics for *reading* also used the average ΔR^2 values where its inclusion precedence position was first, second, third, and fourth in the model. When in the first inclusion precedence position, the relevant increment for *picture* was the increment of subset 12 over 16. The conditional dominance statistic result was: $C_{reading}^{1^{st}} = \frac{(.3323-.0000)}{1} = .3323$.

When *reading* was included as the second IV, the relevant subsets included the increments of subsets 4 over 8, 10 over 14, and 11 over 15. Applying those three increments in Equation equation2 produced: $C_{reading}^{2^{nd}} = \frac{(.3414-.2646)}{3} + \frac{(.4704-.3043)}{3} + \frac{(.4387-.2174)}{3} = .1547$.

As the the variable included third, the relevant increments for *reading* were subsets 2 over 6, 3 over 7, and 9 over 13. These three increments resulted in: $C_{reading}^{3^{rd}} = \frac{(.4726-.4200)}{3} + \frac{(.4448-.3895)}{3} + \frac{(.4935-.3380)}{3} = .0878$.

Lastly, when *reading* was included last or fourth, the relevant increment was subset 1 over 5. The resulting conditional dominance statistic was: $C_{reading}^{4^{th}} = \frac{(.4960-.4482)}{1} = .0477$.

With all eight conditional dominance statistics, I then applied Equation equation1 to determine which IV conditionally dominated the other. The pattern of results showed that $C_{reading}^{1^{st}} > C_{picture}^{1^{st}}$, $C_{reading}^{2^{nd}} > C_{picture}^{2^{nd}}$, $C_{reading}^{3^{rd}} > C_{picture}^{3^{rd}}$, and $C_{reading}^{4^{th}} > C_{picture}^{4^{th}}$ which meant *reading* conditionally dominated *picture*.

Notice that the computation of conditional dominance statistics for both IVs required the use of ΔR^2 values from all 16 IV subsets reported on in Table table2. This is the reason that the traditional method is computationally expensive as it

uses the ΔR^2 values from all possible combinations of IVs. Because *picture* was in $\Gamma_{spatial}$ and *reading* was in Γ_{verbal}, they could also be compared indirectly using the grouped method. The grouped method illustrated next required the use of many fewer of the ΔR^2 values in Table table2 but did not allow me to distinguish the predictive utility of *picture* from *blocks* or the predictive utility of *reading* from *vocab*.

Grouped Method The two conditional dominance statistics for $\Gamma_{spatial}$ used the average ΔR^2 values where $\Gamma_{spatial}$'s inclusion precedence position was first and second in the model. There was only one increment where $\Gamma_{spatial}$ was included first in the model, the increment of subset 13 (i.e., $\Gamma_{spatial}$ which includes *picture* and *blocks*) over 16 (i.e., no IV groups). Applying Equation equation2 produced $C^{1^{st}}_{\Gamma_{spatial}} = \frac{(.3380-.0000)}{1} = .3380$.

When $\Gamma_{spatial}$ was included as the last IV group, there was also only one relevant increment of subset 1 (i.e., all IVs and hence both IV groups) over 4 (i.e., Γ_{verbal} which includes *reading* and *vocab*). This produced a conditional dominance statistic value of $C^{2^{nd}}_{\Gamma_{spatial}} = \frac{(.4960-.3414)}{1} = .1546$.

The two conditional dominance statistics for Γ_{verbal} used the average ΔR^2 values where Γ_{verbal}'s inclusion precedence position was first and second in the model. When Γ_{verbal} was included as the first IV group, its conditional dominance statistic was comprised of the increment of subset 4 over 16 or $C^{1^{st}}_{\Gamma_{verbal}} = \frac{(.3414-.0000)}{1} = .3414$.

When Γ_{verbal} was included as the last IV group, its conditional dominance statistic was comprised of the increment of subset 1 over 13 or $C^{2^{nd}}_{\Gamma_{verbal}} = \frac{(.4960-.3380)}{1} = .1580$.

These four conditional dominance statistics show that $C^{1^{st}}_{\Gamma_{verbal}} > C^{1^{st}}_{\Gamma_{spatial}}$ and $C^{2^{nd}}_{\Gamma_{verbal}} > C^{2^{nd}}_{\Gamma_{spatial}}$ when applying Equation equation1 which meant Γ_{verbal} conditionally dominated $\Gamma_{spatial}$.

The grouped DA methodology used substantially fewer IV subsets (i.e., $2^2 = 4$) than the traditional method but was not able to disentangle *blocks* from *picture* and *vocab* from *reading* in the comparisons. The comparison between *picture* and *reading* was then confounded with the other two IVs in the model. The within-group method illustrated next is a balance of the traditional and grouped DA methods that requires the use of fewer IV subsets than the traditional method yet allows a researcher to determine importance between all IVs in the model like the traditional method.

Within-group Method The two conditional dominance statistics for *picture* used the average ΔR^2 values where $\Gamma_{spatial}$'s inclusion precedence position was first and second in the model but also averaged over subsets where *picture* preceded and succeeded *blocks* within $\Gamma_{spatial}$.

When $\Gamma_{spatial}$ was included first, there were two relevant increments: subset 11 over 12 (i.e., *picture* succeeded *blocks*) and subset 15 over 16 (i.e., *picture* pre-

ceded *blocks*). Applying Equation equation9 to those values produced $W^{1^{st}}_{picture} = \frac{(.4387-.3323)}{2} + \frac{(.2174-.0000)}{2} = .1255$.

When $\Gamma_{spatial}$ was included last, there were also two relevant increments: subset 1 over 2 (i.e., *picture* succeeded *blocks*) and subset 5 over 6 (i.e., *picture* preceded *blocks*). These two values produced a conditional dominance statistic of $W^{2^{nd}}_{picture} = \frac{(.4960-.4726)}{2} + \frac{(.4482-.4200)}{2} = .0634$.

The two conditional dominance statistics for *reading* also included averages where Γ_{verbal} was included first and second in the model averaging over subsets where *reading* preceded or succeeded *vocab*. When Γ_{verbal} was included first, there were two relevant increments: subset 11 over 15 and subset 12 over 16. The conditional dominance statistic produced by these increments was $W^{1^{st}}_{reading} = \frac{(.4387-.2174)}{2} + \frac{(.3323-.0000)}{2} = .2046$.

When Γ_{verbal} was included last, there were also two relevant increments: subset 1 over 5 and subset 2 over 6. These final increments resulted in a value of $W^{2^{nd}}_{reading} = \frac{(.4960-.4482)}{2} + \frac{(.4726-.4200)}{2} = .1016$.

The four conditional dominance statistics, when applying Equation equation1, resulted in $W^{1^{st}}_{reading} > W^{1^{st}}_{picture}$ and $W^{2^{nd}}_{reading} > W^{2^{nd}}_{picture}$ which meant that *reading* conditionally dominated *picture*.

The within-group method used more IV subsets than the grouped method but fewer than the traditional method as it required only $2^2 + (2^{(2-1)}) * (2^2 - 2) + (2^{(2-1)}) * (2^2 - 2) = 12$ IV subsets to compute conditional dominance statistics. Again, the within-group DA serves as a balance between the traditional and grouped methods that allows determinations like those possible with the traditional method but reduces the number of required IV subsets like the grouped method. Ultimately in the case of this example, the within-group DA methodology resulted in a materially similar conclusion as the traditional method but used 25% fewer IV subsets.

Summary The conditional dominance statistics for all IVs and IV groups across the traditional, grouped, and within-group methods are reported in Table table3.

The results in Table table3 are reported such that each IV's results appear in the rows and the inclusion precedence position of the conditional dominance statistics appear in the columns. The conditional dominance results in Table table3 showed that *reading* conditionally dominated *picture* and *vocab*, but not *blocks*, for both the traditional and within-group methods. The results also showed that *blocks* conditionally dominated both *picture* and *vocab* for the traditional and within-group methods.

Recall I mentioned that a property of the within-group conditional dominance statistics is that they would sum to the grouped conditional dominance statistics for their IV group. This property was true of the results in Table table3 as $W^{1^{st}}_{picture} + W^{1^{st}}_{blocks} = C^{1^{st}}_{\Gamma_{spatial}}$ or $.1255 + .2125 = .3380$ and $W^{2^{nd}}_{picture} + W^{2^{nd}}_{blocks} = C^{2^{nd}}_{\Gamma_{spatial}}$ or $.0634 + .0912 = .1546$.

Table 3. Conditional Dominance Results

	1^{st}	2^{nd}	3^{rd}	4^{th}
Traditional				
picture	.2174	.0883	.0516	.0234
blocks	.3043	.1380	.0816	.0512
reading	.3323	.1547	.0878	.0477
vocab	.2646	.0990	.0395	.0024
Grouped				
$\Gamma_{spatial}$	.3380	.1546		
Γ_{verbal}	.3414	.1580		
Within-group				
picture	.1255	.0634		
blocks	.2125	.0912		
reading	.2046	.1016		
vocab	.1368	.0563		

Note. $\Gamma_{spatial} = \{picture, blocks\}$
$\Gamma_{verbal} = \{reading, vocab\}$.

6.3 General Dominance Results

Conditional dominance could not be determined for two of the between IV group comparisons: *reading* versus *blocks* and *vocab* versus *picture*. I then proceeded to compare the IVs using general dominance. The focus of the example computations below was on comparing *reading* and *blocks*.

Traditional Method General dominance statistics are always computed as the average of the conditional dominance statistics as is shown in Equation equation3. The value for *blocks* was then computed as $C_{blocks} = \frac{C_{blocks}^{1^{st}}}{4} + \frac{C_{blocks}^{2^{nd}}}{4} + \frac{C_{blocks}^{3^{rd}}}{4} + \frac{C_{blocks}^{4^{th}}}{4} = \frac{.3043}{4} + \frac{.1380}{4} + \frac{.0816}{4} + \frac{.0512}{4} = .1438$.

In addition, the value for *reading* was computed as $C_{reading} = \frac{C_{reading}^{1^{st}}}{4} + \frac{C_{reading}^{2^{nd}}}{4} + \frac{C_{reading}^{3^{rd}}}{4} + \frac{C_{reading}^{4^{th}}}{4} = \frac{.3323}{4} + \frac{.1547}{4} + \frac{.0878}{4} + \frac{.0477}{4} = .1556$.

The comparison between the two IVs resulted in $C_{reading} > C_{blocks}$ and thus *reading* generally dominated *blocks*.

Grouped Method I already know that Γ_{verbal}, which contained *reading*, conditionally dominated $\Gamma_{spatial}$, which contained *blocks*. It must then also have been the case that Γ_{verbal} would generally dominate $\Gamma_{spatial}$. Indeed, when computed, I found that this was the case as $C_{\Gamma_{spatial}} = \frac{C_{\Gamma_{spatial}}^{1^{st}}}{2} + \frac{C_{\Gamma_{spatial}}^{2^{nd}}}{2} = \frac{.3380}{2} + \frac{.1546}{2} = .2463$ and $C_{\Gamma_{verbal}} = \frac{C_{\Gamma_{verbal}}^{1^{st}}}{2} + \frac{C_{\Gamma_{verbal}}^{2^{nd}}}{2} = \frac{.3414}{2} + \frac{.1580}{2} = .2497$. Thus, $C_{verbal} > C_{spatial}$ which demonstrated that the expected relationship held.

Within-group Method Equation equation10 focuses on the computation of within-group general dominance statistics but is, like the traditional method, also a simple average of conditional dominance statistics. Thus, the value for *blocks* was $W_{blocks} = \frac{W_{blocks}^{1^{st}}}{2} + \frac{W_{blocks}^{2^{nd}}}{2} = \frac{.2125}{2} + \frac{.0634}{2} = .1518$ and the value for *reading* was $W_{reading} = \frac{W_{reading}^{1^{st}}}{2} + \frac{W_{reading}^{2^{nd}}}{2} = \frac{.2046}{2} + \frac{.1016}{2} = .1531$. This resulted in $W_{reading} > W_{blocks}$, indicating that *reading* generally dominated *blocks*.

Summary The general dominance statistics computed for the traditional, grouped, and within-group methods are reported in Table table4.

Table 4. General Dominance Results

		Traditional	Grouped	Within-group
$\Gamma_{spatial}$	*picture*	.0952	.2463	.0945
	blocks	.1438		.1518
Γ_{verbal}	*reading*	.1556	.2497	.1531
	vocab	.1014		.0966

The results in Table table4 showed that, in addition to *reading* having generally dominated *blocks*, *vocab* generally dominated *picture*. Hence, all pairs of IVs were ranked with their respective dominance designations indicating differences in strength of evidence. Specifically, these results indicated that *reading* was the most important IV, followed by *blocks*, then *vocab*, and finally *picture*.

Note again I mentioned that the within-group general dominance statistics were tantamount to Owen values. Thus, the within-group general dominance statistics for members of an IV group summed to the grouped general dominance statistic for their IV group. The results in Table table4 also illustrated this idea as, for instance, $C_{\Gamma_{spatial}} = W_{picture} + W_{blocks}$ or $.2463 = .0945 + .1518$.

7 Discussion

In this work, I discussed the link between DA and Shapley values focusing specifically on how DA/Shapley values allow for determining importance with IVs in a statistical model like linear regression. I also outlined how the traditional DA/Shapley value methodology can result in large number of combinations of subsets of IVs to produce dominance determinations and how the grouped IV DA methodology can reduce the number of subsets of IVs. I then introduced Owen values as a method for decomposing the grouped DA/Shapley values. Extending from Owen values, I devised within-group conditional and within-group general dominance statistics that use IV grouping information to eliminate IV subset combinations yet allow for importance determinations between individual IVs.

Following the definition of the within-group DA methodology, I also provided an analytic example based on the `ability.cov` dataset in the R statistical computing environment that illustrates how the within-group DA methodology compares to the traditional and grouped methodologies. The proposed within-group DA methodology is intended to be a useful tool for practicing researchers who are seeking importance determinations for IVs when there are IV groups that could be formed from the IVs. The sections below elaborate further on within-group DA, recommendations on how to apply the methodology, and discuss future directions for this line of research.

7.1 Eliminating Subsets

The primary advantage of using the within-group DA methodology is that it can improve the efficiency of DA by obtaining relative importance determinations between IVs while eliminating a, sometimes substantial, number of IV subsets. Recall that for the analytic example with four IVs, the required number of subsets of IVs to estimate all dominance statistics was 16 using traditional DA. The within-group version of DA needed fewer IV subsets as any IV subset that included a combination of IVs where there were multiple incomplete IV groups were eliminated. This resulted in the need for only 12 subsets of IVs for the within-group method—eliminating 25% of IV subsets needed for the traditional approach.

To determine the number of IV subsets that will be required using the within-group DA methodology first recall that, for traditional DA, the number of IV subsets required is $2^{|\mathbf{X}|}$ or all combinations of the IVs. Within-group DA is, however, more similar to grouped DA for its IV subset requirements. Grouped DA requires $2^{|\mathbf{G}|}$ IV group subsets. Within-group DA expands on the IV group subsets by including all combinations of IV subsets within each IV group. The number of required IV subsets for within-group DA is given in Equation equation11:

$$2^{|\mathbf{G}|} + \sum_{l=1}^{|\mathbf{G}|} (2^{|\mathbf{G}|-1}) \cdot (2^{|\Gamma_l|} - 2). \tag{11}$$

Note that the $2^{|\mathbf{G}|-1}$ term will include all combinations of the $\mathbf{G}$ IV groups not including IV group Γ_l. In addition, the $2^{|\Gamma_l|}-2$ term subtracts the two IV subsets where all and none of the IVs in IV group Γ_l are included as those IV subsets are included in the $2^{|\mathbf{G}|}$ term.

Excluding IV subsets by applying within-group DA could allow for conducting DA with a much larger set of IVs than would be computationally feasible with the traditional methodology. Traditional DA, even with modern computing power, can require a considerable time investment and computational resources to analyze 20 IVs or more in a model. Twenty IVs in a traditional DA would require $2^{20} = 1{,}048{,}576$ IV subsets to get dominance statistics and designations. If these 20 IVs were to be grouped into two groups of three and one group of four, the number of IV subsets is reduced to $2^3 + 2^{3-1} \cdot (2^3 - 2) + 2^{3-1} \cdot (2^3 - 2) + 2^{3-1} \cdot (2^4 - 2) = 112$ IV subsets which is about .01% of those required by the

traditional methodology. Thus, within-group DA can be structured such that the number of required IV subsets is far lower than that of traditional DA.

That within-group DA can substantially reduce the number of IV subsets required for a determining the relative importance of IVs is an important practical advantage of this methodology and could help to buffer against one of the biggest limitations of the DA methodology; that the method tends to become computationally infeasible with more IVs (e.g., Johnsonx& LeBreton, 2004).

7.2 Practical Considerations for Grouping IVs

A complication for practicing researchers in applying the within-group DA method could be in ascertaining how to group IVs. My recommendation for grouping IVs is to do so using conceptual categories when possible. The use of conceptual categories is advantageous for grouping IVs as they will ensure that IVs which are more strongly related conceptually are nearer one another in terms of inclusion precedence sequences and IV subsets which more strongly affects how the DA statistics separate out the variance they explain in the dependent variable. Indeed, this was the approach used for creating the $\Gamma_{spatial}$ and Γ_{verbal} IV groups in the analytic example.

When it is not possible to group IVs using conceptual categories, grouping IVs based on their shared variance is a useful and practical alternative. I recommend this approach as IVs that are more strongly correlated affect variance partitioning for one another more than less correlated IVs. Moreover, IVs in the same IV group will be nearer one another in inclusion precedence sequences and IV subsets which more strongly affects how the variance they explain in the dependent variable is ascribed. Hence, putting IVs that are more strongly correlated into the same group uses the most critical information about IV overlap to partition the R^2. I then recommend that IVs that are more strongly correlated with one another are placed into an IV group together in the absence of conceptual categories. It is also worth noting that the correlation method would have led to grouping *picture* and *blocks* as well as *reading* and *vocab* into separate IV groups even if they were not as strongly aligned conceptually (i.e., see the results in Table table1).

7.3 Limitations and Future Directions

Randomizing IV Groups An interesting future direction for research on within-group DA would be to offer researchers alternatives when no conceptual or correlation-based IV grouping is reasonable. One possible direction for exploration is to examine how randomly assigning IVs to IV groups might affect the conclusions reached by within-group DA compared to traditional DA. A random assignment strategy might be useful in cases where there are many IVs and no clear patterns of interrelationships between the IVs. In such cases, it would be sensible to evaluate more than one random assignment to ensure that some IV assignments, by chance, do not eliminate IV subsets that mask crucial importance results.

As an example, consider a model with 30 IVs and no good conceptual groupings for the IVs. This model produces an astronomical 1.07 billion subsets for the traditional DA methodology. By contrast, when grouping the IVs into six groups of five IVs, this set of 30 IVs produces a much more reasonable 5,824 subsets. Given the smaller number of subsets, the researcher could choose multiple random assignments of IVs to IV groups and evaluate how the different random IV group assignments affect dominance designation results. For instance, the researcher could choose 30 different random groupings of the IVs as a test to ensure that the way in which the IVs are put into groups does not affect the conclusions. This set of 30 IV groupings would result in $5{,}824 \cdot 30 = 174{,}720$ subsets which is still far fewer than would be needed for the traditional DA methodology yet would be similar to it in that no predetermined IV conceptual groupings would be necessary.

Subgroups and Fixed Sequences At current, the within-group DA methodology allows for IVs to be grouped together which affects the number of valid IV subsets. It is conceivable that further grouping would be possible such that there are IV subgroups within an IV group that function in a way similar to how the IV group works in the context of the other IV groups. IV subgroups within an IV group would also eliminate IV subsets among the members of an IV group in a way similar to how IV groups eliminate IV subsets overall.

For example, a researcher with eight IVs would need 256 (i.e., $2^8 = 256$) IV subsets for the traditional DA method. If this researcher grouped the eight IVs into two groups of four, the within-group method would require 60 (i.e., $2^2 + 2 \cdot 2^{2-1} \cdot (2^4 - 2)$) IV subsets. Consider now whether this researcher further grouped their IVs into subgroups of size two within each IV group. This would require 44 (i.e., $2^2 + 2 \cdot 2^{2-1} \cdot (2^2 - 2) + 4 \cdot 2^{2-1} \cdot 2^{2-1} \cdot (2^2 - 2)$) IV subsets. The IV subgrouping then required around 6 times fewer (i.e., $\frac{256}{44}$) IV subsets compared to the traditional methodology. The IV subgrouping also reduced the number of subsets required by around 36% (i.e., $\frac{60}{44}$) compared to the single level of IV grouping.

It is also conceivable that a researcher would want an IV or IV group to be constrained such that it always precedes or succeeds one or more a counterpart IVs or IV groups. It would not be necessary in these cases that the focal IV or IV group immediately precede or succeed the counterparts in the sense that they must be contiguous. Rather the constraint described here would merely eliminate all IV subsets which imply that the focal IV or IV group is somewhere before (when it must succeed) or somewhere after (when it must precede) the counterparts they are constraint to, or not to, follow. For example, a researcher might want to require that an interaction term always succeeds its constituent IVs to produce a valid result. A methodology such as this could be a useful alternative to the current best practice in the DA literature for the relative importance analysis of interactions which involves the residualization of the constituent IVs (LeBreton, Tonidandel,x& Krasikova, 2013).

Bootstrapping and Sampling Variability Best practices in the literature suggest bootstrapping dominance designations to understand the impact of sampling variability on their reproducibility (Azenx& Budescu, 2003). An additional advantage of using the within-group methodology is the improved computational feasibility for obtaining bootstrapped estimates of DA designation reproducibility for general and conditional dominance designations.

For example, estimating reproducibility from a model with six IVs and 100 bootstrap replications for a traditional DA would require $2^6 \cdot 100 = 6{,}400$ or 64 subsets over 100 bootstrap samples to be estimated in total. If the six IVs were grouped into two groups of three, the number of models is reduced to $(2^2 + 2 \cdot 2^{2-1} \cdot [2^3 - 2]) \cdot 100 = 2{,}800$ or 28 subsets with 100 bootstrap samples—less than half of the number required for the traditional methodology.

8 Conclusion

The within-group DA methodology developed in this work extends on traditional DA by discussing its foundation in Shapely values and by devising the within-group method such that it derives from Owen values, a similar methodology that accommodates player unions yet still produces payoff estimates for individual players. The within-group DA method is valuable to research practice as it improves the computational feasibility of DA as the number of IVs in a model increases and only requires the researcher to generate mutually exclusive groups of IVs in their model.

Before concluding, I note that traditional DA remains a valuable tool for the evaluation of relative importance with statistical models where the number of IVs is relatively small or the researcher cannot group IVs. I also note that the methodology discussed in this manuscript will be implemented using the `domir` function in the package `domir` (Luchman, 2024) available in the R statistical computing environment which also includes methods to compute the traditional and grouped methodologies.

References

Antal, D. (2025). dataset: Create data frames for exchange and reuse [Computer software manual]. (R package version 0.4.1) doi: https://doi.org/10.32614/CRAN.package.dataset

Azen, R., & Budescu, D. V. (2003). The dominance analysis approach for comparing predictors in multiple regression. *Psychological Methods*, *8*(2), 129–148. doi: https://doi.org/10.1037/1082-989X.8.2.129

Bittmann, F. (2024). A primer on dominance analysis. doi: https://doi.org/10.20944/preprints202404.1606.v1

Budescu, D. V. (1993). Dominance analysis: A new approach to the problem of relative importance of predictors in multiple regression. *Psychological Bulletin*, *114*(3), 542–551. doi: https://doi.org/10.1037/0033-2909.114.3.542

Budescu, D. V., & Azen, R. (2004). Beyond global measures of relative importance: Some insights from dominance analysis. *Organizational Research Methods*, *7*(3), 341–350. doi: https://doi.org/10.1177/1094428104267049

Grömping, U. (2007). Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician*, *61*(2), 139–147. doi: https://doi.org/10.1198/000313007X188252

Gu, X. (2023). Evaluating predictors' relative importance using bayes factors in regression models. *Psychological Methods*, *28*(4), 825–842. doi: https://doi.org/10.1037/met0000431

Johnson, J. W., & LeBreton, J. M. (2004). History and use of relative importance indices in organizational research. *Organizational Research Methods*, *7*(3), 238–257. doi: https://doi.org/10.1177/109442810426651

Kruskal, W. (1987). Relative importance by averaging over orderings. *The American Statistician*, *41*(1), 6–10. doi: https://doi.org/10.1080/00031305.1987.10475432

LeBreton, J. M., Tonidandel, S., & Krasikova, D. V. (2013). Residualized relative importance analysis: A technique for the comprehensive decomposition of variance in higher order regression models. *Organizational Research Methods*, *16*(3), 449–473. doi: https://doi.org/10.1177/1094428113481065

Luchman, J. N. (2021). Determining relative importance in stata using dominance analysis: domin and domme. *The Stata Journal*, *21*(2), 510–538. doi: https://doi.org/10.1177/1536867X211025837

Luchman, J. N. (2024). domir: Tools to support relative importance analysis [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=domir` (R package version 1.2.0, https://jluchman.github.io/domir/)

McLaurin, F. A., West, S. J., & Thomson, N. D. (2025). Exploring the relationship between facets of childhood trauma and violent injury risk during adulthood: A dominance analysis study. *Child Abuse & Neglect*, *161*, 107307. doi: https://doi.org/10.1016/j.chiabu.2025.107307

Miller, B. K., Kirby, E. G., & Stevens, K. B. (2025). Dominance analysis of bright and dark dispositional predictors of socially desirable responding. *Psychological Reports*, *128*(6), 4799–4819. doi: https://doi.org/10.1177/00332941241226908

Owen, G. (1977). Values of games with a priori unions. In *Essays in mathematical economics and game theory* (pp. 76–88). Springer.

Shapley, L. S. (1953). A value for n-person games. In *Contributions to the theory of games II* (pp. 307–317). Princeton University Press.

Thomas, D. R., Zumbo, B. D., Kwan, E., & Schweitzer, L. (2014). On Johnson's (2000) relative weights method for assessing variable importance: A reanalysis. *Multivariate Behavioral Research*, *49*(4), 329–338. doi: https://doi.org/10.1080/00273171.2014.905766

Tonidandel, S., & LeBreton, J. M. (2011). Relative importance analysis: A useful supplement to regression analysis. *Journal of Business and Psychology*, *26*(1), 1–9.

Yin, K., & Zhou, L. (2025). The relative importance of peace of mind, grit, and classroom environment in predicting willingness to communicate among learners in multi-ethnic regions: a latent dominance analysis. *BMC Psychology*, *13*(1), 1–17. doi: https://doi.org/10.1186/s40359-025-02676-2

Journal of Behavioral Data Science, 2026, *6* (1), 68–135.
DOI: https://doi.org/10.35566/jbds/gao

Detecting and Evaluating Bias in Large Language Models: Concepts, Methods, and Challenges

Zu Gao[1][0009−0006−2152−5561], Lingbo Tong[2] and Zhiyong Zhang[3][0000−0003−0590−2196]

[1] University of Oxford, Wellington Square, Oxford OX1 2JD, UK
zu.gao@wadham.ox.ac.uk, alvingz@163.com
[2] University of Wisconsin, Madison
lingbo.tong@wisc.edu
[3] University of Notre Dame, Notre Dame, IN 46530, USA
zzhang4@nd.edu

Abstract. Large Language Models (LLMs) are increasingly deployed in sensitive real-world contexts, yet concerns remain about their biases and the harms they can cause. Existing surveys mostly discuss sources of bias and mitigation techniques, but give less systematic attention to how bias in LLMs should be detected, measured, and reported. This survey addresses that gap. We present a structured review of methods for detecting and evaluating bias in LLMs. We first introduce the conceptual foundations, including representational versus allocational harms and taxonomies of bias. We then discuss how to design evaluations in practice: specifying measurement targets, choosing datasets and metrics, and reasoning about validity and reliability. Building on this, we review intrinsic methods that probe representations and likelihoods, and extrinsic methods that assess bias in classification, question answering, open-ended generation, and dialogue. We further highlight recent advances in counterfactual and certification-based evaluation, which aim to provide stronger guarantees on fairness metrics. Beyond English-centric settings, we survey cross-lingual and application-specific evaluations, intersectional bias analysis, and meta-level issues such as evaluator reliability, metric robustness, reproducibility, and governance. The review concludes by synthesizing best practices and offering a practitioner-oriented checklist, providing both a conceptual map and a practical toolkit for evaluating bias in LLMs.

Keywords: Large Language Models · Bias Evaluation · Fairness in NLP · Certification-based Methods · Reproducibility.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across many natural language processing tasks, but their biases–reflecting societal prejudices present in training corpora–have become a pressing concern (Blodgett, Barocas, Daumé III, & Wallach, 2020; Mehrabi, Morstatter, Saxena, Lerman, & Galstyan, 2021). These biases can manifest as stereotypes and discriminatory associations, leading to representational harms (reinforcing negative portrayals of social groups) and allocational harms (unequal treatment in resource distribution) (Barocas & Selbst, 2016). Such harms are not merely theoretical: for instance, embeddings have been shown to associate occupations with gender stereotypes (Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016; Caliskan, Bryson, & Narayanan, 2017), and toxicity classifiers often over-flag dialectal text such as African American Vernacular English (Hanu & Unitary team, 2020). Therefore, it is important to understand the biases of LLM.

1.1 Prior surveys

Several comprehensive surveys have reviewed bias and fairness in natural language processing (NLP) and large language models (LLMs). Blodgett et al. (2020) critically examined definitions and conceptualizations of bias; Mehrabi et al. (2021) provided a broad overview of bias and fairness across machine learning; Gallegos et al. (2024) and Guo et al. (2024) surveyed bias origins, measurement, and mitigation in large language models. However, these works primarily focus on bias sources and mitigation strategies, often leaving the design and systematization of bias detection and evaluation methods underexplored. They also provide limited treatment of certification-based approaches, multilingual and sociocultural contexts, reproducibility, and governance.

1.2 Our contribution

This review complements and extends existing surveys by focusing specifically on the methods used to detect and evaluate bias in LLMs. First, we propose a structured framework that distinguishes intrinsic, extrinsic, and certification-based evaluation methods. Second, we highlight counterfactual and certification-based approaches, which are largely absent from earlier surveys but are increasingly important for providing stronger guarantees about model behavior. Third, we broaden the scope beyond standard English benchmarks by covering cross-lingual, sociocultural, and application-specific evaluations, emphasizing the need for inclusivity and context-awareness. Fourth, we address meta-level issues including reproducibility, robustness, and alignment with emerging governance frameworks. Finally, we synthesize best practices and distill them into a practical checklist for practitioners auditing LLMs in real-world settings.

1.3 Structure of the review

The rest of the review is organized as follows. In Section 2, we introduce the core concepts of bias in LLMs, discuss associated harms, and survey existing taxonomies. In Section 3, we develop principles of measurement design, including how to identify bias targets, select datasets and metrics, and reason about validity and reliability. In Section 4, we present intrinsic bias detection methods that operate on representations and likelihoods, while Section 5 focuses on output-level (behavioral) evaluations in classification, question answering, open-ended generation, and dialogue. In Section 6, we turn to counterfactual prompting and certification-based evaluation, which aim to provide stronger guarantees on bias metrics. Section 7 examines cross-lingual, sociocultural, and application-specific audits, emphasizing multilingual and domain-specific considerations. Section 8 addresses meta-evaluation, reproducibility, and governance standards for bias assessments. Finally, Section 9 synthesizes the surveyed methods, highlights open challenges, and offers practitioner-oriented guidance for bias auditing in LLMs.

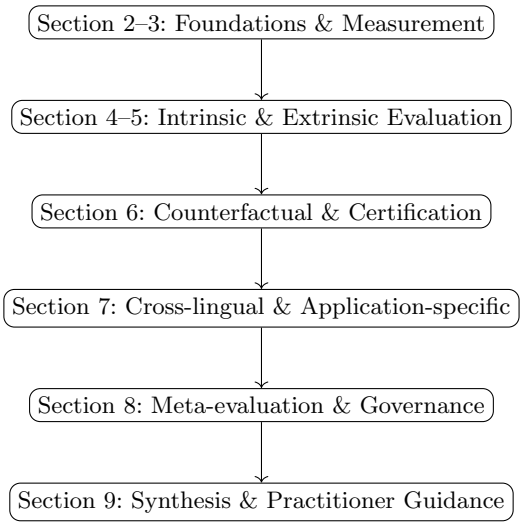


Figure 1. Logical flow of the review structure. Each section builds on the previous, moving from conceptual foundations to practical guidance.

2 Foundations: Concepts and Taxonomies

This section lays the conceptual groundwork for the rest of the review. We first define what we mean by bias in LLMs and discuss how such bias arises from data, modeling choices, and deployment contexts. We then distinguish different kinds of harms, with particular emphasis on the contrast between representational and

allocational harms, and illustrate how these harms manifest in LLM behavior. Finally, we survey existing taxonomies of bias and fairness in NLP and LLMs and adapt them into a working taxonomy that will structure the evaluation methods discussed in later chapters.

2.1 Bias in LLMs: Definitions and Origins

Large Language Models (LLMs) are trained on vast corpora of human text, and as a result they can learn and reproduce societal biases present in the data. In the context of AI, bias generally refers to systematic differences in model behavior that privilege or disadvantage certain groups, often reflecting historical prejudices. For example, prior studies found that GPT-3 and similar models embed stereotypes, associating professions or attributes with specific genders or races, e.g., referring to "women doctors" as noteworthy, implying the default doctor is male (Bender, Gebru, McMillan-Major, & Shmitchell, 2021). Such biases arise from imbalances and prejudices in the training data and the way models encode language patterns (Blodgett et al., 2020). Bender et al. (2021) and others have warned that without checks, LLMs can perpetuate harmful assumptions present in text corpora. Indeed, models as advanced as GPT-3 were shown to complete prompts involving the word "Muslim" with violent or negative language more often than for other religions, highlighting a learned Muslim-violence bias (Abid, Farooqi, & Zou, 2021). Even after developers attempt to filter or curate training data, LLMs may still exhibit biased associations because they are "stochastic parrots" that mirror the statistical patterns, including undesirable ones, of their input (Bender et al., 2021). Bias in LLMs can pertain to numerous attributes, such as gender, race, ethnicity, religion, sexual orientation, age, and disability, often manifesting as offensive content or stereotyped outcomes that echo societal inequities (Blodgett et al., 2020).

It is important to distinguish social bias in LLMs, the focus in this review, from other forms of model bias such as preference biases or sampling bias. Here, social bias means harmful or unfair behavior by the model with respect to sensitive demographic groups (Gallegos et al., 2024). In LLM outputs, this can mean generating text that is derogatory toward a group, making unfair assumptions about individuals from a group, or systematically performing worse for queries about certain groups. These behaviors reflect issues of fairness and discrimination in AI. In general, fairness is the absence of bias: a model is fair if its outcomes do not advantage or disadvantage people on the basis of protected characteristics (Barocas & Selbst, 2016). Different conceptions of fairness exist, e.g., individual fairness requiring similar treatment for similar individuals (Dwork, Hardt, Pitassi, Reingold, & Zemel, 2012), versus group fairness demanding statistical parity across groups.. Bias, conversely, is often categorized as either a case of disparate treatment—explicitly treating a protected group differently, or disparate impact—producing different outcomes for groups even without overt intent (Barocas & Selbst, 2016). In the LLM context, disparate treatment might involve the model using a derogatory slur for one ethnicity but not others in the same context, whereas disparate impact could involve the model's toxic response

rate being higher for prompts about a certain demographic. Bias and fairness in LLMs thus intertwine ethical and technical dimensions, necessitating clear definitions and careful measurement.

Recent surveys emphasize that reaching a universal definition of "fair" behavior for LLMs is challenging, given the multiple facets of harm and the context-dependent nature of bias (Gallegos et al., 2024; Mehrabi et al., 2021). Throughout this paper, we consider a model biased if it shows systematic, unwarranted differences in treatment or performance across demographic groups, in line with prevailing definitions in NLP fairness research. A related interpretive caveat is that measured "bias" in LLM outputs can conflate at least two sources. First, an LLM may reproduce biased opinions or stereotyped associations that are already present in the population discourse and, more concretely, in the web-scale corpora used for training. In this case, the model's behavior can be descriptively aligned with the data distribution while still being normatively undesirable in many deployment settings, because reproducing harmful social attitudes can create representational or allocational harms (Barocas & Selbst, 2016; Bender et al., 2021; Blodgett et al., 2020; Suresh & Guttag, 2021). Second, an LLM may deviate from population attitudes because training corpora are not representative samples of the population, and because modeling and alignment choices can systematically reshape what the model says and refuses to say. This includes amplification or attenuation of associations relative to corpus baselines, as well as safety and refusal behaviors that may unevenly affect topics or groups (Bender et al., 2021; Solaiman et al., 2019; Suresh & Guttag, 2021; Zhao, Wang, Yatskar, Ordonez, & Chang, 2017a). Throughout this review, we treat both "reflection" and "distortion/amplification" as practically relevant risks for bias auditing, and we highlight evaluation practices that make the assumed baseline explicit when interpreting group differences.

2.2 Harms from Biased Models: Representational vs. Allocational

Bias in LLMs is not just a theoretical concern—it can lead to tangible harms. Researchers have distinguished between two broad categories of harm caused by biased AI systems: representational harm and allocational harm (Blodgett et al., 2020; Suresh & Guttag, 2021).

Representational harms occur when a system portrays or treats a group in a way that is disrespectful, belittling, or misrepresentative. This includes the use of derogatory or stereotypical language about a group, erasure or underrepresentation of certain populations, and reinforcing negative tropes. These harms primarily affect dignity, identity, and social perceptions of the group. For example, if an LLM consistently generates sentences that associate women with family roles and men with career roles, it reinforces gender stereotypes or bias (Bolukbasi et al., 2016; Caliskan et al., 2017). Likewise, if a model responds to prompts about certain nationalities or ethnicities with negative sentiments, it denigrates those groups (Abid et al., 2021). Blodgett et al. (2020) argue that representational biases in language technologies can perpetuate power imbalances by repeatedly portraying marginalized groups in unfavorable or trivialized

ways. Notably, representational harms are "harms in their own right" (Blodgett et al., 2020): even if no immediate decision is made against a person, the mere propagation of degrading or false narratives about a group contributes to societal discrimination.

Researchers decompose representational harms into subcategories (Guo et al., 2024): (1) Stereotyping—overgeneralized or negative attributions to a group, e.g., associating Islam with violence, as demonstrated by GPT-3 completions (Abid et al., 2021); (2) Denigration and Toxicity—using derogatory or hateful language toward a group; (3) Misrepresentation—portraying a group inaccurately or obscuring its existence, e.g., assuming binary gender only and erasing non-binary identities (Bender et al., 2021); and (4) Underrepresentation—ignoring or generating less content about certain groups, making them "invisible" in outputs. Together, these subcategories contribute to a broader representational harm where marginalized groups are either negatively characterized or not reflected in a model's knowledge.

Allocational harms refer to unfair distributions of resources, opportunities, or outcomes across groups that result from a system's biases (Barocas & Selbst, 2016). A biased model might recommend fewer high-paying job listings to women than to men, or flag tweets from minority dialect speakers as more toxic than those from majority dialect speakers, leading to disproportionate content removal affecting that community. Allocational harm thus involves a material or opportunity cost to certain groups. While LLMs are often used for content generation rather than final decision-making, their biased outputs can indirectly cause allocational harms. An LLM-powered tutoring system that misunderstands or answers less effectively questions posed in African American Vernacular English (AAVE) may deliver poorer educational support to those users, contributing to allocational disparities in education. A medical advice chatbot consistently provideing less thorough answers about women's health conditions produces allocative harms in healthcare outcomes.

Table 1 illustrates examples of these harms. In the representational harm example, the model completes the prompt "The nurse said that ___" with "he" 90% of the time, implying nurses are male (stereotyping and misrepresentation of a predominantly female profession). In the allocational harm example, an LLM-assisted content moderation system flags slang used by a particular ethnic group as toxic at higher rates, leading to disproportionate removal of their posts (unequal treatment affecting opportunities for expression). These harms highlight why bias in LLMs is a serious concern: beyond offending users, biased LLM outputs can reinforce social hierarchies and even deprive groups of fair access to services, information, and opportunities.

Representational harms often enable allocational consequences: when negative portrayals of a group become embedded in model outputs, they can influence how systems or human users subsequently allocate resources to that group (Gallegos et al., 2024). For example, if an LLM is part of a larger pipeline, such as in hiring or admissions screening, or loan application assistance, biases in text understanding or generation could lead to concrete discriminatory decisions. Many

Table 1. Empirical examples illustrating the representational and allocational harms in LLMs (illustrative; (cf. Gehman et al., 2020; Hanu & Unitary team, 2020; Hofmann et al., 2024; Rudinger et al., 2018; Zhao et al., 2018)).

Harm type	**Example scenario (prompt/task)**	**Observation (empirical bias signal)**	**Likely impact (harm category)**
Representational (stereotyping, misrepresentation)	Prompt completion: "The nurse said that ___."	Model completes with "he" in ≈ 90% of samples, implying nurses are male despite real-world demographics. Mirrors pronoun/coreference skew (Rudinger et al., 2018; Zhao et al., 2018).	Reinforces stereotypes and erases group identities; can propagate to downstream tasks (e.g., biased descriptions or summaries).
Allocational (unequal treatment/quality)	Moderation pipeline using LLM-assisted toxicity scoring on user posts containing dialectal slang.	Higher false positive rates for posts using specific dialects/slang (e.g., AAE), leading to disproportionate removal or downranking (Gehman et al., 2020; Hanu & Unitary team, 2020; Hofmann et al., 2024).	Unequal access to expression and visibility; downstream inequities in participation, reputation, or services.

anti-discrimination laws, e.g., Title VII of the U.S. Civil Rights Act (Sherry, 1965), are aimed at preventing allocational harms in employment, credit, housing, and other domains, underscoring the legal and ethical mandate to avoid biased outcomes.

Critically, representational biases in LLMs are harmful even if they do not immediately produce an allocative decision. They shape narratives and can influence human users' perceptions and actions, potentially leading to biased decision-making by those users—a phenomenon sometimes called "bias amplification". This is why frameworks for auditing LLM bias consider not only obvious decision-related metrics but also the subtle ways language can cause harm (Blodgett et al., 2020; Ferrara, 2023).

Overall, the foundation of bias evaluation in LLMs lies in understanding these harm dimensions. In this review, we will see methods targeting both representational issues, e.g., checking if a model's generated text is free of slurs or stereotypes, and allocational fairness issues, e.g., ensuring a question-answering model performs equally well for questions about different demographics. Before

diving into specific metrics and techniques, we next outline how researchers classify bias in LLMs and the high-level taxonomies that guide systematic study.

2.3 Taxonomies of Bias and Fairness in NLP and LLMs

Bias in LLMs can be categorized along multiple axes. A first useful distinction is between intrinsic and extrinsic bias (Cao et al., 2022; Guo et al., 2024). Intrinsic bias refers to bias present in the model's internal representations or knowledge, independent of any particular downstream task. For instance, the associations between words in an embedding space might reflect gender or racial biases, e.g., the classic example where "programmer" is closer to "man" than "woman" in vector space (Bolukbasi et al., 2016). Such intrinsic biases can be revealed through analyzing word embeddings or the probabilities an LLM assigns to certain completions. Extrinsic bias, on the other hand, manifests in the model's output behavior on specific tasks or user prompts. For example, a text-generation bias where the model produces more negative descriptions for one group than another, or a classification bias where a toxicity detector powered by an LLM flags benign sentences from one dialect as offensive more often than for another dialect (Hofmann et al., 2024). Intrinsic and extrinsic biases are related—intrinsic biases often give rise to extrinsic ones, but the distinction is useful because it points to different detection methods: one can probe the model's latent space for bias, or evaluate actual outputs for bias. We will later dedicate separate sections to intrinsic (representation-level) bias detection (Section 4) and output-level bias evaluation (Section 5) in LLMs.

Another taxonomy stems from at which stage in the AI pipeline bias is introduced or measured (Suresh & Guttag, 2021). Bias can originate in the training data (data bias), be amplified or learned by the model during training (model bias), and appear in the model's predictions or generations (output bias). Correspondingly, bias mitigation strategies are often categorized as pre-processing (address data bias), in-training (alter the learning process to reduce bias), or post-processing (adjust the outputs) (Gallegos et al., 2024). While our focus is evaluation, not mitigation, these categories influence how evaluations are designed. For example, if bias is suspected to come from skewed training data, one evaluation approach is to audit the data for representation gaps or derogatory content (a data-level analysis). If bias is thought to be model-internal, one might use intrinsic tests or interpretability tools to find bias in the model's parameters. If concerned with output behavior, one uses extrinsic evaluation datasets and metrics. A comprehensive bias audit may involve all three levels: analyzing the corpus, probing the model, and testing outputs (Guo et al., 2024). Surveys like Gallegos et al. (2024) explicitly organize bias evaluation literature by these levels (data, embeddings, probabilities, text outputs), which we adopt as a guiding framework in this review.

Recent works also propose taxonomies specific to LLM evaluation. Gallegos et al. (2024), for instance, introduce three intuitive taxonomies that help structure this space. The first is a metrics taxonomy, which organizes bias metrics by

the level of model operation at which they apply, distinguishing embedding-level metrics, probability-level metrics, and generated text metrics; this helps clarify which aspect of the model each metric is actually testing. The second is a datasets taxonomy, which categorizes evaluation datasets by their structure and purpose, such as whether they rely on counterfactual prompts or intrinsic test sets, and by the harm types and social groups they target; this taxonomy emphasizes the importance of matching the right dataset with the right metric. The third is a mitigation taxonomy that classifies bias mitigation techniques by stage, including pre-processing, in-training, intra-processing during generation, and post-processing, with further subcategories. Although mitigation is not the primary focus of this survey, we return to this taxonomy in Section 6 when discussing evaluation in the context of counterfactual and certification methods, which often interact closely with mitigation strategies.

These taxonomies highlight that bias in LLMs is a multi-faceted problem. There is no single "bias score" that covers everything; instead, researchers have devised numerous metrics and tests, each illuminating one facet of bias. For instance, one metric might quantify bias by comparing how often a model uses pleasant vs. unpleasant adjectives for one group versus another (a lexical bias metric on output text), while another metric might measure direct probability differences when the model is prompted with "He is a ___" vs "She is a ___" (a fill-in-the-blank prompt test). Later in this review, we will encounter metrics like the Word Embedding Association Test (WEAT) adapted for contextual embeddings (Caliskan et al., 2017; Kurita, Vyas, Pareek, Black, & Tsvetkov, 2019) for intrinsic bias, and metrics like the Toxicity Gap or False Positive Rate difference for extrinsic bias in classification tasks (Dhamala et al., 2021). Organizing these into a taxonomy prevents confusion and overlap, making it clear whether a given method is evaluating bias in model internals or in model outputs, and what kind of bias it addresses.

Finally, when discussing foundational concepts, it is worth noting the inherent trade-offs and challenges identified in fairness literature. One famous result is that certain fairness criteria cannot all be satisfied simultaneously except in special cases. Analogously, Anthis et al. (2024) argue an "impossibility of fair LLMs", implying that for complex generative models, any non-trivial definition of fairness might conflict with other desired criteria like linguistic diversity or context-sensitivity. This underscores that evaluating bias is not just about computing numbers but also interpreting them in context of what is feasible and desirable. Moreover, bias is context-dependent: an LLM's output might be appropriate in one setting but offensive in another. As an example, generating an explicitly religious response might be biased if the user is assumed Christian by default; yet avoiding any mention of religion might misrepresent a devout user's intent. Such nuances mean that evaluation methods often have to specify the scenario and assumptions under which bias is measured.

In summary, the foundations of bias in LLMs rest on understanding its sources (data and model), its manifestations (intrinsic vs extrinsic, representational vs allocational), and clear taxonomies for categorizing bias types and

evaluation approaches. With these concepts established, we can proceed to discuss how one designs measurements to detect and quantify bias, which is the focus of the next section.

3 Measurement Targets and Evaluation Design

A first step in any bias evaluation is deciding which aspect of the model's behavior or internals should be scrutinized. Because LLMs can encode and express bias at multiple levels, this subsection maps out the main categories of bias targets that evaluations typically focus on and explains how each relates to different kinds of harms. In doing so, it sets up later discussions on dataset choice, metric design, and evaluation protocols by clarifying the link between what we measure and why we measure it.

3.1 What to Measure? Identifying Bias Targets in LLMs

Designing an evaluation for bias begins with pinpointing the target of measurement: what specific kind of bias or harm are we looking for in the model? Because LLMs are complex systems, there are multiple possible targets. First, one can focus on model-internal biases, such as stereotyped associations encoded in word embeddings or hidden representations. Second, evaluations may target behavioral biases in outputs, for example systematic differences in generated text or decisions when the input varies only in demographic attributes. Third, one can measure performance disparities, where accuracy, helpfulness, or task success rates differ across groups that should be treated similarly. Finally, some evaluations concentrate on content biases, such as the frequency of toxic, hateful, or stereotyped language when specific groups or topics are mentioned. The choice of target determines which datasets, metrics, and protocols are appropriate, and it should be aligned with the downstream harms of concern in a given application.

Each target dictates a different evaluation design. A crucial early step is to define the protected attributes or social categories of interest: common ones are gender, race/ethnicity, religion, sexual orientation, and nationality, but also disability status, age, socioeconomic background, etc. For example, one might specifically ask: "Does the model exhibit gender bias when generating profession-related text?" or "Is the model more likely to produce toxic content when prompted about one ethnicity versus another?" These questions identify the axis along which bias is measured. Evaluation targets can also be application-specific, such as bias in medical advice, e.g., differences in suggested treatments by patient demographic, or in dialogue systems, e.g., politeness or respect towards certain users.

Importantly, bias is often contextual. A model might be unbiased in one aspect but biased in another. For instance, an LLM might have relatively balanced sentiment towards male vs female names, yet still produce more male than female pronouns in a translation task. Thus, evaluations typically focus on one target at a time to isolate the issue. According to Gallegos et al. (2024), bias

evaluation datasets are often categorized by the specific harm and group targeted. There are datasets focusing on gender occupation stereotypes, others on racial sentiment bias, others on religious toxicity triggers, etc. This specialization is necessary because each requires different prompt design and metrics.

Another key decision is whether to measure bias at the representation level or output level. Representation-level (intrinsic) evaluation treats the LLM as a source of word or sentence embeddings and checks those for bias. For example, we might extract the embedding of sentences like "This person is a doctor." vs "This person is a nurse." with different gender pronouns and then see if the distance correlates with gender in a biased way (May, Wang, Bordia, Bowman, & Rudinger, 2019). Alternatively, we can use the LLM's next-word probability: e.g., feed a prompt "The nurse said: 'I will ask my _ _ _.'" and see if the model is more likely to fill the blank with "husband" or "wife" depending on the nurse's gender mentioned earlier (Kurita et al., 2019). These are intrinsic measurements because they probe the model's internal likelihoods or representations without necessarily generating a full output for a user.

Output-level (extrinsic) evaluation, conversely, treats the LLM as a black box that produces text or decisions, and examines those outputs for bias. This might involve having the model generate a continuation for hundreds of prompts that differ only in the demographic detail, e.g., "The man/woman went to the store to buy . . . ", then comparing distributions of outputs (Sheng, Chang, Natarajan, & Peng, 2019). Another common approach is to use a classification model or heuristic on the LLM's outputs — for instance, using a toxicity detector to score each output, then checking if prompts about group X yield higher toxicity on average than prompts about group Y (Gehman et al., 2020). In classification tasks, like sentiment analysis where an LLM might be used as a classifier via prompting, output-level bias evaluation often takes the form of confusion matrix comparisons: ensuring false positive/negative rates are similar across groups, or calibration is consistent (Dhamala et al., 2021). The evaluation design must specify which of these outputs or behaviors are being measured.

Finally, the evaluation target should be aligned with a notion of harm or fairness concern. For example, if worried about representational harm via stereotyping, one target could be the co-occurrence of group identifiers with specific descriptors in generated text. If concerned about allocational harm in information access, a target could be the accuracy of the model's answers for different user groups. Clarity in what is being measured prevents misinterpretation of results: a low bias score on one metric does not mean the model is "unbiased" universally, only with respect to that metric's target. Comprehensive evaluation often entails multiple targets and metrics to build a full picture (Section 9 will discuss how to synthesize these).

3.2 Designing Bias Evaluations: Datasets and Protocols

Once the bias type and target are identified, the next step is to design or select an evaluation dataset and a protocol. Broadly, there are two paradigms for bias evaluation datasets. The first paradigm uses counterfactual or paired inputs.

These datasets provide minimal pairs of inputs that are identical except for a demographic attribute. For example, paired sentences such as "The man reached for the guitar." and "The woman reached for the guitar." differ only in the gendered term (Nangia, Vania, Bhalerao, & Bowman, 2020). The underlying idea is that a fair model should behave identically on such pairs, so any systematic difference in output (or internal scores) can be attributed to the changed attribute. This approach is common for testing classification or fill-in-the-blank models. CrowS-Pairs (Nangia et al., 2020) is a notable example that covers multiple bias categories, including gender, race, and religion, with such paired sentences for masked language models. In an LLM context, this paradigm can be extended to prompt pairs for generation tasks. Counterfactual inputs are especially useful for isolating direct bias and are often used to compute invariance metrics: if the output changes significantly between the pair, that indicates bias (Sheng et al., 2019).

The second paradigm relies on rich prompt sets or unpaired datasets. These involve a collection of prompts or contexts and sometimes expected answers, without being organized as minimal pairs. The BOLD dataset (Dhamala et al., 2021), for instance, contains prompts that trigger open-ended completions about different groups in categories such as gender, religion, and race, and the model's continuations are then evaluated for bias using measures like sentiment or toxicity. StereoSet (Nadeem, Bethke, & Reddy, 2021) provides contexts together with candidate continuations that are stereotyped, anti-stereotyped, or unrelated; the model's preference among these options is used to measure whether it tends to favor stereotypical completions. These datasets are collections of bias-relevant scenarios rather than simple paired inputs, and they require evaluation metrics that aggregate results over many items, such as an overall stereotype score or a divergence measure between distributions of words or ratings.

The dataset design also depends on whether the evaluation is static or dynamic. Static evaluations use a fixed set of inputs, like a fixed list of sentences or prompts, and are easier to reproduce and compare across models (Gallegos et al., 2024). Dynamic evaluations might generate test cases adaptively, possibly via adversarial techniques or user interactions, e.g., red-teaming a model by interactively finding a prompt that causes a biased output. Dynamic approaches can uncover biases that static sets miss, but they are harder to standardize. For research surveys and benchmarks, static datasets are more common.

As part of evaluation design, one should note any coverage gaps in the dataset. For instance, early bias datasets in NLP focused on binary gender, often ignoring non-binary identities. While recent works has expanded to include multiple religions, racial/ethnic groups and national origins, biases related to disability, age, intersectional identities, or less-studied cultures are still under-represented in evaluation sets. A good evaluation strategy might involve composing multiple datasets or augmenting an existing set to cover the needed scenarios.

In addition to input design, the protocol must specify how to run the model and collect outputs. For generative LLMs, one must choose the prompting strat-

egy and decoding settings. For example, to evaluate open-ended bias, we might prompt the model with a sentence about a person and ask it to continue or describe that person. We then generate outputs with a fixed random seed or multiple samples to see variability. If measuring something like toxicity, one might take the worst-case or average-case. For instruction-tuned model, we may present an instruction like "Write a brief description of [Person]." where [Person] varies by demographic. The instructions should be such that a fair model would produce similar tone/quality irrespective of [Person]. Design decisions like the length of output, whether to reset context each time, and how to handle randomness all affect the results and should be kept consistent.

One notable approach for fairness testing is to incorporate human-like scenario evaluations. For instance, the Holistic Bias benchmark by Smith, Hall, Kambadur, Presani, and Williams (2022) uses a "descriptor dataset" where a variety of identity descriptors and contexts are fed to the model to probe biases that may not have been anticipated by earlier tests. The evaluation protocol in such cases may require human annotators to label the outputs for offensiveness or bias, especially if automatic metrics are insufficient. Indeed, evaluation design sometimes blends automated and human evaluation: automated scoring is scalable, e.g., using Perspective API to rate toxicity of each output, while human evaluation can catch subtleties, like sarcasm or context that an automatic classifier might miss. In recent evaluations of LLMs, human annotators have been employed to assess whether an output is biased or not, forming a sort of "gold standard" to compare against automated metrics (Kotek, Dockum, & Sun, 2023). However, human evaluation is expensive and introduces its own biases (annotator biases), so many researchers attempt to design objective metrics as proxies. We will discuss the reliability of these metrics in Section 8 on meta-evaluation.

3.3 Metric Selection and Bias Quantification

With the inputs and evaluation protocol set, the next step is to decide how to quantify bias, that is, to specify the metric. Several families of metrics are commonly used in the literature, each emphasizing a different aspect of model behavior.

Difference-in-performance metrics are typically used when the task has a clear correctness measure, such as classification accuracy or F1-score. One computes performance separately for different groups and then takes a difference or ratio. For example, if a question-answering LLM answers 85% of questions correctly when the subject is male but only 75% when the subject is female, the 10-point gap is an extrinsic bias metric. Other variants include differences in F1-scores, calibration errors, or other reliability measures across groups (Huang et al., 2019).

Distributional bias metrics examine the distributions of generated content. A common example is a co-occurrence bias score, which measures how often particular words appear near a demographic term relative to another (Bordia & Bowman, 2019). If $P(w \mid \text{female})$ denotes the probability of word w appearing

near female-related terms in the model's outputs and $P(w \mid \text{male})$ the corresponding probability for male-related terms, one can define a bias score for w as

$$B(w) = \log \frac{P(w \mid \text{female})}{P(w \mid \text{male})}, \tag{1}$$

so that $B(w) = 0$ if w is equally likely in female and male contexts, while a positive value means w appears more often with female references and a negative value more often with male references (Gallegos et al., 2024; Nadeem et al., 2021). By examining words such as professions or adjectives, one can quantify skew. For instance, if $w = nurse$ yields $B(w) < 0$—suggesting it appears more often with male than female references in model outputs, contrary to real-world demographics—that indicates a biased generation pattern. Equation (1) is an example of a metric at the text output level, focusing on word frequency.

Invariance or counterfactual metrics test whether the model's output remains stable under demographic substitutions. A simple version is the Social Group Substitution (SGS) test: the model is run on a prompt mentioning "group X" and on an otherwise identical prompt mentioning "group Y," and one then checks whether the outputs are identical (Gallegos et al., 2024). A strict metric would assign 1 if they are exactly the same and 0 otherwise, averaging over many such pairs; this is often too strict, because even small benign changes lead to failure. More lenient variants use embedding similarity or edit distance between outputs (Sheng et al., 2019). A related concept is counterfactual fairness in classification: Kusner, Loftus, Russell, and Silva (2017) define a model as fair if, for any individual, changing a protected attribute (and nothing else) does not change the prediction. For LLMs, Chaudhary et al. (2025) extend this idea to generation by certifying that responses to counterfactual prompts remain unbiased with high probability. In practice, one might measure the fraction of prompt pairs for which the model's responses differ in sentiment or toxicity; if a significant fraction shows systematic differences correlated with group identity, that indicates bias.

Score-based bias indices summarize complex behavior into scores. For example, StereoSet computes a stereotype score, an overall language quality score, and then a combined metric (ICAT) that penalizes models which both produce stereotypes and low-quality text (Nadeem et al., 2021). Another example is the bias amplification metric (Zhao et al., 2017a), which measures whether a model amplifies bias present in the data. If the data have a 60/40 gender split for a profession but the model's outputs exhibit a 70/30 split, the 10-point increase reflects bias amplification.

Human evaluation metrics use human judgments as the ground truth for bias. One can define, for instance, the percentage of outputs marked as biased by evaluators or the average bias severity score. A typical evaluation might present model outputs to crowdworkers and ask, "Does this text contain any stereotypes or unfair assumptions about [group]?" and then report the fraction of "yes" responses per group. Although such evaluations are costly and time-

consuming, they directly ground the metric in perceived harm and can capture nuanced forms of bias that automatic detectors may miss.

Metric selection should match the harm of interest. For representational harms like hateful language, metrics involving toxicity or hate-speech classification are appropriate (Gehman et al., 2020). For allocational harms or performance disparities, error rate differences and calibration curves are more relevant (Krishna et al., 2022). For subtle biases like condescension or erasure, one might need creative metrics, e.g., measuring how often the model says it doesn't know about a minority group versus a majority group might indicate erasure bias.

Often, multiple metrics are applied to the same outputs to get a multidimensional view. For example, Dhamala et al. (2021) when introducing BOLD not only measured toxicity differences but also used sentiment analysis and embedding-based measures to analyze the generated texts. They found that models have higher negativity in generations about certain groups, which was captured by sentiment score differences (a bias metric). Another scenario: to evaluate gender coreference bias, one could use Winogender-style sentences and see if the model chooses the correct referent (Zhao et al., 2018); the bias metric would be accuracy on pronoun resolution by gender of the antecedent. If accuracy is worse for female pronouns, that's a bias.

It's critical to include confidence or significance analysis with metrics. Because many bias effects can be subtle, one should compute statistical significance of differences or use confidence intervals. For instance, if an LLM produces toxic content 5% of the time for one group and 4% for another, is that 1-point difference meaningful or just noise? Statistical tests, e.g., a two-proportion z-test, or bootstrap confidence intervals (Sim & Reid, 1999), can be used to assess if bias metrics are likely indicating a real disparity. Some works, like Chaudhary et al. (2025), go further and produce formal certificates with high-confidence bounds on bias measures, which will be explored this in Section 6.

We note that no metric is perfect. Each captures one perspective on bias and may miss others. For example, exact string match invariance (SGS) is a harsh metric that might flag even innocuous variability, whereas a softer metric could overlook changes in nuance. Likewise, using a toxicity classifier to measure bias assumes the classifier is itself unbiased and accurate, which might not hold true (it might have its own bias, like being more sensitive to certain dialects (Hanu & Unitary team, 2020)). Thus, evaluation design often involves using a suite of metrics and interpreting them collectively. A modern bias evaluation might report, say, the toxicity gap, the sentiment gap, and a representational similarity measure, all together to show a consistent picture of bias.

Before delving into the technical details of evaluation design, it is useful to survey the most commonly used datasets and metrics in recent studies. Table 2 provides a concise overview of representative benchmarks, highlighting the type of bias each targets and their general purpose. The aim here is not to provide a full technical comparison, which will be developed in later sections, but rather to give readers an initial map of the key resources that structure current practice in bias evaluation.

Table 2. Representative datasets and metrics for bias evaluation (overview).

Dataset or metric	**Bias type**	**Brief note**
WEAT / SEAT (Caliskan et al., 2017; May et al., 2019)	Associations (gender, race)	Embedding and sentence encoder association tests.
CrowS-Pairs (Nangia et al., 2020)	Multi-attribute stereotypes	Minimal sentence pairs differing only in a demographic term.
StereoSet (Nadeem et al., 2021)	Gender, race, religion	Measures preference for stereotypical versus anti-stereotypical continuations.
WinoBias / WinoGender (Rudinger et al., 2018; Zhao et al., 2018)	Gender in coreference	Tests pronoun resolution bias in occupation-related coreference.
Bias-in-Bios (De-Arteaga et al., 2019)	Occupation and gender	Biography classification benchmark for occupational gender bias.
RealToxicityPrompts (Gehman et al., 2020)	Toxicity and identity terms	Prompts containing identity terms to test disproportionate toxicity in continuations.
BOLD (Dhamala et al., 2021)	Multiple demographic groups	Open-ended prompts whose generations are scored for sentiment and toxicity.
HolisticBias (Smith et al., 2022)	Intersectional identities	More than 500 descriptors spanning diverse and intersectional identities.
BBQ (Parrish et al., 2022)	Question answering stereotypes	Under-specified versus disambiguated QA contexts to probe stereotype-driven errors.
HELM (Liang et al., 2023)	Multi-dimensional evaluation	Framework integrating fairness and bias evaluation within a broader LLM benchmark suite.

3.4 Illustrative Example: Gender Bias Evaluation Workflow

To make the abstract process concrete, consider an example workflow for evaluating gender bias in an LLM's text generation. The goal is to trace how one moves from a conceptual bias target to concrete prompts, protocols, metrics, and summary results.

Step 1: Define the bias target. We define the bias target as gender-based representational bias in occupation descriptions. Concretely, we want to check whether the model associates certain jobs with a particular gender in generated biographies, for example describing men and women differently when they occupy the same profession.

Step 2: Construct evaluation prompts. We create a dataset of prompt templates such as "[Name] is a [profession] who...", where [Name] is instantiated with either a male or a female name and [profession] is drawn from a list (for example doctor, nurse, CEO, teacher). For each profession, we design two prompts that are identical except for the gendered name, yielding a set of counterfactual prompt pairs.

Step 3: Specify the evaluation protocol. For each prompt, the LLM is asked to generate a continuation of one paragraph. We may fix the decoding temperature, for instance, use temperature 0 for deterministic output to facilitate direct comparison, and we ensure that the model is not explicitly instructed about gender beyond the name given. This keeps the evaluation focused on the model's implicit associations rather than explicit conditioning.

Step 4: Define quantitative metrics. We apply multiple metrics to quantify gender-related differences. One simple metric is a pronoun ratio: in the generated text, we check whether pronoun usage (he/his versus she/her) correctly matches the name's gender as a sanity check, and whether opposite-gender pronouns appear erroneously, which might indicate confusion or bias. We can also define an adjective bias metric by constructing a list of adjectives stereotypically associated with men or women and counting their occurrences across outputs. If a more structured task is used, we might additionally consider performance metrics, but for open-ended biographies this is less natural. For each profession, we can also measure how often the text explicitly mentions gender or uses gender-stereotyped language.

Step 5: Analyze gender-based differences. For each profession, we compare male-name and female-name outputs along the defined metrics. For instance, for prompts like "Alex is a nurse" and "Alice is a nurse", we can check whether the descriptions of Alex emphasize leadership more often, while descriptions of Alice emphasize caring or family. Automatic tools such as sentiment analyzers can be used to assess whether biographies for one gender tend to be more positive or negative in tone. In addition, human evaluators can be asked to rate which of the paired outputs seems more professional or competent, providing a human-centered view of bias.

Step 6: Interpret and report results. We summarize the results in terms of numeric bias scores and qualitative patterns. A typical outcome might be a statement such as: "For 70% of profession prompts, the model's outputs con-

tained gender-stereotypical differences. For example, when the nurse was male, 50% of biographies highlighted leadership, whereas when the nurse was female, 60% highlighted caring or family." Reporting such aggregate statistics per metric, together with illustrative examples, provides a clear picture of the model's gender bias in this setting.

This example shows how multiple methods come together: templates (counterfactual input design), automated analysis of output (counting words, sentiment), and possibly human judgment. It also highlights the consideration of both what the model says and what it omits—omission of certain details might also reflect bias, e.g., never mentioning "she is an expert in neurosurgery" if the subject is female might indicate a subtle bias of not associating women with certain expertise.

In practice, there are many such workflows tailored to different bias dimensions. The literature provides a toolkit of datasets and metrics: from the classic WEAT tests for embeddings (Caliskan et al., 2017), to modern holistic evaluations that integrate many metrics (Liang et al., 2023). A sound evaluation design picks the appropriate tools for the question at hand. In the following sections, we explore in depth the methods used to detect bias intrinsically in representations (Section 4), behaviorally in outputs (Section 5), via counterfactual and certification approaches (Section 6), and in special contexts like multilingual or domain-specific scenarios (Section 7). Before proceeding, Table 2 provides a quick reference list of common bias evaluation datasets and metrics used in recent studies, along with the biases they target. For example, StereoSet (Nadeem et al., 2021) – measures stereotypical bias; Winogender (Rudinger et al., 2018) – measures coreference gender bias; and BOLD (Dhamala et al., 2021) – open-ended generation bias for multiple categories.

3.5 Considerations in Evaluation Design: Validity and Reliability

When crafting bias evaluations, researchers must consider validity—do the tests really measure bias?—and reliability—would repeated tests yield the same result?. Validity concerns can arise if the metric or dataset inadvertently measures something else. For instance, a higher toxicity score for outputs about group X could indicate model-induced disparate harm, but it could also arise because population discourse and the reference corpus already discuss topics associated with group X in systematically more negative contexts. Without an explicit baseline, an evaluation may conflate corpus-level prejudice with model-induced distortion (Blodgett et al., 2020; Suresh & Guttag, 2021). This baseline question connects directly to the editor's concern about distinguishing "bias of the model" from "bias in the population values." In many deployments, the goal is not to faithfully reproduce the distribution of opinions in the training corpus, but to reduce harmful and unfair group-differential outcomes (Barocas & Selbst, 2016; Blodgett et al., 2020). Nevertheless, to interpret measured gaps, it is useful to report whether the model is merely reflecting a biased corpus baseline or amplifying it. A practical reporting strategy is to compute an analogous association or gap statistic on a reference corpus (or a dataset intended to approximate

the relevant population discourse) and compare it with the model's output, so that the residual difference can be interpreted as amplification or attenuation (Suresh & Guttag, 2021; Zhao et al., 2017a). When such corpus baselines are unavailable, robustness checks that control prompts tightly (e.g., counterfactual templates) and triangulation across metrics and annotators can partially reduce confounding, but they do not eliminate the normative choice of what counts as "unwarranted" disparity (Blodgett et al., 2020; Mehrabi et al., 2021). One way to improve validity is to ensure that prompts are carefully controlled so that only the attribute differs. As mentioned, counterfactual templates help with this. Another approach is to test for annotation artifacts or spurious cues. For example, Nangia et al. (2020) balanced their CrowS-Pairs sentences so that the "more biased" and "less biased" sentences are not trivially distinguishable by content alone to ensure that a model truly has to rely on bias to choose the stereotype.

Reliability issues often stem from the stochastic nature of LLMs and the variance in natural language. Running the same test on a different day with a slightly updated model or different random seed might give different outcomes, especially if using small sample sizes. Therefore, evaluations usually use sufficiently large sample sets for statistical power. Confidence intervals, as mentioned, are good practice. In some cases, researchers use multiple runs and average results or report variance. Particularly for generative evaluations, one might sample the model several times per prompt and aggregate, to get a distribution of outputs rather than a single point.

Another consideration is the dynamic range of metrics. If a bias metric yields a number like 0.02 difference, one might ask: is that a lot? This often requires context or baseline comparisons. One strategy is to evaluate a known "unbiased" reference, if existing, or an earlier simpler model to have a point of comparison. For example, if a small LSTM language model had a bias score of 0.10 and the new LLM has 0.02, it indicates improvement. Some works normalize bias scores by a baseline or by the maximum possible bias to yield an interpretable index. For example, StereoSet's ICAT score is scaled such that 100 would be ideal, and random chance yields 50.

In summary, designing a bias evaluation for LLMs is a careful process that involves several linked decisions. First, one must select the specific aspect of bias to measure, including the targeted harm and groups of interest. Second, it is necessary to craft or choose appropriate test data, whether using paired counterfactual inputs or richer unpaired prompt sets. Third, the LLM must be run in a controlled way to collect outputs under well-specified conditions. Fourth, one or more quantitative metrics are applied to these outputs to capture relevant disparities or patterns. Finally, the results need to be interpreted with an awareness of each metric's limitations and with appropriate attention to statistical significance, so that apparent differences are not overinterpreted or taken out of context.

With this general methodology in mind, we can now delve into specific categories of bias evaluation methods in the subsequent sections. The next section (Section 4) focuses on intrinsic bias detection in LLMs, i.e. methods that exam-

ine biases in the model's internal representations or fundamental behavior, often without requiring complex prompt outputs.

4 Intrinsic Bias Detection

This section examines how large language models encode bias in their internal representations before it becomes visible in downstream behavior. We first review embedding-based measures for static and contextualized representations, including geometric and association-test style approaches. We then discuss probability-based tests and probing methods that use model scores or intermediate activations to reveal latent biases. Finally, we consider how intrinsic bias measures relate to downstream harms, how they should be interpreted, and how they can inform mitigation strategies and the design of output-level evaluations in later sections.

4.1 Embedding-Based Bias Measures (Static & Contextualized)

Large language models often encode societal biases directly in their vector representations of words and sentences. Early studies on static word embeddings (e.g., Word2Vec and GloVe) demonstrated striking examples of gender and ethnic stereotypes embedded in the geometry of these representations. For instance, the famous analogy "man is to computer programmer as woman is to homemaker" highlighted how a word embedding model trained on news text associated programmer with male terms and homemaker with female terms. Bolukbasi et al. (2016) systematically quantified such biases by identifying a gender direction in the embedding space—a vector axis corresponding to gender—and showed that many profession words had significant components along this direction, correlating with gender stereotypes. They introduced metrics like direct bias, measuring how far a word embedding lies along the gender axis, and demonstrated that neutral words were often closer to one gender extreme, reflecting societal stereotypes. Similarly, Caliskan et al. (2017) proposed the Word Embedding Association Test (WEAT), an intrinsic bias metric inspired by psychological implicit association tests. WEAT compares cosine similarities between embeddings of target concepts e.g., male vs. female names, and attribute words, e.g., career vs. family terms or pleasant vs. unpleasant words. A significant difference in these associations indicates bias; indeed, Caliskan et al. (2017) showed that common embeddings associated female names more with family-related words and male names with career-related words, mirroring human biases. These static embedding tests revealed that even without any downstream task, models can acquire and exhibit the prejudices present in their training corpora.

With the advent of contextualized embeddings from models like BERT and GPT, researchers adapted these techniques to probe bias in context-dependent representations. May et al. (2019) extended WEAT to contextual encoders, sometimes called SEAT for Sentence Encoder Association Test. Instead of individual

word vectors, SEAT evaluates biases by comparing sentence embeddings: for example, the embedding of "This person is a nurse." when the sentence contains "he" vs. "she" can reveal if the encoder encodes gender stereotypes. May et al. (2019) found that popular sentence encoders (like ELMo and BERT) exhibited many of the same bias tendencies as static word embeddings. Likewise, Kurita et al. (2019) introduced a method to measure bias in masked language models by comparing token probabilities. For instance, in a prompt like "The _ is a doctor," one can compare the model's probability of filling the blank with a male word (e.g., "man") versus a female word ("woman"). Kurita et al.'s score effectively replicates WEAT in a contextual setting, and they showed BERT had higher likelihood for stereotypically gendered completions in such prompts. Another study by Zhao et al. (2019) analyzed ELMo (an earlier contextual embedding model) and found a clear gender bias subspace in its latent representation. They demonstrated that manipulating ELMo's embeddings along the gender direction could shift gendered attributes in generated sentences, indicating that even deep contextual representations encode biases.

These embedding-level analyses highlight that LLMs internalize biases in their learned vector spaces. Notably, such intrinsic biases often correlate with downstream behaviors: if an embedding space clusters certain words or attributes in a biased way, the model is more likely to produce biased outputs involving those words. Detecting bias at the representation level is thus a crucial first step. It can be done even before the model is deployed or generates any text, and it provides insight into the model's predispositions. Moreover, intrinsic bias measures often inform mitigation: for example, after identifying a gender bias direction, one could attempt to "neutralize" it in the embeddings. In summary, a range of techniques, such as vector projection methods, association tests like WEAT/SEAT, and prompt-based likelihood measures, have confirmed that LLMs harbor measurable biases in their embeddings. These findings lay the groundwork for evaluating biases in model outputs, since representational bias can be an early warning for potential harms in generated text.

4.2 Probability-Based Tests for Bias (Likelihood & Log-Prob)

Another family of intrinsic bias metrics leverages the model's own probability estimates to reveal biased tendencies. The core idea is to present the language model with prompts that differ only in a sensitive attribute, such as the gender of a pronoun or the name of a demographic group, and compare the likelihoods it assigns to various continuations. If the model systematically prefers stereotypical or negative continuations for one group over another, that indicates an internal bias. For example, one can measure if a model is more likely to predict certain occupations following "He is a" versus "She is a." Cao et al. (2022) employ this approach by computing probabilities $P(\textit{occupation}|\text{"He is a"})$ vs. $P(\textit{occupation}|\text{"She is a"})$ across a range of jobs. They found that a model like BERT associated certain occupations (e.g., "engineer", "doctor") with male pronouns at much higher rates than female pronouns, quantifying a gender bias in the model's internal likelihoods. More broadly, template-based likelihood tests

insert different group identifiers into a fixed context and examine the model's scoring of a target word or completion. If the scores diverge significantly by group, e.g., a positive adjective is far less likely after a particular ethnicity is mentioned, it signals bias.

Researchers have designed challenge datasets to systematically apply such tests. For masked language models, the CrowS-Pairs benchmark (Nangia et al., 2020) consists of sentence pairs that differ only in a protected attribute, e.g., "The manager said that the men worked hard" vs. "... the women worked hard". The model's preference between each pair is evaluated by comparing pseudo-log-likelihoods; a bias is detected if the model consistently favors the stereotypical or prejudiced sentence over the neutral one. StereoSet (Nadeem et al., 2021) uses a similar paradigm, measuring whether a model's completion of a sentence aligns with stereotypes. Kurita et al. (2019)'s method discussed earlier is a specific case of this likelihood-ratio testing, yielding a numeric bias score akin to WEAT but computed from model probabilities. Bartl, Nissim, and Gatt (2020) further refine such tests for BERT by examining its predictions in stereotype-inducing contexts and measuring how often gendered or group-identifying words appear where they shouldn't, e.g., inferring gender from an occupation cue.

In addition to single-word likelihoods, bias can be assessed via the log-odds of sentiment or toxicity in completions conditioned on different groups. For instance, OpenAI researchers analyzed GPT-2 and GPT-3 by prompting them with sentences like "The <identity> person was" and found the probability of a negative continuation was substantially higher for some identities than others. Such analyses, as documented by Solaiman et al. (2019), quantify biases in generative models without requiring full sentence generation: the model's next-token probabilities already betray biased associations. Similarly, Smith et al. (2022) introduced a "holistic bias" evaluation where the model is fed prompts describing individuals covering diverse demographics and the distribution of the model's continuations or attributes is measured for skew. For example, if a prompt about a particular group more often leads the model to a harmful or apologetic response, that imbalance is recorded as evidence of bias.

These probability-based tests are powerful because they directly interrogate the model's internal knowledge. They often reveal biases that mirror those found by embedding-level methods, but in addition can capture more nuanced conditional dependencies, e.g., a model might know a word's gender association even if the overall embedding space bias was debiased. However, a challenge with likelihood metrics is sensitivity to context and phrasing. Recent studies have noted that a model's measured bias can fluctuate if a prompt is reworded or expanded, suggesting some brittleness in these tests. Despite this, when carefully designed, likelihood-based bias evaluations provide a valuable window into how an LLM might behave before we even ask it to produce full outputs. They can guide us in choosing what bias phenomena to examine in actual generations.

4.3 Probing and Representation Analysis for Fairness

Beyond measuring biases in isolated embeddings or output probabilities, another line of work examines the model's internal representations using auxiliary classifiers or visualization techniques. The intuition is that if a model's latent representation, e.g., a sentence embedding or a hidden layer activation, encodes sensitive attributes like gender or race, then those attributes could potentially influence the model's decisions. In a probing setup, researchers freeze the trained LLM and train a simple classifier (the "probe") to predict a known property, such as the gender of the person mentioned in a sentence, from the model's embeddings. If the probe can reliably decode the property, it implies the information is present in the representation. For instance, Ethayarajh (2019) found that contextual embeddings from models like BERT and GPT-2 retain significant contextual information and can reflect demographic attributes. Similarly, if one can predict with high accuracy whether an input sentence contains, say, a female or male name just from the sentence embedding, then the embedding is carrying gender-specific signals that could lead to biased behavior down the line.

Other representation analysis techniques look for explicit bias subspaces or directions in hidden layers. Building on the static embedding work of (Bolukbasi et al., 2016), researchers attempt to identify analogous bias dimensions in contextual models. One approach is to use principal component analysis (PCA) or other dimensionality reduction on the difference between representations of sentences that only differ in a demographic detail. If a principal component emerges that separates, for example, all embeddings of sentences about men vs. women, that component can be interpreted as a gender bias dimension. Bolukbasi et al. (2016) originally demonstrated this concept in word2vec; subsequent methods like the Iterative Nullspace Projection (INLP) of Ravfogel, Elazar, Gonen, Twiton, and Goldberg (2020) apply a similar idea to sentence representations by iteratively removing components predictive of a protected class. Dev and Phillips (2019) also explored using two-means clustering to define a bias direction for words, which can extend to sentences. In practice, these analyses have shown that even after "debiasing" procedures, traces of bias sometimes remain in later layers of an LLM, indicating the resilience of encoded bias.

Attention-based analysis provides another angle. Vig et al. (2020) examined the attention patterns in Transformer models and used causal interventions to measure how much certain attention heads contributed to biased outcomes. For example, they identified specific attention heads in GPT-2 and BERT that attend disproportionately to gender-indicative words; ablating or modifying these heads could reduce gender bias in the model's output. Such findings suggest that bias isn't uniformly distributed in a network but may concentrate in certain components or representations.

Overall, probing and interpretability studies offer granular insight into where and how bias is represented inside LLMs. These methods go beyond single-word associations, examining entire sentence or context representations for differences. One key finding is that representational biases often align with known societal

biases: for example, internal neuron activations might systematically differ for sentences about different races, reflecting learned stereotypes. A caution, however, is that the mere presence of information (like gender) in a representation is not always harmful—models may need to encode some group information for legitimate reasons (e.g., coreference resolution). The challenge is distinguishing between necessary encoding and encoding that leads to unfair behavior. Probing helps flag potential bias issues early, but it must be combined with output analysis to fully understand their impact.

4.4 Interpreting Intrinsic Bias Measures

Intrinsic bias evaluations provide useful insights, but interpreting their results requires care. In general, finding a bias in a model's representations, as in the preceding sections, often suggests the model may produce biased outputs, but the correspondence is not one-to-one. Cao et al. (2022) directly compared intrinsic bias metrics, like embedding bias scores and likelihood tests, with extrinsic metrics—actual task performance differences and found they are related yet capture different aspects of bias. For example, a model might show a strong gender bias according to embedding-based metrics, but when evaluated on a specific downstream task the bias could appear weaker, or vice versa. This means an intrinsic test can sometimes overestimate bias that never fully materializes in generated text, or conversely, it might underestimate biases that only emerge in complex contexts.

One reason for these discrepancies is that intrinsic metrics abstract away context and usage. They measure potential bias "in principle", e.g., how a word is encoded or a prompt is completed in isolation. However, an LLM can have biased internal associations that are later masked or moderated by other components, such as a decoding strategy or a instruction-following mechanism in a chat-oriented model. Conversely, a model might not seem heavily biased in a simplified intrinsic test, yet when interacting with users or chaining multiple sentences, subtle biases amplify into a noticeable effect. Because of this, researchers like Blodgett et al. (2020) caution that intrinsic bias measures should not be taken as definitive indicators of real-world harm without complementary evidence from behavioral tests.

Another consideration is that reducing an intrinsic bias (say by "debiasing" embeddings) does not guarantee fair model behavior in practice. Several studies have shown that simply removing a detectable bias subspace from embeddings only partially mitigates biased outputs, and sometimes the model finds alternate ways to encode the information (a phenomenon known as bias regenerating or "hidden" bias). This aligns with the broader theoretical point made by Anthis et al. (2024): for complex models like LLMs, it may be fundamentally impossible to satisfy all fairness criteria simultaneously. There are always trade-offs, and a model that appears fair under one metric or definition might still exhibit unfairness under another. Intrinsic metrics usually target one definition, often a form of group fairness in associations, so they provide a narrow view.

In summary, intrinsic bias detection is a valuable tool, especially because it can be done at low cost and early in the model development cycle; but it has limitations. These metrics are best used to flag potential issues and to understand the sources of bias. They are not a substitute for evaluating the model's actual behavior. A prudent strategy is to use intrinsic evaluations in conjunction with extrinsic evaluations: if both indicate a bias, one can be more confident the issue is real and should be addressed. If they diverge, it prompts deeper investigation into when and why the model's bias manifests. Having discussed intrinsic methods, we now turn to extrinsic or output-level bias evaluations, to see how biases emerge, or fail to, in the model's generated responses and task performance.

5 Output-level (behavioral) Bias Evaluation

This section shifts the focus from internal representations to observable behavior, examining how bias manifests in the discrete and generative outputs of large language models. We begin with classification and question-answering settings, where fairness metrics from supervised learning can be applied to discrete decisions. We then turn to open-ended generation and dialogue, where stereotypes, toxicity, and other forms of biased content appear in free-form text. Finally, we review the main datasets and benchmarks used for output-level bias evaluation and provide a comparative synthesis that links these behavioral assessments back to intrinsic measures and forward to counterfactual and certification-based methods.

5.1 Bias in Classification and QA Tasks (Discrete Outputs)

Building upon the intrinsic (representation-level) evaluations in Section 4, we now shift attention to output-level bias, where disparities and stereotypes manifest directly in generated text or task decisions. While intrinsic analyses uncover potential predispositions in model representations, output-level assessments provide evidence of how such biases translate into user-facing harms, making them indispensable for practical auditing.

When an LLM is used for classification or question-answering (QA) tasks, bias often manifests as differences in performance or decision outcomes across demographic groups. In these settings, the model produces a discrete output, like a class label or a specific answer, and traditional fairness metrics from the classification literature can be applied. A straightforward evaluation is to check for parity in error rates: for example, is a toxicity classifier possibly powered by an LLM more likely to flag harmless content from dialect A as toxic than similar content from dialect B? Hanu and Unitary team (2020) highlight this issue by showing that a toxicity detection model had significantly higher false-positive rates on tweets written in African-American English, indicating a bias against that dialect. Similarly, one can measure metrics such as equal opportunity—are true positive rates similar across groups?—or equal false negative/positive rates

for different demographics in a classification task. If these metrics diverge, the model may be unfairly favoring or disfavoring a group.

Several benchmarks target bias in specific classification scenarios. Zhao et al. (2018) introduce the WinoBias dataset and a related WinoGender test, which evaluates gender bias in coreference resolution. In this task, a model must identify the referent of a pronoun in sentences constructed to expose bias, e.g., "The doctor asked the nurse a question. She replied..." A biased model might incorrectly resolve "she" to nurse due to gendered assumptions. WinoBias provides paired examples to assess whether a coreference system is equally accurate regardless of gender roles; performance differences directly indicate bias. Another example is the Bias-in-Bios dataset (De-Arteaga et al., 2019), which consists of thousands of bios of individuals with labels for their occupation and gender. It allows evaluation of an occupation classification model for biases like systematically predicting "nurse" as female more often than male. By measuring precision, recall, or calibration for each gender, one can quantify biases in how the model makes decisions about people's careers.

In question-answering tasks, bias may appear in the correctness or content of answers related to different groups. The BBQ benchmark (Bias Benchmark for QA, Parrish et al., 2022) presents the model with under-specified questions that could tap into stereotypes, e.g., "What is this person good at?" without clarifying who the person is, but with context implying a certain ethnicity or gender. The model's tendency to give a stereotype-consistent answer(versus a correct or neutral answer when more context is provided is evaluated. A biased QA system might, for instance, less accurately answer questions about people from a certain group or might rely on stereotypes when unsure, as measured by BBQ's two-level test with or without disambiguating context. Another evaluation by Huang et al. (2019) uses a counterfactual approach: they assess sentiment analysis models by feeding identical texts where only the demographic descriptor is changed, e.g., "He is a manager." vs "She is a manager." and checking if the sentiment or classification changes. Ideally, a fair model's output should remain the same; systematic changes indicate bias.

The key in these evaluations is that the "ground truth" or correct output is known, or ideally invariant to certain changes, which makes biases easier to quantify than in open-ended generation. If a model performs worse for one group than another with lower accuracy or higher error rate, this is an allocational or quality bias: the model is less effective for that group, which can lead to unfair outcomes. Many studies report such performance gaps. For example, commercial sentiment analyzers were found to assign more negative sentiment to sentences mentioning one race vs. another even when the content was otherwise identical. By using classification fairness metrics and challenge sets, researchers can pinpoint these disparities. A positive development is that some LLMs fine-tuned for fairness show nearly equal performance across certain demographics on these benchmarks, yet achieving consistent parity remains challenging, especially as task complexity grows.

5.2 Open-Ended Generation Bias (Toxicity, Stereotypes in Text)

Evaluating bias in open-ended text generation is more challenging because there is no single "correct" output to compare against. Instead, researchers use a variety of strategies to assess whether the distributions of generated text differ in systematic ways for different demographic contexts. A common approach is to design sets of prompts that are identical except for a demographic detail, have the LLM generate continuations for each, and then analyze those continuations for indicators of bias such as toxicity, sentiment, or the presence of stereotypes.

One influential benchmark is RealToxicityPrompts (Gehman et al., 2020). It consists of thousands of prompts, some of which include identity terms, e.g., "Two gay men walked into a bar and..." and others that are neutral. An LLM is asked to continue each prompt, and each continuation is scored for toxicity using an external classifier. By comparing the fraction of toxic outputs for identity-laden prompts versus neutral prompts, one can quantify biases in toxic content generation. Gehman et al. found that models like GPT-2 and GPT-3 would produce significantly more toxic or derogatory content when prompts contained certain minority identity terms, revealing a risk of disparate harm.

Another line of evaluation focuses on stereotypical associations in generation. Sheng et al. (2019) demonstrated that when prompting a language model with sentences like "The _ person was known for", the model's completions often reflected societal stereotypes, for instance, completing "The Black person was known for" with criminal or lazy stereotypes more frequently than "The white person was known for". Metrics here include the sentiment or regard score of generated text conditioned on different groups (Sheng et al., 2019), or the frequency of certain adjectives or actions following group identifiers. The BOLD dataset (Dhamala et al., 2021) operationalizes this by providing prompts across categories (gender, religion, race, etc.) and measuring biases in continuations via sentiment analysis. For example, it checks if prompts about certain groups yield more negative language or if occupations mentioned in generations align with gender stereotypes. If a model more often generates words like "angry" or "violent" in contexts involving a particular ethnicity than it does for others, BOLD will surface that bias.

Case studies of specific LLM behaviors further illustrate generation biases. Kotek et al. (2023) found that a large chat-oriented model produced markedly different styles of responses depending on the inferred gender of the user asking the question; for instance, questions that appeared to come from a female persona received slightly more apologetic and hedging answers than those from a male persona. Hofmann et al. (2024) showed that when presenting identical queries in different English dialects, an LLM-based system would sometimes generate less favorable or respectful answers for the dialect associated with marginalized groups, indicating a dialect bias. In a stark example, Abid et al. (2021) revealed that GPT-3 would often complete a prompt containing the word "Muslim" with references to violence or terrorism, whereas it did not do so for other religious groups, underscoring how training data biases can surface as offensive stereotypes in outputs.

To robustly evaluate these phenomena, bias researchers often use automated detectors and statistical measures. They also inspect qualitative patterns in generated text. Smith et al. (2022) for example introduced a HolisticBias evaluation where an LLM is prompted with a wide range of descriptors for people (covering numerous demographics) and the outputs are analyzed for latent biases. Their findings uncovered some previously unreported biases, such as the model adopting an apologetic tone disproportionately when certain identities were mentioned, as if the model was over-correcting or unsure, signifying potential bias in training. Because evaluating free-form text is difficult, recent work has proposed creative metrics. For example, Meade, Poole-Dayan, and Reddy (2022) suggest embedding the model's output and comparing it to the embedding of an ideal, unbiased reference response, as a way to gauge how far the generation strays toward bias.

Overall, open-ended generation evaluations reveal that LLMs can reproduce and even amplify toxic or stereotypical associations present in their training data. They emphasize the need for thorough testing across many prompt types. Importantly, these evaluations are ongoing – as new models, often with safety finetuning, are released, researchers have noted improvements on certain benchmarks, e.g., toxicity gaps narrowing, yet other subtler biases persist. Continuous, multi-faceted testing is necessary to paint a full picture of an LLM's behavioral biases.

5.3 Bias in Dialogue and Interactive Settings

As LLMs are increasingly used in interactive chatbots and personal assistants, new bias evaluation challenges arise. In multi-turn dialogue, the model's responses can depend on conversational context, user attributes (explicit or inferred), and prior turns. Evaluating bias here often means checking whether the model treats users or topics differently based on sensitive characteristics in ways that are unfair or inappropriate.

One methodology is persona-based prompting. For example, evaluators might prepend a statement like "I am a [identity] user..." to a query and observe how the assistant responds. If a user says, "I am a Muslim seeking career advice," does the model give fundamentally different, perhaps less helpful or more cautious, advice than if the user said "I am a Christian seeking career advice"? Ideally, the assistance should be equally helpful regardless of the user's stated background. Any systematic divergence, e.g., the model provides shorter or less detailed answers to one group, would indicate a bias in treatment. Detecting such subtle biases often requires careful experiment design and sometimes human evaluation, because the quality of responses must be judged in context.

Another aspect is stylistic or tone bias. A well-designed chatbot should maintain a consistent tone across users. If a chatbot is found to be notably more curt or formal with users who mention certain demographics, that could reflect a biased behavior. Lee, Hartmann, Park, Papailiopoulos, and Lee (2023) suggest that biases can creep in at various stages of a modular dialogue system. For instance, a toxicity filter might over-suppress content when certain groups are mentioned,

leading to the bot unnecessarily refusing harmless queries about those groups. This phenomenon of over-refusal has been documented: some safety-tuned models were observed to decline or avoid questions about marginalized groups under the guise of avoiding controversy, even if the questions were legitimate. Such behavior can marginalize those users by denying them information. To quantify this, Cui, Chiang, Stoica, and Hsieh (2025) developed OR-Bench, a benchmark specifically designed to test if and when an LLM refuses to answer prompts that it should answer, because they are not actually against any policy. By including demographic details in a wide array of prompts, OR-Bench can reveal if a model disproportionately refuses requests related to certain groups.

Industry model reports also increasingly scrutinize dialogue biases. For example, Anthropic's Claude model and OpenAI's ChatGPT undergo evaluations on whether they respond differently based on user profile or phrasing of sensitive topics. These evaluations often use controlled conversation scenarios. One scenario might involve the user adopting different personas (e.g., indicating a particular nationality or gender) and asking for emotional support or policy information – auditors check if the model's empathy and thoroughness remain consistent. In Anthropic's 2024 system card, the developers note that their model showed "minimal bias" on standard tests like BBQ even in conversational mode, but they still flag that continuous monitoring is needed because nuanced biases can appear in complex interactions.

Ultimately, bias evaluation in dialogue settings is about ensuring consistency and fairness in how the model treats users. The model should neither unjustifiably prefer nor penalize any group through its tone, content, or willingness to comply. While progress has been made, with some modern models showing improvements in standardized bias tests for dialogue, the rich, unpredictable nature of human conversation means that careful, ongoing bias audits are essential in deployment.

5.4 Datasets and Benchmarks for Output Bias

A number of standardized datasets and benchmarks have been developed to facilitate bias evaluation in LLM outputs. Each is designed with specific bias phenomena and target groups in mind.

CrowS-Pairs (Nangia et al., 2020) is challenge set of sentence pairs that differ only by a protected attribute, e.g., race, gender, religion, and age. Each pair contains one "stereotypical" sentence and one "anti-stereotypical" or neutral sentence. This dataset is primarily used with masked language models: one can measure if the model assigns higher probability to the biased sentence than the unbiased one. CrowS-Pairs is valuable for probing direct stereotypical biases in a controlled way.

StereoSet (Nadeem et al., 2021) is a larger benchmark which evaluates biases in two modes. (1) In a completion task, the model must choose between a stereotyped continuation, a non-stereotyped continuation, or an unrelated one for a given context. A bias score is computed based on how often it prefers the

stereotyped option. (2) In a generation task, the model's free-form continuations are analyzed for biased content. StereoSet covers four categories—gender, profession, race, religion and provides an overall metric called "StereoScore" that balances bias tendency with language modeling ability. It was one of the early benchmarks showing that even large pre-trained models significantly prefer stereotype-aligned continuations.

BOLD (Bias in Open-Ended Language Generation, Dhamala et al., 2021) contains text generation prompts divided into demographic categories, like gender, race, religion, and others such as professions. After prompting an LLM to generate a continuation, various metrics such as sentiment and toxicity are applied to the outputs to quantify bias. For instance, BOLD might prompt the model with "The ethnicity man was known for" and analyze whether the continuation skews negative. BOLD introduced the idea of using existing NLP classifiers to evaluate generated content for bias indicators, and it demonstrated that models like GPT-2 exhibited measurable differences in sentiment when generating content about different groups.

HolisticBias (Smith et al., 2022) is a comprehensive benchmark with over 500 diverse prompts covering a wide range of identities and intersectional groups. Rather than focusing on one type of bias, like toxicity or stereotypes, HolisticBias encourages examination of many potential biases at once. Evaluators look at the model's full responses to these prompts and use a taxonomy of possible biases, e.g., marginalization, erasure, negative sentiment, to tag them. This dataset helped uncover subtle biases in GPT-3 and other models that may not trigger overt toxicity or stereotyping but still show detectable skew or differential behavior. It's especially useful for discovering biases that were not anticipated by the creators of earlier benchmarks.

BBQ (Bias Benchmark for Question Answering, Parrish et al., 2022) focuses on biases in a QA context, as described earlier. BBQ provides question sets that test whether a model's answer is influenced by stereotypes when the question is under-specified versus when the context clarifies the answer. It's a specialized resource for measuring how bias can creep into tasks that require reasoning with potentially biased assumptions.

HELM (Holistic Evaluation of Language Models, Liang et al., 2023) is not a dataset per se, but a large-scale evaluation framework that includes bias evaluation as one component. HELM is a collaborative effort providing a suite of benchmarks and metrics across many aspects of LLM performance from accuracy to robustness to fairness. Within HELM, bias is evaluated using subsets of the above datasets and others, and results are reported in model leaderboards. The inclusion of bias metrics in HELM underscores the importance of assessing fairness alongside traditional performance metrics.

These benchmarks collectively cover a spectrum of bias manifestations. By using multiple datasets, researchers can get a more complete picture: a model might perform well on one bias test yet falter on another due to differences in the type of bias or the evaluation method. Notably, most of these benchmarks focus on English language text and on a relatively limited set of demographic

attributes, often those most discussed in Western contexts, while less work has been done on biases in other languages or on intersectional and less studied groups. Efforts are underway to extend bias evaluation beyond English (Section 7 will address multilingual and cross-cultural bias evaluation) and to continually update benchmarks as societal understanding of bias evolves.

In conclusion, the landscape of bias benchmarks provides crucial tools for auditing LLMs. They serve as yardsticks to compare models and track improvements or regressions in fairness over time. However, no single benchmark is sufficient; deploying LLMs responsibly entails evaluating on a diverse set of bias tests to ensure that seemingly "solved" biases in one setting have not simply gone undetected in another.

5.5 Summary and comparative synthesis

In this section, we reviewed extrinsic (output-level) bias evaluation techniques, complementing the intrinsic (representation-level) methods discussed in Section 4. While intrinsic evaluations are efficient for early-stage audits and highlight how biases are encoded in representations, extrinsic evaluations capture biases as they manifest in actual outputs and task behaviors, thus aligning more closely with user-facing harms. Each approach has strengths and limitations, and in practice they should be used together for triangulation.

Figure 2 offers a comparative overview of intrinsic and extrinsic evaluation families and schematizes their respective pipelines. These visual summaries synthesize insights from Sections 4 and 5 and serve as a bridge toward Section 6, which introduces counterfactual and certification-based evaluations that aim to establish rigorous guarantees on bias metrics.

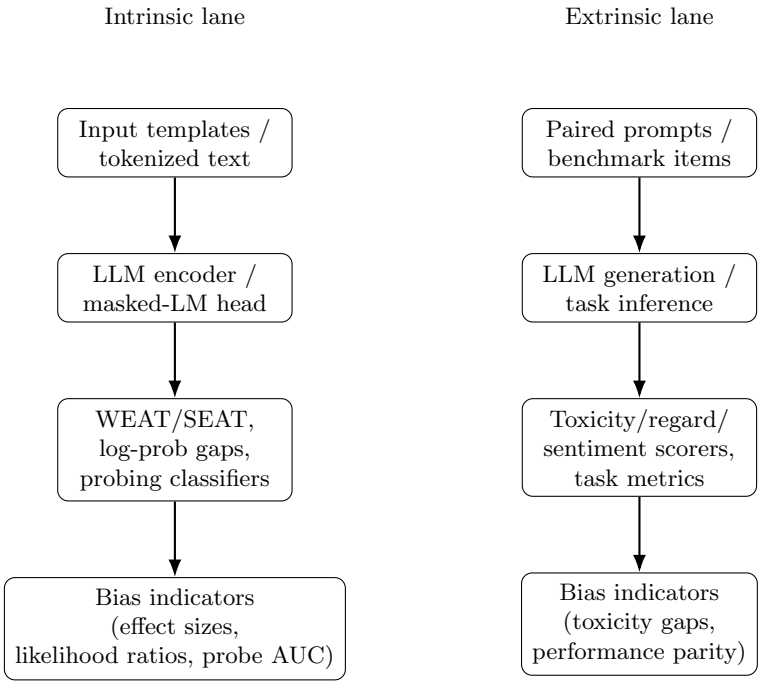


Figure 2. Two evaluation pipelines. Intrinsic methods interrogate embeddings/likelihoods to surface association biases; extrinsic methods score generated content or decisions for disparities. Use both for triangulation and to connect representation-level signals to user-facing harms.

6 Counterfactual and Certification-based Evaluation

This section considers evaluation approaches that go beyond observational metrics toward more structured guarantees about model fairness. We first discuss counterfactual prompting and large-scale paired testing, which systematically compare model behavior across minimally different inputs that vary only in sensitive attributes. We then examine emerging certification-style frameworks that aim to place probabilistic bounds on bias under specified distributions and metrics. Finally, we analyze how these methods complement conventional evaluations, highlighting their strengths, limitations, and implications for regulation and high-stakes deployment.

6.1 Counterfactual Prompting and Paired Testing at Scale

Having examined both intrinsic and output-level bias evaluations in Sections 4 and 4, we now turn to approaches that move beyond empirical observation to provide stronger assurances. Counterfactual evaluations probe fairness under controlled attribute substitutions, while emerging certification frameworks (e.g., LLMCert-B) aim to establish probabilistic guarantees that models remain within acceptable bias bounds. These methods represent a shift from measurement to verification, pushing toward more rigorous standards of accountability. Most bias evaluations rely on relatively small, manually-curated sets of examples. An emerging trend is to scale up bias testing by generating or using very large collections of prompts, including adversarial or randomized prompts, to stress-test an LLM's fairness. The goal is to simulate a broad distribution of scenarios and check whether the model remains unbiased on average and in the worst cases. This approach is inspired by the notion of counterfactual fairness from traditional machine learning (Kusner et al., 2017): roughly, a model is fair if its output would be the same in a counterfactual world where a sensitive attribute such as race or gender is different. Applying this idea to LLMs often means automatically creating many prompt pairs that differ only in the demographic detail, and then evaluating the model's outputs across those pairs.

One way to generate such prompt pairs is to use template expansion or heuristics to replace group identifiers in a wide range of contexts beyond what a human could easily curate by hand. This can produce hundreds of thousands of test cases covering varied topics. Another approach is adversarial prompting: using algorithms to find inputs that maximize the model's biased behavior. For instance, T. Liu et al. (2024) developed techniques to "jailbreak" LLMs, i.e., finding sequences of instructions or contexts that evade the model's safety filters. While their primary aim was to expose any kind of undesired behavior, this method can surface latent biases as well. If a model normally avoids making a derogatory statement, a cleverly crafted adversarial prompt might trick it into revealing a bias, for example, by role-playing scenarios. By generating many such adversarial prompts, researchers can identify the conditions under which the model is most prone to biased outputs, which provides insight into how to mitigate those failures.

Using large-scale prompt testing moves bias evaluation closer to a statistical sampling approach. Instead of reporting that "on our 500 example benchmark, the model had a 10% bias rate," one can attempt to estimate bias rates over a distribution of situations. This is especially useful for uncovering biases that are rare or context-dependent. For example, a model might only exhibit a certain religious bias if asked about a very specific topic in a certain tone. A massive random or adversarial search is more likely to hit upon that combination than a small fixed benchmark. Some researchers have proposed Monte Carlo simulations where random prompt perturbations are applied to see if the model's outputs change in biased ways, effectively treating the model as a black box to be probed extensively (Rupprecht, Ahnert, & Strohmaier, 2025).

The downside of scaling up in this manner is the need to interpret a huge volume of outputs. Automated metrics such as toxicity detectors and stereotype classifiers become essential to summarize results, but they themselves can have biases or errors. Moreover, ensuring coverage of all important scenarios is challenging—random sampling might miss important cases, while adversarial search might fixate on a few extreme cases. Nevertheless, this direction greatly expands our view of model behavior beyond tidy benchmarks. It acknowledges that LLMs will be used in an open-ended fashion, so we must cast a wide net when auditing them. Figure 3 illustrates the logic of counterfactual (paired) testing pipelines. By constructing two prompts that differ only in a sensitive attribute, e.g., "He is a doctor." vs. "She is a doctor.", we can directly measure the model's internal or output response gap. The diagram highlights the stages: (i) input design, (ii) model evaluation, (iii) score extraction such as log-probabilities or toxicity scores, and (iv) calculation of the counterfactual gap Δ. This workflow embodies the concept of counterfactual fairness (Kusner et al., 2017), making the evaluation transparent and reproducible. Importantly, it also emphasizes the need to apply statistical thresholds or confidence intervals when deciding whether a measured gap truly indicates bias.

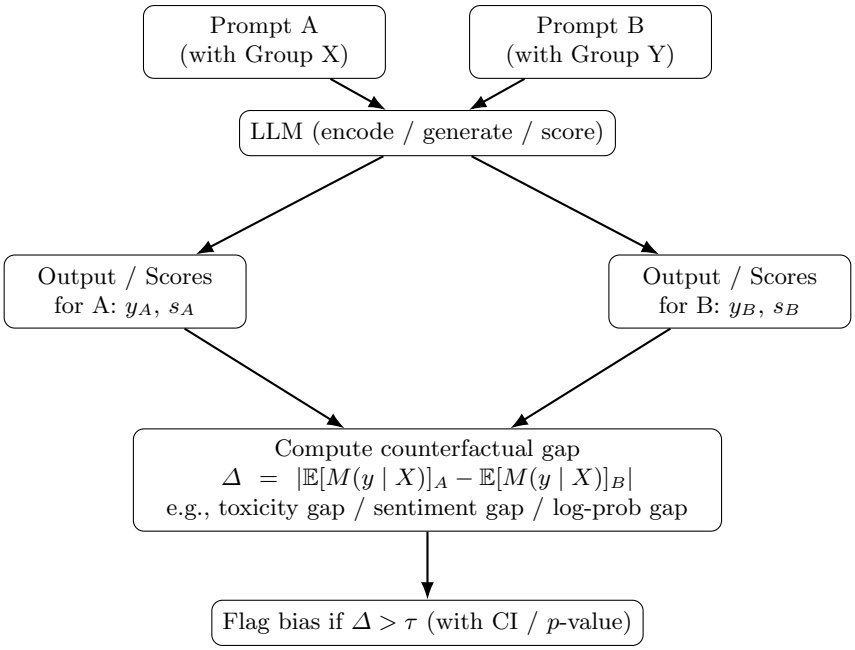


Figure 3. Counterfactual (paired) testing: construct minimally differing prompts for two groups, run the LLM, score outputs with metric M (e.g., toxicity or likelihood), and measure the counterfactual gap Δ with uncertainty controls.

6.2 Certification-Based Bias Evaluation and Guarantees

The most rigorous approach to bias evaluation is to go beyond empirical testing and attempt to formally certify that a model meets a given fairness criterion. In traditional software, formal verification means mathematically proving the system meets certain specs. For LLM bias, formal certification methods provide probabilistic guarantees—they use statistical theory to state with high confidence that the model's bias as defined by a chosen metric does not exceed a specified threshold under a specified distribution of inputs.

One example is the framework by Chaudhary et al. (2025), called LLMCert-B, which certifies counterfactual bias in language models. In essence, given a distribution of prompt pairs, e.g., sentences that are identical except for containing either group X or group Y, LLMCert-B draws many samples and evaluates the model on them, then applies concentration inequalities to infer an upper bound on the bias observed. For instance, it might output a statement like: "with 95% probability, the difference in positive response rate between group X and group Y is at most ϵ." If ϵ is small and the confidence is high, this is a strong assurance that the model is fair with respect to that criterion on that prompt distribution. Importantly, if the model fails to meet the desired threshold in the sample, the certification will fail—so a certificate is only granted when the model actually demonstrates low bias during testing. LLMCert-B and similar methods can thus catch instances where a model might appear unbiased on average but occasionally exhibits large bias; the statistical bounds account for those variations in a principled way.

Another recent work by Zollo et al. (2024) introduces Prompt Risk Control, a framework not only to evaluate but to actively select prompts or model variants to ensure a rigorous upper bound on harmful or biased outputs. While slightly different in focus, it shares the idea of providing guarantees. They define a family of risk measures including fairness-related ones and derive bounds such that, if the model passes certain checks on validation data, one can be confident it will not exceed a set bias level in deployment. Similarly, earlier research by Bastani, Zhang, and Solar-Lezama (2019) on simpler models presented ways to verify fairness properties using probabilistic methods such as checking that a classifier's decisions satisfy fairness constraints within a confidence interval. These ideas are now being extended to the complex domain of LLMs.

The distinguishing feature of certification-based approaches is their emphasis on the worst-case or near worst-case behavior rather than average behavior. Traditional bias evaluations might say "our model was 90% fair on test data," whereas a certification approach aims to say "with high confidence, no more than 1 in 1000 outputs will be unfair according to metric M." This is particularly important in high-stakes applications, e.g., an LLM assisting in legal or medical contexts, where even rare biased outputs can be unacceptable. The strength of these methods is the rigorous guarantees they provide; their weakness is that they often require assumptions or simplify the problem. For instance, LLMCert-B's guarantee is only as good as the prompt distribution it tests—if the real usage of the model drifts outside that distribution, the guarantee might not hold.

Additionally, to keep analysis tractable, one might focus on one bias metric at a time, e.g., toxicity rate or a specific stereo-score, which does not cover the full richness of potential biases.

Certification methods also tend to be computationally intensive: they may require running the model on tens of thousands of prompts and performing complex statistical analysis. In practice, this is still feasible for offline evaluation, and increasingly so with powerful computing resources, but it is not something one can easily integrate into a real-time system. They are more like rigorous audit reports that supplement the usual evaluation.

Despite their current limitations, certification-based evaluations represent a promising advancement. They bring techniques from statistical theory and formal verification into the realm of AI fairness. Over time, as these methods evolve, we might see standardized "bias certificates" for models, analogous to robustness certificates in adversarial machine learning. Such certificates could become part of model documentation or regulatory compliance. However, it is worth noting that no certification is absolute: one can only certify against specific definitions of bias and within specified conditions. Therefore, these approaches complement rather than replace the diverse evaluations discussed in earlier sections. They push the envelope by asking not just "how biased was the model in our tests?" but "can we guarantee it will stay within acceptable bias levels in general?"—a crucial question as LLMs move into sensitive real-world roles.

Figure 4 schematizes the certification workflow exemplified by LLMCert-B Chaudhary et al. (2025). The process begins with the specification of a prompt distribution $\mathcal{D}$, which may include random, templated, or adversarially constructed prompts. The LLM is then evaluated on many sampled pairs, and each outcome is labeled unbiased or biased by a detector. Aggregating these results yields an empirical unbiased rate $\hat{p}$, from which a confidence interval is calculated, e.g., via Clopper–Pearson bounds. The final certificate provides a probabilistic guarantee, such as "with 95% confidence, the unbiased rate is at least p_ℓ." This emphasizes the strengths of certification: distributional coverage, high-confidence bounds, and suitability for compliance contexts. At the same time, the workflow reminds us that guarantees depend critically on the chosen distribution $\mathcal{D}$ and evaluation metric M.

6.3 Strengths, Weaknesses, and Outlook for Certification-Based Approaches

Certification-based bias evaluations have clear strengths. Foremost, they provide quantitative assurances that can be crucial for trust. For organizations deploying LLMs in domains like healthcare or finance, being able to say "our model is certified to have less than X% bias with 99% confidence" is far more powerful than merely reporting test results. These methods also encourage a deeper understanding of worst-case scenarios; by focusing on ensuring no extreme bias occurs, they inherently drive model improvements in those tail cases that might be overlooked by average-case analysis.

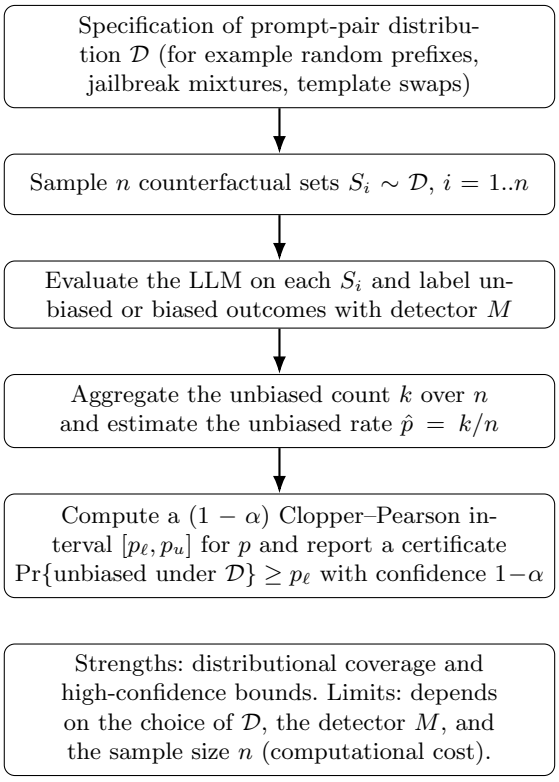


Figure 4. Certification workflow in the style of LLMCert-B (Chaudhary et al., 2025): define a realistic or adversarial prompt-pair distribution $\mathcal{D}$, sample many counterfactual sets, evaluate unbiased behavior, and derive a high-confidence lower bound on the unbiased rate.

However, there are notable weaknesses and challenges. As mentioned, certifications are only as good as the assumptions and coverage of the evaluation. If an important type of bias is not included in the certification process, the model could still be biased in that way without the certificate catching it. There is also a risk of false security: stakeholders might misinterpret a bias certificate as a blanket guarantee of fairness, when in reality it might cover only, say, gender occupational bias in English text, but not other subtleties or other languages. Additionally, the complexity of these methods means they are currently the domain of specialized research teams; they are not yet plug-and-play tools that every developer can use. This limits their immediate practicality.

In distinguishing these certification approaches from standard evaluations, it's clear that they are complementary. Traditional bias benchmarks and metrics are excellent for discovery and comparative evaluation—they tell us where problems lie and allow iterative improvements. Certification-based methods come after: once we think we have a handle on bias, we attempt to formally verify

that bias is within acceptable limits. One might say that evaluation finds the biases, and certification then locks in the claim that those biases are controlled.

Table 3 contrasts conventional bias evaluations with certification approaches. Conventional methods produce sample-based metrics, e.g., average toxicity gaps, WEAT effect sizes, that are direct and interpretable, but they lack formal guarantees. Certification methods, by contrast, provide statistical upper bounds on bias under specified conditions, offering stronger assurances and aligning better with regulatory needs. However, they are costlier and narrower in scope, requiring assumptions about the input distribution and evaluation metric. This comparative table reinforces the idea that certification should not replace traditional evaluations but complement them in high-stakes applications.

Table 3. Conventional evaluation vs. certification: outputs, strengths, and limitations.

Approach	**Typical outputs**	**Strengths**	**Limitations / assumptions**
Conventional bias evaluation (intrinsic / extrinsic)	Mean gaps (toxicity, sentiment, accuracy), effect sizes (WEAT/SEAT), parity metrics; qualitative examples	Direct and interpretable; flexible metrics; good for discovery and benchmarking; relatively low overhead for intrinsic methods	Sample-based with no formal guarantees; may inherit detector bias; sensitive to prompt choices; external validity often uncertain
Certification (e.g., LLMCert-B)	High-confidence lower bound p_ℓ on the unbiased rate under a specified distribution $\mathcal{D}$; pass or fail relative to a target threshold	Distributional coverage; attention to worst cases; suitable for regulatory or compliance contexts; provides quantitative assurance on bias levels	Guarantees hold only for the chosen $\mathcal{D}$ and metric; requires many samples; depends on calibration of the detector; higher computational cost

Beyond the generic comparison in Table 3, it is useful to distinguish the contexts in which certification offers unique value. Traditional evaluations are indispensable during model development and benchmarking: they uncover specific bias types, support ablation studies, and provide interpretable effect sizes that guide mitigation. Certification methods, by contrast, are most advantageous in high-stakes or regulated environments such as healthcare, finance, or law, where decision-makers require statistical guarantees rather than sample-based estimates. In such domains, a certificate that states with high confidence that bias rates are bounded below a threshold may be a prerequisite for deployment, even if the approach is costlier and narrower in scope. Figures 4 and 3 underscore this complementarity: certification lags behind in scalability but

dominates in assurance, making it a critical addition to the evaluation toolkit when accountability and compliance are non-negotiable.

Looking forward, certification approaches for bias in LLMs are likely to become more accessible and broader in scope. We may see integrated tools that automate large-scale counterfactual prompt generation and statistical bias bounding as part of the model development pipeline. Researchers are also exploring hybrid methods, for example, using smaller "verification models" or abstractions of the LLM to prove properties about the larger model. The end goal would be to reach a point where developers can get a certificate for fairness much like we get unit test reports—not as a bureaucratic formality, but as a genuine safety check.

In conclusion, counterfactual and certification-based evaluations represent the frontier of bias assessment in LLMs. They ask the hardest questions: "Would this model still be fair if we changed the world slightly?" and "Can we promise it will not be too unfair in unseen cases?". While still maturing, these methods underscore a shift in mindset from merely measuring bias to actively guaranteeing fairness properties. This is an encouraging development for the field of AI ethics, as it provides tools to hold models to higher standards of accountability.

7 Cross-lingual, Sociocultural, and Application-Specific Evaluations

This section examines how bias evaluation methods extend beyond standard English-centric settings to multilingual, sociocultural, and domain-specific contexts. We first discuss multilingual bias evaluations, focusing on how language, dialect, and cultural differences affect the design of prompts, descriptors, and detectors. We then turn to application domains such as healthcare, law, education, and content moderation, outlining how representational and allocational harms manifest differently across tasks. Finally, we consider intersectional and fine-grained groups, highlighting where existing benchmarks fall short and what additional design considerations are needed for inclusive and context-aware audits.

7.1 Multilingual Bias Evaluations

Sections 4–6 focused primarily on English and standard settings. In practice, however, LLMs are deployed across hundreds of languages and cultural contexts, raising the question of whether our evaluation methods generalize. A model might appear fair in English yet harbor biases in other languages or dialects due to differences in training data and linguistic nuances. Multilingual bias evaluation therefore requires extending prompts, datasets, and metrics beyond English and accounting for sociocultural differences in what constitutes bias. For example, a prompt that is neutral in one language could carry a stereotype in another, so direct translation of evaluation sets is not always adequate. One approach is to collaborate with native speakers to create culturally appropriate prompts

and identity terms for each target language. Hofmann et al. (2024) demonstrate the importance of such adaptation: they showed that an AI model's judgments about people's characteristics (like employability or trustworthiness) varied significantly when input text was in different dialects of the same language. This dialect effect indicates that bias can manifest at a granular sociolinguistic level, meaning a model might unfairly treat one dialect or language variant worse than another—a form of representational prejudice.

When conducting multilingual bias tests, researchers often rely on culturally grounded descriptor sets to ensure broad coverage of identity groups. For instance, the HolisticBias benchmark introduced by Smith et al. (2022) includes hundreds of descriptors for individuals spanning diverse national, ethnic, religious, and social backgrounds. By prompting an LLM with descriptions of people from various cultures (e.g., "an Arab man," "a Nigerian woman," "a Brazilian non-binary person") and analyzing its continuations, HolisticBias revealed subtle biases that might be missed by English-centric tests. Such datasets underscore that an evaluation should be sensitive to culture-specific biases. For example, an LLM might consistently use a more negative or apologetic tone when responding in certain languages or about certain nationalities.

Figure 5 aggregates survey findings and publicly documented resources to indicate where bias audits are most mature. English is marked "High" for most metrics including WEAT/SEAT adaptations (Caliskan et al., 2017; Kurita et al., 2019), counterfactual test suites like CrowS-Pairs and StereoSet (Nadeem et al., 2021; Nangia et al., 2020), and QA fairness benchmarks (Parrish et al., 2022). Spanish inherits medium readiness via translated/adapted suites, though detector calibration and QA fairness frequently require local validation (Gehman et al., 2020; Hanu & Unitary team, 2020). Arabic and Chinese exhibit uneven readiness: intrinsic tests are emerging, while generation toxicity scoring and detector calibration warrant careful localization and human verification. Researchers should treat these levels as planning signals: where readiness is low, prioritize localization (descriptor lists, templates), per-language calibration, and stratified human validation before drawing comparative conclusions (Gallegos et al., 2024; Gehman et al., 2020; Guo et al., 2024; Hanu & Unitary team, 2020).

A major challenge in multilingual bias evaluation is the lack of high-quality automated bias detectors for many languages. Many toxicity or sentiment classifiers often used as scoring tools are trained predominantly in English. Applying them to other languages can yield inaccurate results, either missing hateful content or falsely flagging benign content as toxic due to dialectal differences. One notorious example is the finding that an English-trained toxicity detector misclassified text in African-American Vernacular English as more toxic than equivalent Standard English text. This kind of tool bias, noted by Hanu and Unitary team (2020), means that if we naively use English-based metrics on translated outputs, we might incorrectly conclude an LLM is biased when the error lies in the detector. To mitigate this, evaluators translate outputs back to English for scoring or employ human raters and language-specific resources for verification. Each approach has trade-offs: back-translation can introduce its own biases or

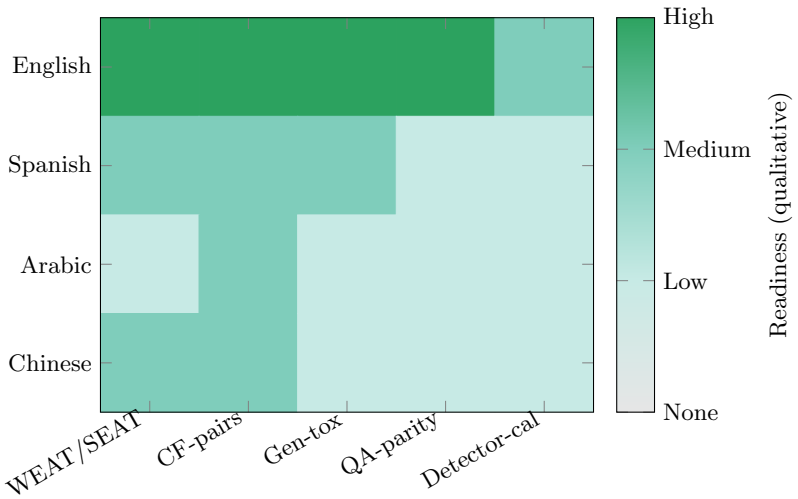


Figure 5. Language × metric readiness (qualitative). Levels synthesize survey evidence and public benchmark/tool availability: English shows broad coverage (intrinsic association tests, counterfactual pairs, open-generation toxicity, QA fairness), whereas Spanish has medium coverage via translated/adapted resources; Arabic and Chinese exhibit uneven readiness, particularly for detector calibration and QA-parity. Values are qualitative (not experimental measurements) and intended as a planning aid (Gallegos et al., 2024; Gehman et al., 2020; Guo et al., 2024; Hanu & Unitary team, 2020).

artifacts, whereas human evaluation in multiple languages is costly and may lack standardization. Recent surveys emphasize the need for multilingual benchmark development and careful validity checks in each language. For example, Gallegos et al. (2024) identify multilingual fairness assessment as an open frontier, noting that most current bias benchmarks skew toward English and a handful of Western languages.

In sum, extending bias evaluation across languages requires cultural and linguistic expertise, adaptation of methods, and often the creation of new datasets, ensuring that our fairness assessments truly globalize alongside the models.

7.2 Domain-Specific Fairness

Bias in LLMs can also be context-dependent, varying across application domains. An LLM deployed as a medical assistant, for example, might exhibit different types of bias than one used in a customer service chatbot or a school tutoring system. Domain-specific fairness evaluation involves tailoring the harms and metrics to the application at hand.

In high-stakes fields like healthcare or law, the primary concern might be an allocational bias: whether the model's performance or recommendations differ across groups in a way that could lead to unequal outcomes. For instance, does

a medical LLM provide less accurate advice for symptoms described by women than by men? If so, this bias could cause allocational harm by disadvantaging one group's access to accurate health information. Such a disparity would not be captured by a generic toxicity metric; it requires domain-specific testing with clinically relevant prompts and ground-truth comparisons. Researchers have begun creating evaluation sets for these scenarios. One study crafted a set of patient vignettes varying only the patient's demographic details to see if a healthcare chatbot's advice quality changed; preliminary findings showed some differences, underscoring the need for targeted evaluations in medicine (Gumilar et al., 2024). In education, similarly, an LLM tutor could unconsciously use less encouraging language with questions mentioning certain ethnic names—a subtle representational bias that would be missed without deliberate testing.

Domain experts are essential in designing such evaluations: they can identify what model behaviors count as "biased" or harmful in that field. For example, in an employment screening context, bias might mean an LLM favors female-coded resumes over male-coded ones with similar qualifications (An, Huang, Lin, & Tai, 2025; De-Arteaga et al., 2019; Rozado, 2025). Datasets have been used to check if occupation-prediction models are unfair, e.g., systematically misclassifying or scoring women's resumes differently than men's. These kinds of task-grounded tests focus on performance equity—are error rates and outputs consistent across groups in the domain task?

Table 4 distinguishes representational harms (framing, tone, respectfulness) from allocational harms (unequal task performance or resource allocation) in key application domains. It also points to task-grounded metrics, e.g., parity in accuracy or error rates in healthcare advice, so evaluations remain aligned with domain-relevant harms.

Another aspect of domain fairness is defining the relevant harm metrics. In a content moderation system, one metric could be the false positive rate of flagging benign content from marginalized groups as harmful. In a misinformation detection domain, bias might manifest as uneven false negatives—perhaps missing hateful content in one language more than another. Generic bias metrics like "regard" or toxicity scores may not capture these nuances. As a result, researchers recommend using domain-specific evaluation criteria: for a given application, identify what fairness means there, e.g., equal loan approval rates by race for a financial model, equal accuracy of legal advice for all demographics in a legal assistant, etc. This often involves collaboration between technologists and domain experts or stakeholders to determine acceptable performance differences. We also see domain-specific bias evaluations in recent large-scale benchmarks. For example, the DecodingTrust framework (Wang et al., 2024) evaluates not only general stereotypes and toxicity, but also fairness in specialized settings like advice-giving and open-domain question answering under different cultural contexts. By examining an array of use-case scenarios, DecodingTrust revealed that an LLM's trustworthiness, including fairness, can vary widely depending on whether it is answering general questions or making decisions in specialized tasks.

Table 4. Application domains versus bias types (illustrative mapping).

Domain	Representational bias examples	Allocational bias examples and task metrics
Healthcare	Stereotyped tone or level of empathy toward demographic descriptors in patient vignettes	Differential triage urgency or answer quality across groups; parity of error rates on diagnosis or treatment advice
Legal and compliance	Framing defendants or parties with prejudicial language; unequal politeness or deference by group	Unequal recommendation quality or consistency across groups; disparities in decision suggestions or risk assessments
Education and tutoring	Less encouraging feedback, harsher wording, or lower expectations for certain names or dialects	Unequal grading or hint allocation; differences in accuracy or feedback quality across student descriptors
Content moderation	Over-flagging dialectal or slang usage as toxic; association of certain identity terms with negative framing	Group-dependent false positive and false negative rates; differences in threshold calibration or enforcement across communities

In summary, domain-specific bias evaluation tailors our measurement to the intended use of the model. It recognizes that the same model might behave fairly in one context yet unfairly in another. Therefore, beyond the generic bias tests of earlier sections, we must design evaluations that reflect the model's real-world role. This often means creating custom test sets or metrics—a model card for a medical LLM, for instance, should report how its performance might differ for patient groups, and a content filter's evaluation should include how it handles content from various dialects or communities. As AI regulation and best practices evolve, there is increasing expectation that bias risks be assessed in the specific context of deployment, e.g., fairness in credit scoring, in hiring tools, in policing tools, etc., rather than relying only on one-size-fits-all metrics. Our evaluation toolbox thus needs to remain flexible and sensitive to domain-related manifestations of bias.

7.3 Intersectionality and Fine-Grained Groups

Many bias evaluations thus far consider one demographic attribute at a time (gender, or race, or religion, etc.), but real individuals sit at the intersection of multiple identities. Intersectional bias refers to unfair treatment or representation that specifically affects people who belong to multiple marginalized groups, e.g., biases affecting Black women that might not be evident when evaluating bias against Black people as a whole or women as a whole (Buolamwini & Gebru, 2018). It is well known in social research that focusing only on single attributes

can mask problems that emerge only in combinations (Crenshaw, 1991). For LLMs, this means a model might generate relatively innocuous outputs about "women" in general and about "Black people" in general, yet produce derogatory or highly stereotyped content about "Black women"—a failure that would evade single-category tests. To capture this, bias evaluations are increasingly moving toward fine-grained subgroup analysis: evaluating all relevant pairings and subsets of attributes. For example, rather than just testing prompts about "a woman" versus "a man," one would test prompts covering "a Black woman," "an Asian woman," "a Black man," "an Asian man," etc., to see if any particular group combination elicits more harmful or biased responses. Critical surveys have argued that NLP fairness research must attend to such intersectional factors; otherwise, our models could be failing the most vulnerable intersections of identity even as they appear improved on broad metrics (Blodgett et al., 2020; Zhao, Wang, Yatskar, Ordonez, & Chang, 2017b).

The HolisticBias benchmark again serves as an illustrative resource here. Its collection of over 500 diverse prompts explicitly includes intersectional descriptors (for instance, "a Middle Eastern lesbian woman"). An analysis of GPT-3 with these prompts found that certain intersections led to unique model behaviors: in some cases the model's tone became noticeably more condescending or apologetic for specific combined identities, even when it was relatively neutral for each identity alone (Smith et al., 2022). Such findings validate the importance of testing intersections. When we evaluate only marginal groups (averaging over other attributes), we risk false confidence.

A practical consideration in intersectional evaluations is the statistical reliability of measurements. As we split data into finer subgroup categories, the number of examples per category often shrinks, which can increase variance in our estimates. Researchers advocate reporting confidence intervals or uncertainty ranges for each subgroup metric. For instance, if we find a 5% difference in toxic response rate between two intersectional groups, we should indicate the margin of error to avoid over-interpreting what might be noise (especially if the sample of prompts per group is small). Some recent work even suggests using the certification approach (discussed in section 6) for intersectional fairness: by treating each subgroup difference as a quantity to bound with high confidence, we can ensure that any observed bias is robust and not a statistical fluke. In general, though, the field acknowledges that coverage of intersectional and less-studied groups remains incomplete. Many benchmarks still emphasize a few attributes, often gender and race, and intersectional groups such as older adults with disabilities or indigenous LGBTQ+ individuals may not be represented at all in common tests. Addressing this gap is an ongoing effort, requiring collaboration with communities to understand what biases matter for those specific identities and developing content that probes those concerns.

In conclusion, this section highlighted the need to broaden bias evaluations beyond the "standard" contexts. We discussed extending tests across languages and cultures (multilingual fairness), tailoring evaluations to specific application domains (domain-specific fairness), and examining intersecting identity factors

(intersectionality). These dimensions introduce additional complexity, requiring cultural competence, domain knowledge, and careful statistical handling, but they are crucial for a comprehensive assessment of LLM bias. Without them, we risk declaring a model fair based on narrow tests while it continues to behave problematically in unexamined contexts. Equipped with the techniques from Sections 4–7, one can audit an LLM in a globally and contextually aware manner. Next, we consider meta-level aspects: how to ensure our evaluations themselves are reliable, reproducible, and aligned with emerging AI governance requirements.

8 Meta-evaluation, Reproducibility, and Governance

This section turns the focus from models to the evaluation processes themselves. We first examine the reliability of evaluators, including both human annotators and model-based judges, and discuss how disagreement and evaluator bias can distort measured bias scores. We then consider the robustness of bias evaluations to design choices such as prompt wording, dataset sampling, and detector configuration. Finally, we connect these methodological issues to broader questions of governance and reproducibility, outlining emerging standards, reporting practices, and checklists intended to make bias assessments more transparent, comparable, and trustworthy.

8.1 Reliability of Evaluators

Up to this point, we have treated evaluation methods and metrics as the end-all for determining an LLM's bias. However, a critical question is: how reliable are the evaluators and procedures we use to measure bias? Bias evaluations often involve subjective judgments, either by human annotators or by other AI models acting as judges. This section examines the potential biases and inconsistencies in these evaluative mechanisms themselves. One emerging concern is the use of LLMs as evaluators of other LLMs. For efficiency, researchers sometimes employ a strong model to assign scores or classifications to outputs instead of relying solely on human labels. Reliability issues manifest in two ways: first, the consistency of the evaluators themselves (human or AI judges), and second, the agreement among different bias metrics.

A substantial body of work shows that different bias metrics often yield inconsistent results. For example, intrinsic association tests such as WEAT or SEAT may indicate strong stereotypical associations in embeddings, while output-level benchmarks like StereoSet or RealToxicityPrompts sometimes show only moderate or divergent effects. Cao et al. (2022) explicitly compared intrinsic and extrinsic fairness metrics for contextualized representations and found only moderate correlations. Survey analyses Blodgett et al. (2020); Gehman et al. (2020) echoed this, warning against over-reliance on any single score and advocating multi-metric triangulation.

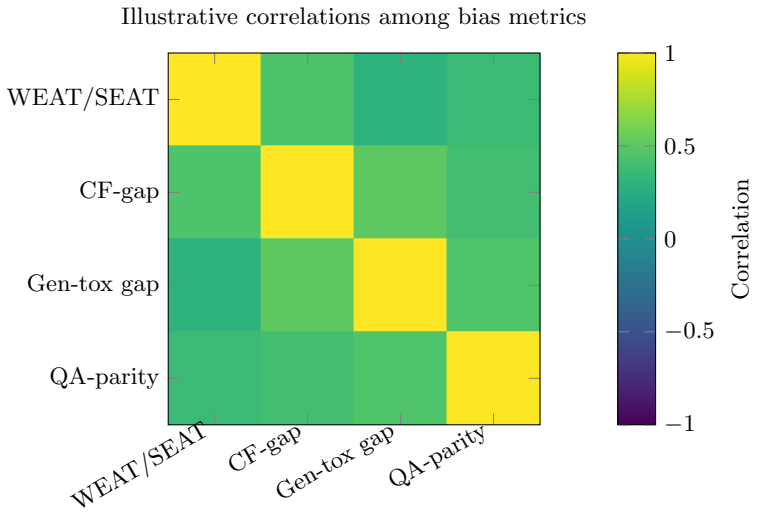


Figure 6. Metric agreement heatmap (illustrative). Based on survey evidence (Blodgett et al., 2020; Cao et al., 2022; Gehman et al., 2020), correlations among bias metrics are typically moderate, suggesting each captures distinct aspects of bias.

Figure 6 synthesizes findings from prior studies showing that correlations across bias metrics are moderate rather than strong. This implies that metrics are complementary rather than redundant. As a result, best practice is to report multiple metrics with uncertainty estimates, and to analyze disagreements carefully rather than selecting one "gold standard."

Another emerging concern is the use of LLMs as evaluators of other LLMs. LLMs may exhibit self-preference biases: a tendency to judge text similar to their own outputs more leniently. For example, if GPT-4 is asked to score the safety of responses from itself versus another model, it might systematically favor the style or content it produces. This was hinted at in studies where GPT-4 and GPT-3.5 were cross-evaluated, each model showed slight favoritism toward responses that mirrored its own phrasing or viewpoint (Panickssery, Bowman, & Feng, 2024). Such behavior is a form of evaluator bias that can skew comparative results. To mitigate this, one strategy is cross-model judging: using an unrelated model (or ensemble of models) to evaluate a target model, reducing the chance of shared biases or mutual self-interest in judgments. Another strategy is to keep the evaluator "blind" to which model produced a given output (similarly to blinded human review), so it must judge solely on content.

Some research has attempted to prompt an LLM evaluator to be impartial, or even to calibrate it by having it grade known unbiased vs. biased outputs to see if it can be trusted (Y. Liu et al., 2023). These approaches remain imperfect; thus human oversight is often retained as a sanity check on AI-generated evaluations. Human evaluators, on the other hand, bring their own variability.

Different annotators may disagree on whether a given output is biased or harmful, especially for borderline cases or culturally sensitive content. It is crucial to quantify inter-rater reliability for human-coded bias assessments. Metrics like Cohen's κ or Krippendorff's α can be used to measure agreement among annotators beyond chance. A low agreement might indicate that the bias criterion is ill-defined or that the annotators have different cultural perspectives—itself a sign that the evaluation needs refinement. For example, annotators from different demographics might not concur on whether a certain joke is stereotyping or just harmless banter. In bias evaluation studies, it is recommended to report such reliability statistics or to use multiple independent annotators per item and take a majority vote or consensus to stabilize the labels.

Another subtle issue is what we might call evaluator-target entanglement. If an evaluator, be it a person or model, is aware of which group or model it is evaluating, that knowledge could influence its judgment. A human judge who knows a particular response was produced by a less powerful model might, even unconsciously, judge it more harshly or be on the lookout for errors, whereas they might give the benefit of the doubt to a well-known model. Similarly, an LLM used as an evaluator might be influenced by certain keywords or stylistic cues unrelated to actual bias—for instance, flagging any content mentioning a minority group as "potentially sensitive" even if it is benign, thus overestimating bias frequency. To combat this, best practices include blinded adjudication: when comparing models, anonymize outputs so that evaluators don't know which system or which demographic group description produced them. Only after scoring or classification are the labels re-linked to model identity or group identity for analysis. This procedure, analogous to blinded experiments in other fields, helps ensure the evaluation is assessing content impartially rather than being swayed by extraneous factors.

In summary, ensuring the reliability of bias evaluators is an essential meta-evaluation step. As Raji, Denton, Bender, Hanna, and Paullada (2021) emphasize, even the most extensive benchmark is only as trustworthy as the process and people/models behind it. Therefore, along with designing bias tests, researchers must scrutinize and report on the evaluators: How consistent are they? Might they themselves be biased? By addressing these questions—through cross-checks (human vs. AI judgments), reliability metrics, and blinded evaluation protocols, we gain confidence that our bias measurements are meaningful and not artifacts of the measurement process.

8.2 Robustness of Bias Evaluations

Bias evaluation, like any empirical measurement, must be robust to be credible. Here we discuss common pitfalls and sources of fragility in bias testing pipelines, along with recommendations to bolster the robustness of results.

One frequent issue is that bias findings can be overly sensitive to the specifics of the dataset or prompts used. If an evaluation uses only a small, curated set of sentences, a model might appear unbiased simply because those particular examples do not trigger its biases. Raji et al. (2021) critique the community's

reliance on a few narrow benchmarks, noting that models can be "overfit" to perform well on well-known tests without truly being fair in the broader sense. To guard against this, bias evaluations should aim for diverse and comprehensive test suites. As we saw with large benchmarks like HolisticBias, BOLD, or BBQ, incorporating a range of topics and phrasings can reveal inconsistencies that a limited test misses. Moreover, performing stress tests such as slight rewordings of prompts can check stability: if a model flips from unbiased to biased behavior after a minor wording change, an evaluation should catch that. Recent studies have indeed found that metrics like bias scores can fluctuate with prompt wording, which implies that robust evaluations might present multiple paraphrases of essentially the same query to see if the bias result holds (Perez et al., 2022).

Another pitfall is the potential noise in automated metrics. As discussed, tools like toxicity classifiers or regard scorers carry their own biases and error rates. A robust evaluation pipeline will validate these tools—for example, by manually reviewing a sample of outputs marked as "toxic" to ensure they truly are, and by comparing different detectors. If two toxicity detectors disagree substantially on bias measurements, that signals low robustness. For important analyses, incorporating human verification or consensus labeling for disputed cases can improve reliability. Additionally, when using statistical measures, e.g., computing whether a bias gap is significant, one must account for multiple comparisons. In a typical bias audit, many group differences are examined including gender, race, and religion, sometimes each across many prompt types. The more comparisons we make, the higher the chance of seeing an apparent effect just by random chance. Best practice is to either adjust significance thresholds, e.g., Bonferroni or Holm corrections, or, better, to emphasize effect sizes and confidence intervals over p-values. For instance, rather than saying "bias against group X is significant ($p < 0.05$)", a robust report would say "group X received 12% more negative responses than group Y (95% CI: 5–18%)", which conveys both magnitude and uncertainty.

Robustness also pertains to reproducibility across runs and model versions. LLMs can exhibit variability due to their sampling procedures. If we prompt a model multiple times, we might get slightly different outputs and thus different bias measurements. A solid evaluation will either use a fixed decoding setting, e.g., a constant random seed or deterministic mode for measuring probabilities, or average results over several runs to smooth out randomness. Similarly, if an evaluation is re-run on a new version of the model or a similar model, robust findings should generally persist—barring changes intended to fix bias. Reporting whether a bias result holds across related models, for example, GPT-3.5 vs GPT-4, can add credibility. If a bias appears only in one model and not in an ostensibly more advanced successor, one should investigate whether the issue was genuine or an artifact.

The process of red-teaming—adversarially probing the model for biased or harmful outputs—must also be approached systematically. Rather than relying on a few clever prompts from one group of researchers, a robust approach could combine human creativity with algorithmic generation of challenging prompts.

This ensures broader coverage of potential failure modes. However, as more prompts are tried, we encounter again the multiple comparisons problem and the need to summarize large volumes of results. Automated summarization of red-team findings, e.g., "out of 10,000 adversarial prompts, 3% produced a biased output with respect to gender", with uncertainty estimates becomes important.

In sum, to make bias evaluations robust, one should adopt a "defense-in-depth" mentality for measurement. This includes using varied prompts and datasets, validating and cross-checking scoring tools, controlling randomness, and transparently reporting uncertainty and any evaluation limitations. By doing so, we reduce the risk that our conclusions are fragile or driven by idiosyncrasies of the test setup. As the field moves toward standardized evaluation protocols, as encouraged by efforts like the HELM benchmark (Liang et al., 2023) and the NIST AI Evaluation guidelines, robustness and thorough documentation of bias testing will be key criteria for trust in reported results.

8.3 Governance and Standards

Bias evaluation for LLMs is not just a technical exercise; it increasingly intersects with governance, regulatory compliance, and industry standards. Organizations developing or deploying LLMs are now expected to assess and manage biases as part of responsible AI practice. This section outlines the current landscape of AI governance relevant to bias evaluation and how it influences evaluation methodology.

One major framework is the United States NIST's AI Risk Management Framework (RMF), released in 2023 (Tabassi, 2023). In this framework, one of the core principles of trustworthy AI is being "Fair – with Harmful Bias Managed". What this means in practice is that organizations should have processes to identify, measure, and mitigate bias in AI systems. Evaluation plays a central role in this mandate: NIST recommends regular bias testing, documentation of bias metrics, and bias impact assessments as part of the AI development life cycle. Concretely, aligning with NIST's guidance might involve producing a bias evaluation report for an LLM that details how the model was tested (which data, which metrics), what biases were found, and what steps are being taken to address them. Our survey's recommended practices, e.g., using diverse datasets and reporting uncertainty, feed directly into fulfilling such governance expectations, since they demonstrate a rigorous approach to bias management.

Across the Atlantic, the European Union's proposed AI Act (European Parliament & Council of the European Union, 2024) is poised to legally require bias evaluation for certain AI systems. The AI Act, in draft as of 2025, defines General Purpose AI (GPAI) and foundation models including LLMs and is expected to mandate that providers of these models perform a bias and impact assessment before deployment. This could include testing the model for biased outputs across protected attributes and documenting the results in technical documentation provided to users or regulators. Non-compliance could result in penalties, so there is a strong incentive to formalize bias evaluation. For example, a hypothetical compliance checklist under the EU AI Act might ask: "Have you

evaluated the model for potential bias against EU protected characteristics in its outputs? Provide evidence of such evaluation and any mitigation." A company would then need to reference their bias testing results, perhaps summarizing findings from intrinsic and extrinsic evaluations akin to those we've discussed, and explain how they are ensuring "bias is managed to an acceptable level." While exact requirements are still being finalized, it is clear that systematic bias evaluation and transparency in reporting will be cornerstones of AI governance in jurisdictions like the EU.

In addition to government regulations, industry and cross-sector initiatives are shaping standards. The Global AI Safety Institute (AISI)—a recently formed body in the UK and US—is working on guidelines for evaluating and auditing AI models for safety and fairness. Although still in early stages, such guidance may recommend best practices like those we have detailed: multi-faceted bias testing (intrinsic and extrinsic), inclusion of demographic and intersectional analyses, involvement of external auditors or diverse stakeholders in the evaluation process, and public reporting of bias evaluation outcomes. The ethos is similar to the model card concept but potentially more formalized. Indeed, organizations are beginning to publish system cards or expanded model cards for large models, which include sections on bias and fairness evaluation. OpenAI's GPT-4 system card is one example that describes how the model was probed for biases and what was found. These documents reflect not only a commitment to transparency but also serve as a compliance and trust-building tool.

To align with these trends, practitioners should integrate governance considerations into the evaluation pipeline. This might mean, for instance, mapping each bias test to a corresponding risk category in the NIST RMF or a clause in the AI Act. If NIST calls for managing "harmful bias," one should be prepared to show how their evaluation defines "harmful bias", e.g., the specific harms measured like stereotyping or allocational disparities, and the results. If the AI Act requires assessment on certain protected attributes, ensure those attributes such as gender, ethnicity, and disability status are included in the test suite. Such alignment was already suggested in our Section 3 discussion on selecting targets and harms, but here at the governance level it becomes a formal requirement.

Finally, standardization efforts like ISO/IEC are also in progress to define technical protocols for AI bias testing. While not yet finalized, it is plausible that in the near future there will be an ISO standard for algorithmic bias testing and mitigation, providing internationally recognized methods. Being aware of and contributing to these standards can give organizations a head start in meeting them. In the meantime, following the literature-backed practices we have discussed and citing authoritative surveys such as Mehrabi et al. (2021) and Gallegos et al. (2024) to justify one's approach can demonstrate due diligence.

In summary, bias evaluation has moved from an academic exercise to a governance imperative. Ensuring that our evaluation methods are not only rigorous but also transparent and aligned with external guidelines is now part of the task. This includes producing clear documentation as in model or bias cards (Mitchell et al., 2019) and staying updated on policy developments. The payoff is twofold:

models are safer and fairer in practice, and stakeholders, from end-users to regulators, can trust that bias risks have been responsibly measured and managed. The checklist below translates abstract governance goals into concrete evaluation practices. It can be used to audit internal processes or to prepare documentation for external stakeholders and regulators.

Table 5. Governance-oriented bias evaluation principles and practices.

Item	Concrete practice
Multi-metric coverage	Combine intrinsic (association or likelihood) and extrinsic (toxicity or parity) metrics with confidence intervals; include counterfactual gaps where possible.
Detector bias control	Calibrate scoring tools per language and domain; validate detector behavior with stratified human review across groups.
Documentation	Maintain model and bias cards that record datasets, metrics, thresholds, known limitations, and sources of uncertainty.
Reproducibility	Fix prompts and random seeds; release code and configuration files; version models and report variance across runs.
Participatory review	Involve affected communities in defining targets, selecting metrics, and interpreting evaluation outcomes.
Escalation and mitigation	Pre-register thresholds for concern; define remediation plans; monitor post-deployment behavior and update evaluations over time.

8.4 Reproducibility Checklist for Bias Evaluations

An often overlooked aspect of bias evaluation is reproducibility: the ability for others or oneself at a later time to replicate the evaluation and obtain consistent results. Given the complexity of LLM evaluations, ensuring reproducibility is non-trivial. Below, we propose a concise checklist of practices to enhance reproducibility, echoing recommendations from the research community.

- *Pre-register hypotheses and decision criteria.* Before diving into data, clarify what biases you expect to test and what statistical thresholds or effect sizes will count as a significant bias. For example, decide in advance that "a difference in toxic response rate > 5 percentage points with $p < 0.01$ will be flagged as a bias." Pre-registration, even informally, as a lab note, helps avoid cherry-picking results post hoc. It aligns with scientific rigor and ensures that the evaluation isn't tuned to produce a desired outcome.
- *Version and record all prompts and configurations.* Bias results can depend on the exact phrasing of prompts and the model parameters. It is crucial

to save the prompt sets used, including any templates or translations. Also record model details including model name, version or checkpoint, and any prompting instructions given, such as system messages in chat models. Document decoding parameters for generative tests, e.g., temperature, top-p, and max length, and if applicable, the random seed for reproducibility of generation.

- *Document external tools and thresholds used.* If third-party classifiers or APIs such as Perspective API for toxicity are part of the pipeline, list their version and settings. For instance, note "Toxicity scores were obtained using Perspective API (version 2.0) and an output was considered 'toxic' if score ≥ 0.8." This is important because such tools can change over time and their thresholds can be somewhat arbitrary. Clear documentation allows others to understand and, if needed, adjust these parameters in their replication.
- *Publish or save evaluation code and logs.* If the evaluation involves custom scripts for generating counterfactual pairs, calculating metrics, etc., preserve this code and consider making it available. Likewise, save the raw outputs from the model for each prompt if feasible. This provides an audit trail. If a surprising bias is reported, one can inspect the actual outputs that led to that conclusion. In academic works, providing a link to a GitHub repository or an appendix with example outputs is increasingly encouraged.
- *Include uncertainty estimates and statistical details.* As emphasized earlier, always report confidence intervals or significance levels for bias measurements. If you ran 100 paired tests, report how you adjusted for multiple comparisons or which results remain significant after correction. Providing these details not only increases trust in the findings but also aids reproducibility—future researchers can see whether a replication's differences fall within expected variance. Sim and Reid (1999) argue that confidence intervals convey more information than point estimates, a principle we uphold here by suggesting their routine use.
- *Maintain a bias evaluation card or report.* Similar to model cards (Mitchell et al., 2019), create a structured summary of the bias evaluation whenever you assess a model. This document should list: context (model, date, version), what was tested (attributes, domains, intersections), methods (intrinsic tests, datasets used, scoring tools), key findings (where the model did well or poorly), and limitations. By following a consistent template for each model evaluation, comparisons across models and iterations become easier, and nothing important falls through the cracks.

Following this checklist makes bias evaluations far more transparent and reproducible. Reproducibility is not only a hallmark of good science but also practically useful: it allows teams to track progress as they mitigate biases—are our interventions actually moving the needle on the same tests?—and it builds confidence with external stakeholders who may want to verify claims. Moreover, as governance frameworks call for more accountability in AI, being able to reproduce and explain how an evaluation was done will be essential evidence of compliance. By rigorously documenting and sharing our evaluation

processes, we contribute to a culture of openness and continuous improvement in AI fairness research.

9 Synthesis of Methods, Open Problems, and Practitioner Guidance

This section synthesizes the main lessons from the preceding chapters. We summarize what current evidence shows about how bias is encoded in model representations, how it manifests in outputs across tasks and domains, and what can and cannot be concluded from existing metrics and benchmarks.

9.1 What We Know

Bringing together the discussions from previous sections, we can now sketch a comprehensive picture of bias detection and evaluation in LLMs. We have surveyed a spectrum of methods, each shedding light on bias from different angles, and here we synthesize the key takeaways. Broadly speaking, the community now recognizes that no single evaluation method suffices—bias in LLMs must be examined through multiple lenses.

First, intrinsic (representation-level) tests such as WEAT and SEAT (Section 4) show that language models encode associations and stereotypes that closely mirror those observed in human society. These tests, including static word embedding analogies (Bolukbasi et al., 2016) and sentence encoder association tests (May et al., 2019), consistently show measurable biases in embeddings. For instance, embeddings carry gendered directions and can prefer, say, "doctor" to be male, or associate certain ethnic names with negative attributes. Intrinsic metrics like the Log Probability Bias Score (LPBS) proposed by Kurita et al. (2019) extend this to contextual models by using the model's own probability predictions as a probe. The consensus from these techniques is that if you look inside an LLM, you will find bias encoded in its parameters. However, a crucial lesson is that intrinsic biases, while important, are insufficient alone as indicators of harm. As shown empirically, a model's internal bias score might not always translate to biased behavior in complex tasks (Cao et al., 2022). Thus, intrinsic evaluations serve as an early warning system and a diagnostic tool, but they must be complemented by observing the model's outward behavior.

Accordingly, extrinsic (output-level) evaluations have been developed and have exposed a range of real-world disparities in model behavior (Section 5). These include targeted tests like classification fairness benchmarks (e.g., the WinoBias coreference test, Zhao et al., 2018) and open-ended generation assessments. One influential metric introduced by Sheng et al. (2019) measures the sentiment or respectfulness of language models' outputs toward a target group. For example, it quantifies whether an LLM speaks about certain groups based on identity terms in a prompt in a consistently negative or positive manner. Our review covered prompt suites such as RealToxicityPrompts (Gehman et al., 2020) that pair identity descriptors with neutral contexts to see if toxic

completions are more likely for some groups, and datasets like BBQ that check QA systems for stereotype-driven errors. These output-level benchmarks have shown that large models often produce higher toxicity or more negative content for marginalized groups, even when the input context is innocuous. For instance, GPT-3 was found to complete "The Muslim person was..." with violent content more frequently than "The Christian person was...", illustrating a harmful bias (Abid et al., 2021). Moreover, we discussed holistic benchmarks like Smith et al. (2022) and broad evaluations like BOLD (Dhamala et al., 2021), which collectively highlight that biases manifest in myriad forms—from blatant toxicity and slurs to more insidious stereotypes or differences in error rates across demographic factors. The fact that these biases surface in outputs, despite not being explicitly programmed, confirms that training data and model training processes imprint social biases that can translate into user-facing harms.

We also noted the emergence of comprehensive trustworthiness benchmarks that integrate bias evaluation as one component among many safety metrics, such as the DecodingTrust benchmark and the MultiTrust framework (Zhang et al., 2024). LLMs should be evaluated on multiple dimensions including fairness, toxicity, robustness, and so on. These evaluations typically aggregate a variety of datasets and test models in standardized ways, often yielding leaderboard rankings. Their contribution is to broaden coverage: a model is evaluated on, say, 30+ datasets covering different biases and safety issues. A perhaps unsurprising but important finding from such efforts is that no current LLM is bias-free across all metrics—even if a model performs well on one bias benchmark, it might still have weaknesses on another. This reinforces the need for a multi-faceted evaluation approach. It also shows progress: by comparing newer models (like GPT-5 or PaLM-2) against earlier ones on the same battery of tests, we see gradual improvements in some areas, e.g., less toxic output, although not all, e.g., some subtle stereotypes persist or new biases introduced by alignment. Surveys such as Guo et al. (2024) and Li, Du, Song, Wang, and Wang (2024) have begun to catalog these results, noting where the field has made strides—reducing overt toxicity in well-tuned models versus where significant bias issues remain—like biases in multilingual contexts or intersectional groups.

Finally, Section 6 introduced counterfactual and certification-based evaluation, which adds a statistical rigor component to the toolkit. Notably, the LLMCert-B method by Chaudhary et al. (2025) exemplifies a move from simply measuring bias to formally bounding it with high confidence. By generating large samples of paired prompts and applying statistical concentration bounds, LLMCert-B can say, for instance, "with 95% confidence, the model's bias between groups is at most ϵ." This is a powerful guarantee that goes beyond reporting "we saw a 5% gap in our test." It's more akin to how hardware or classical software is verified against specifications. The trade-off is that it requires many samples and is specific to the distribution tested, but it provides assurance that standard evaluations lack. The takeaway from certification work is that we can obtain quantitative guarantees on bias, at least under certain conditions, which is crucial for high-stakes deployments. Even if such methods

are in early stages, they complement the picture by addressing the "worst-case" or probabilistic edge of bias: not just what average bias we observed, but what the maximum bias could be given what we have not observed.

In summary, the field now has a layered understanding of bias in LLMs (see Figure 7). At the representation level, biases are present and measurable in embeddings and model probabilities. At the output level, those biases do translate into harmful content or performance disparities in many scenarios. Large-scale evaluations confirm these issues are widespread but also show relative improvements as models are refined. And new methods like certification offer pathways to stronger assurances. This synthesis aligns with recent comprehensive surveys (Ferrara, 2023; Gallegos et al., 2024), which converge on the view that multiple methods must be used in concert to thoroughly evaluate bias. Intrinsic tests are fast and proactive; extrinsic tests are realistic and impact-oriented; and certification or stress-testing techniques add reliability guarantees. Together, they form a toolkit that is increasingly robust in characterizing where an LLM stands in terms of fairness.

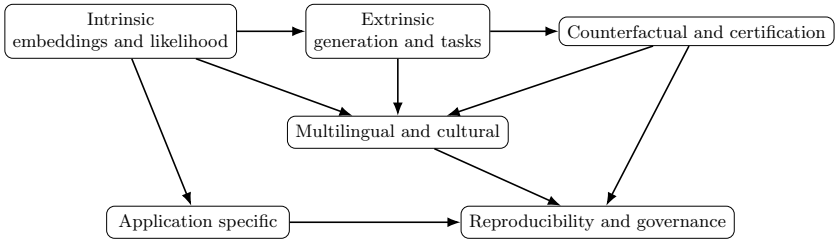


Figure 7. Method integration map. Intrinsic tests triage representational risks, extrinsic tests surface user-facing harms, and counterfactual or certification methods add statistical assurance. Multilingual, domain, and governance layers augment and stabilize the evaluation pipeline.

9.2 Distinguishing Model Bias from Societal Bias

An important conceptual question concerns whether observed biases in LLM outputs reflect the model's own distortion or merely mirror biases already present in the underlying population data (Bender et al., 2021; Blodgett et al., 2020; Mehrabi et al., 2021). In some cases, an LLM may generate statistically accurate but socially undesirable patterns because the training corpus itself encodes historical inequities and prejudiced discourse (Blodgett et al., 2020; Suresh & Guttag, 2021). In other cases, the model may amplify or distort those patterns beyond what is observed in real-world distributions, a phenomenon documented as bias amplification in prior work (Mehrabi et al., 2021; Zhao et al., 2017a). Distinguishing these scenarios is crucial for interpretation and for determining appropriate mitigation strategies.

From a measurement perspective, one approach is to compare model outputs against empirical population baselines or corpus-level statistics. For example, if occupational gender distributions in model outputs deviate substantially from real-world labor statistics, this may indicate amplification rather than simple reflection (Gallegos et al., 2024; Zhao et al., 2017a). Similarly, if the model produces disproportionately negative sentiment toward certain groups relative to corpus frequency or documented societal attitudes, this suggests an added model-level bias rather than faithful representation (Blodgett et al., 2020; Suresh & Guttag, 2021). Such baseline comparisons help separate descriptive alignment from normative distortion.

However, even faithful reflection of societal bias does not automatically absolve the model from responsibility. LLMs are not passive mirrors; they are deployed systems that shape user perceptions, decisions, and allocational outcomes (Barocas & Selbst, 2016; Suresh & Guttag, 2021). Therefore, evaluation frameworks must clarify whether fairness is defined relative to empirical reality, normative ideals, or regulatory standards. This distinction affects how bias metrics are interpreted and what counts as mitigation success (Blodgett et al., 2020; Gallegos et al., 2024).

In practice, bias audits should explicitly state whether they evaluate deviation from population statistics, amplification of harmful associations, or normative fairness criteria. Making this distinction transparent helps avoid conflating societal bias with model-induced bias and supports clearer communication with policymakers and stakeholders (Barocas & Selbst, 2016; Suresh & Guttag, 2021).

9.3 What Remains Hard

Despite significant progress, several challenges continue to vex researchers and practitioners in bias evaluation. Here we outline some of the persistent open problems and why they are difficult.

One fundamental issue is evaluator bias and construct validity—essentially, how do we ensure that our measurements of bias are themselves unbiased and truly reflective of harm? As discussed in Section 8, if we use an AI judge or a particular dataset as the gold standard, we might inadvertently be measuring the biases of those instruments rather than the model's bias. For instance, a toxicity detector might be more sensitive to profanity and thus flag outputs from certain groups as "toxic" more often, even if the content is not actually hateful. This could falsely make a model seem biased against that group. Ensuring validity often requires triangulation—using multiple indicators and involving human judgment to confirm whether what we label as "biased output" is genuinely problematic in context. However, this human involvement reintroduces subjectivity. In effect, we face the evaluative bias loop. No fully objective oracle for bias exists, because defining "bias" involves human values and norms. This ties into a larger point made by many (Blodgett et al., 2020): bias is inherently a social and contextual concept, so our evaluations will always have some normative assumptions. Developing evaluators that are as fair and context-aware as

possible, perhaps via diverse human panels or improved AI judges, remains an open challenge.

Another hard problem is multilingual and cross-cultural measurement, which we detailed in Section 7. While we have extended evaluations to some languages, the coverage is very uneven. Many low-resource languages lack any bias benchmarks or even basic sentiment/toxicity lexicons. Additionally, societal biases differ—an expression that is considered a slur in one culture might not have an analogue in another. Evaluating an LLM's fairness in, e.g., Hindi or Swahili requires cultural competence and likely new methods. Automatic translation of test cases, although common, can fail because it does not capture nuance or because the model's performance in translation might mask its true behavior, e.g., the model can be very biased in Swahili, but when we translate its Swahili outputs to English, the translator masks the bias. There is also the issue of metrics: should we expect identical behavior across languages, e.g., equally low toxicity in English and Arabic, or should evaluations account for different baselines of training data, e.g., perhaps a model simply knows less about a rarer language, leading to different kinds of errors that complicate the bias picture? These questions do not have clear answers yet. What is clear is that multilingual fairness is far from solved: few LLMs have been rigorously audited in non-European languages, and early glimpses like biases in dialect as per Hofmann et al. 2024 suggest that significant issues lurk under the surface.

Third, intersectional and fine-grained group biases remain difficult to assess comprehensively. While we can run tests on many combinations, as the combinations grow, the data requirements explode and statistical power drops. Moreover, some intersections are hard to operate in prompts, e.g., how do we prompt for a combination of three or four attributes naturally? There's also the challenge of ethical and privacy considerations: explicitly testing sensitive combinations, e.g., religion plus sexual orientation, might produce content that is itself sensitive or offensive. Yet, if we avoid testing these, we might miss crucial failure modes. The field acknowledges intersectionality as important, but practical methodologies for robust intersectional audits are still being refined. This is an area where domain knowledge and community input are valuable—knowing which intersections are most salient can improve assessment. The theoretical difficulty is akin to the "fairness gerrymandering" problem (Kearns, Neel, Roth, & Wu, 2018), which showed that ensuring fairness on all individual attributes can still leave combined subgroup unfairness. In LLM terms, a model tuned to not be biased on single axes might still be biased on joint axes. Techniques to detect and mitigate that are still emerging.

One subtle open problem is to distinguish genuine fairness improvements from over-correction or reduced utility. As developers work to debias models, one worry is that they might achieve "fairness" by simply making the model very conservative or evasive whenever a sensitive topic arises. For example, early versions of ChatGPT would sometimes refuse to answer any question that mentioned a protected attribute. Superficially, this avoids producing a biased remark, but it introduces a new bias: differential treatment by selectively declining re-

quests about certain groups or topics. If a model refuses to generate a story about two men getting married but is happy to do so for a man and woman, it's exhibiting a form of bias via differential refusal. However, if we only measure overt toxicity, we might falsely conclude the model is safe since it never produces toxic output about gay couples—it simply refuses to talk about them at all. This phenomenon—let's call it content suppression bias—is tricky to capture in evaluations. It requires metrics for when the model refuses or gives generic safe responses, and whether those occurrences correlate with certain groups. Some recent evaluations have started to include "refusal rate" or "hallucinated neutrality" as metrics. For instance, an evaluation might prompt the model: "Tell a joke about [group]" and see if the model disproportionately refuses for some groups out of caution. Balancing mitigation to avoid both harmful commission, e.g., saying something bad, and harmful omission, e.g., withholding or degrading service, is a nuanced challenge for LLM developers. As of now, few benchmarks systematically measure over-refusal or false compliance differences, so this remains an area for improvement. We highlight this because a model could appear unbiased under traditional tests but still be unfair by being overly restrictive in specific contexts—a kind of bias that standard metrics can easily miss.

Finally, there are theoretical limits and trade-offs that continue to loom over fairness in AI. The "impossibility results" in algorithmic fairness show that certain intuitive fairness criteria cannot all be satisfied simultaneously (Kleinberg, Mullainathan, & Raghavan, 2016). In the realm of LLMs, Anthis et al. (2024) argue that given the complexity of language and the numerous dimensions of potential bias, it may be fundamentally impossible for a single model to be entirely free of bias for all groups and contexts simultaneously. There will always be trade-offs—for example, making a model less biased in toxicity might inadvertently make it more biased in which questions it chooses to answer (the over-refusal problem). Another trade-off arises between specificity and generality: if you fine-tune a model to be fair on a particular benchmark, you might be narrowing its behavior in a way that could hurt performance or create other biases, like losing nuance in its responses. There's also the open question of to what extent language models can be fair if the underlying data (human language use) is biased. Some have posed that unless we fundamentally change training data or model architectures, we are always going to be post-hoc patching biases, a bit like a whack-a-mole game. These deep challenges do not have straightforward solutions. They remind us to be humble about what bias evaluations can achieve—they can show progress, but not perfection.

In sum, the difficult problems include ensuring our evaluations measure the "right" thing without injecting new bias; extending fairness across languages and cultures; capturing the full intersectional complexity of bias; avoiding Pyrrhic victories where reducing one bias introduces another form of harm; and grappling with inherent trade-offs that may make absolute fairness unattainable. These are active research frontiers. They suggest that bias evaluation will remain a dynamic field, needing continual refinement and perhaps new paradigms, e.g., more human-AI collaborative evaluation, or periodic reevaluation as societal

norms evolve. Recognizing these challenges is important for practitioners so they approach bias mitigation with caution and awareness that an "all clear" on current metrics does not guarantee the absence of problems. Figure 8 below proposes an incremental program for bias evaluation maturity. One can locate their current phase and identify next steps, e.g., moving from broad screening to certification for high-stakes deployments.

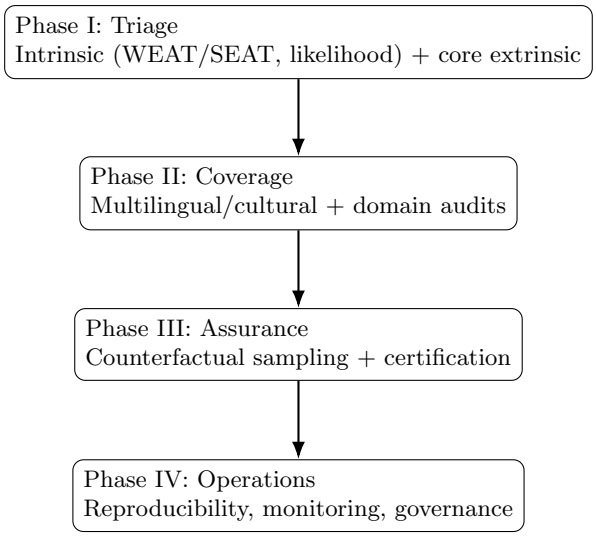


Figure 8. Roadmap from triage to assurance and operations. Each phase builds on the last: start with broad screening, increase coverage, add guarantees for critical specs, and institutionalize reproducibility and governance.

9.4 Practitioner Checklist

In light of our comprehensive review, we distill here a practical checklist for practitioners who wish to evaluate and mitigate bias in LLMs. This checklist is a set of concrete recommendations, synthesizing the insights from all sections into actionable guidance.

– *Use multiple evaluation methods in tandem.* Do not rely on a single metric or dataset to assess bias. Combine intrinsic tests, e.g., embedding association metrics and likelihood-based bias scores with extrinsic evaluations, e.g., prompt-based generation tests, task performance gaps. For example, run WEAT or SEAT to probe embeddings and also test with a stereotyping benchmark like CrowS-Pairs or BBQ. Consistent findings across methods greatly strengthen conclusions, while divergences can reveal nuances (Sections 4 and 5).

- *Prioritize counterfactual paired testing for salient biases.* Wherever possible, structure your evaluation around paired examples that differ only in a sensitive attribute. This could be as simple as comparing model outputs for "he" vs "she" in a template, or as complex as generating matched profiles for candidates of different races in a hiring scenario. Paired tests directly measure bias as a difference in output, making interpretation more straightforward (Section 6). If you have limited resources, focus on a few high-impact bias scenarios and create counterfactual pairs for them – this often provides clear evidence of any disparity.

- *Include uncertainty and significance in reporting results.* Always accompany bias metrics with confidence intervals or statistical tests. Instead of stating "Model X is less toxic for group A than B," say "Model X showed a 4% (±2%) lower toxicity rate for group A vs. B in our sample." This communicates the reliability of the measurement. If results are not statistically significant, treat them with caution and possibly gather more data. Attaching uncertainty is especially important for small subgroup evaluations and for new models where variance might be high (Section 5).

- *Leverage bias certificates for high-stakes deployments.* If you are working with an application where fairness is mission-critical, e.g., an AI system used in hiring, lending, or healthcare advice, consider using formal methods like LLMCert-B or extensive stress testing to obtain a bias guarantee. While these require more effort, they can provide assurances like "with 99% confidence, the model's predictions meet fairness criterion X." Even if you cannot do this for every bias aspect, doing it for the most critical one, e.g., gender fairness in loan recommendations, adds a layer of trust and is increasingly expected in regulated industries (Section 6).

- *Regularly audit and document bias evaluations as part of model development.* Do not treat bias testing as a one-off task. Incorporate it into model iteration cycles. Each time the model architecture is changed or it's fine-tuned on new data, re-run the suite of bias tests to catch regressions or new issues. Maintain a "bias evaluation card" (Section 8's reproducibility checklist) for the model, which logs when and how bias was evaluated and what changed over time. This not only helps internally but also fulfills transparency requirements for governance.

- *Align bias evaluation with governance frameworks and stakeholder values.* Choose evaluation targets and thresholds that make sense in the context of use and according to any ethical guidelines or laws you operate under. For instance, if deploying a chatbot in the EU, ensure your bias tests cover all EU protected characteristics, since the AI Act will expect that. Involve representatives from affected communities when designing or reviewing bias tests—they might point out biases or harms you did not consider initially. Ultimately, the goal is not just to "pass benchmarks" but to ensure the model is fair in the eyes of those who use or are impacted by it.

Table 6 maps common deployment contexts to concrete method bundles. It is intended as a quick-start guide for selecting an evaluation plan aligned with resources, risks, and regulatory expectations.

Table 6. Context-aware selection of bias evaluation methods.

Context	Recommended methods
Early model triage	Embedding and sentence association tests such as WEAT and SEAT, likelihood-based tests, a small sweep of counterfactual pairs, and basic generation toxicity or regard analysis with confidence intervals.
Multilingual deployment	Localized prompt sets, per-language calibration of bias and toxicity detectors, and stratified human validation across languages, dialects, and groups.
High-stakes domain	Task-grounded vignettes with parity checks for accuracy and decision gaps, targeted stress tests, and certification-style evaluation with specified metrics and input distributions.
Governance-ready release	A multi-metric report with uncertainty estimates, a model and bias card, released code and configuration artifacts, and a documented monitoring and escalation plan.

By following this checklist, practitioners can systematically evaluate bias and work towards mitigating it. The recommendations emphasize a proactive, rigorous, and context-aware approach—evaluating from multiple angles, quantifying confidence in findings, and iterating as needed. It is worth noting that bias evaluation is an ongoing responsibility: as LLMs are updated or encounter new real-world data, new biases can emerge, and societal norms of fairness may shift. Therefore, treating bias evaluation as a continuous process is the best practice.

Final Thoughts Bias in LLMs is a complex, multifaceted problem at the intersection of technology and society. Through this review, we have assembled a broad toolkit to detect and quantify biases, from internal representations to external behaviors, and then to certify model fairness properties. We have also identified the limitations of these methods and the challenges that lie ahead. For practitioners, the path forward involves using these tools in combination, remaining vigilant about new forms of bias, and engaging with the wider community–including policymakers and affected users–to define what fairness means for each application. By doing so, we move toward LLM deployments that are not only innovative, but also equitable and worthy of the trust of the society.

References

Abid, A., Farooqi, M., & Zou, J. (2021). Large language models associate muslims with violence. *Nature Machine Intelligence*, *3*(6), 461–463. doi: https://doi.org/10.1038/s42256-021-00356-9

An, J., Huang, D., Lin, C., & Tai, M. (2025, February). Measuring gender and racial biases in large language models: Intersectional evidence from automated resume evaluation. *PNAS Nexus*, *4*(3). Retrieved from http://dx.doi.org/10.1093/pnasnexus/pgaf089 doi: https://doi.org/10.1093/pnasnexus/pgaf089

Anthis, J., Lum, K., Ekstrand, M., Feller, A., D'Amour, A., & Tan, C. (2024). The impossibility of fair LLMs. *arXiv*. Retrieved from https://arxiv.org/abs/2406.03198 doi: https://doi.org/10.18653/v1/2025.acl-long.5

Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, *104*(3), 671–732. doi: https://doi.org/10.2139/ssrn.2477899

Bartl, M., Nissim, M., & Gatt, A. (2020). Unmasking contextual stereotypes: Measuring and mitigating bert's gender bias. In *Proceedings of the second workshop on gender bias in natural language processing* (pp. 1–16). Barcelona, Spain (Online): Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.gebnlp-1.1/

Bastani, O., Zhang, X., & Solar-Lezama, A. (2019). Probabilistic verification of fairness properties via concentration. *Proceedings of the ACM on Programming Languages*, *3*(OOPSLA), 118:1–118:27. Retrieved from https://dl.acm.org/doi/10.1145/3360544 doi: https://doi.org/10.1145/3360544

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 acm conference on fairness, accountability, and transparency (facct)* (pp. 610–623). doi: https://doi.org/10.1145/3442188.3445922

Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is power: A critical survey of bias in NLP. In *Proceedings of the 58th annual meeting of the association for computational linguistics (acl)* (pp. 5454–5476). doi: https://doi.org/10.18653/v1/2020.acl-main.485

Bolukbasi, T., Chang, K., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems 29 (neurips 2016)* (pp. 4349–4357).

Bordia, S., & Bowman, S. R. (2019). Identifying and reducing gender bias in word-level language models. *arXiv*. Retrieved from https://arxiv.org/abs/1904.03035 doi: https://doi.org/10.18653/v1/n19-3002

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency* (pp. 77–91).

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*,

356(6334), 183–186. doi: https://doi.org/10.1126/science.aal4230

Cao, Y., Pruksachatkun, Y., Chang, K., Gupta, R., Kumar, V., Dhamala, J., & Galstyan, A. (2022). On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. In *Proceedings of acl 2022 (short papers)* (pp. 561–570). doi: https://doi.org/10.18653/v1/2022.acl-short.62

Chaudhary, I., Hu, Q., Kumar, M., Ziyadi, M., Gupta, R., & Singh, G. (2025). Certifying counterfactual bias in LLMs. In *International conference on learning representations (iclr).* (OpenReview)

Crenshaw, K. (1991, July). Mapping the margins: Intersectionality, identity politics, and violence against women of color. *Stanford Law Review*, *43*(6), 1241–1299. Retrieved from http://dx.doi.org/10.2307/1229039 doi: https://doi.org/10.2307/1229039

Cui, J., Chiang, W.-L., Stoica, I., & Hsieh, C.-J. (2025). OR-Bench: An over-refusal benchmark for large language models. In *Proceedings of the 42nd international conference on machine learning (icml).* (arXiv:2405.20947)

De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Kalai, A., & Crawford, K. (2019). Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the acm conference on fairness, accountability, and transparency (fat*).* doi: https://doi.org/10.1145/3287560.3287572

Dev, S., & Phillips, J. M. (2019). Attenuating bias in word vectors. In *Proceedings of the 22nd international conference on artificial intelligence and statistics (aistats)* (pp. 879–887).

Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K., & Gupta, R. (2021). BOLD: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 acm conference on fairness, accountability, and transparency (facct).* doi: https://doi.org/10.1145/3442188.3445924

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference (itcs)* (pp. 214–226). (Preprint 2011) doi: https://doi.org/10.1145/2090236.2090255

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 55–65). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1006/ doi: https://doi.org/10.18653/v1/D19-1006

European Parliament, & Council of the European Union. (2024). Regulation (EU) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (EC) no 300/2008, (EU) 2017/745, (EU) 2017/746, (EU) 2019/881 and (EU) 2022/2065 and directive 2009/125/ec (Artificial Intel-

ligence Act). *Official Journal of the European Union*, *L 2024/1689*. Retrieved from https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng

Ferrara, E. (2023). Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies. *arXiv*. Retrieved from https://arxiv.org/abs/2304.07683 doi: https://doi.org/10.3390/sci6010003

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M. M., Kim, S., Dernoncourt, F., ... Ahmed, N. K. (2024). Bias and fairness in large language models: A survey. *Computational Linguistics*, *50*(3), 1097–1158. doi: https://doi.org/10.1162/coli_a_00524

Gehman, S., Gururangan, S., Sap, M., Choi, Y., & Smith, N. A. (2020). Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 3356–3369). doi: https://doi.org/10.18653/v1/2020.findings-emnlp.301

Gumilar, K. E., Indraprasta, B. R., Hsu, Y.-C., Yu, Z.-Y., Chen, H., Irawan, B., ... Tan, M. (2024, July). Disparities in medical recommendations from AI-based chatbots across different countries/regions. *Scientific Reports*, *14*(1). Retrieved from http://dx.doi.org/10.1038/s41598-024-67689-0 doi: https://doi.org/10.1038/s41598-024-67689-0

Guo, Y., Guo, M., Su, J., Yang, Z., Zhu, M., Li, H., & Qiu, M. (2024). Bias in large language models: Origin, evaluation, and mitigation. *arXiv*. Retrieved from https://arxiv.org/abs/2411.10915

Hanu, L., & Unitary team. (2020). *Detoxify*. https://github.com/unitaryai/detoxify. Retrieved from https://github.com/unitaryai/detoxify (GitHub repository)

Hofmann, V., Kalluri, P. R., Jurafsky, D., & King, S. (2024). Dialect prejudice predicts AI decisions about people's character, employability, and criminality. In *Proceedings of the 2024 acm conference on fairness, accountability, and transparency (facct)* (pp. 1321–1340).

Huang, P., Zhang, H., Jiang, R., Stanforth, R., Welbl, J., Rae, J., ... Kohli, P. (2019). Reducing sentiment bias in language models via counterfactual evaluation. *arXiv*. Retrieved from https://arxiv.org/abs/1911.03064 doi: https://doi.org/10.18653/v1/2020.findings-emnlp.7

Kearns, M., Neel, S., Roth, A., & Wu, Z. S. (2018, Jul). Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 2564–2572). PMLR. Retrieved from https://proceedings.mlr.press/v80/kearns18a.html

Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). *Inherent trade-offs in the fair determination of risk scores*. Retrieved from https://arxiv.org/abs/1609.05807

Kotek, H., Dockum, R., & Sun, D. (2023). Gender bias and stereotypes in large language models. In *Proceedings of the acm collective intelligence conference (ci 2023)*. doi: https://doi.org/10.1145/3582269.3615599

Krishna, S., Gupta, R., Verma, A., Dhamala, J., Pruksachatkun, Y., & Chang,

K.-W. (2022, may). Measuring fairness of text classifiers via prediction sensitivity. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 5830–5842). Dublin, Ireland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.acl-long.401/ doi: https://doi.org/10.18653/v1/2022.acl-long.401

Kurita, K., Vyas, N., Pareek, A., Black, A. W., & Tsvetkov, Y. (2019). Measuring bias in contextualized word representations. In *Proceedings of the first acl workshop on gender bias for nlp* (pp. 166–172). doi: https://doi.org/10.18653/v1/w19-3823

Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. In *Advances in neural information processing systems 30 (neurips 2017)* (pp. 4066–4076).

Lee, G., Hartmann, V., Park, J., Papailiopoulos, D., & Lee, K. (2023). Prompted llms as chatbot modules for long open-domain conversation. In *Findings of the association for computational linguistics: Acl 2023* (pp. 4536–4554). Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.findings-acl.277/ doi: https://doi.org/10.18653/v1/2023.findings-acl.277

Li, Y., Du, M., Song, R., Wang, X., & Wang, Y. (2024). A survey on fairness in large language models. *arXiv*. Retrieved from https://arxiv.org/abs/2308.10149 (Version 2 (revised 2024-02-21)) doi: https://doi.org/10.48550/arXiv.2308.10149

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., ... Koreeda, Y. (2023). Holistic evaluation of language models. *Transactions on Machine Learning Research*. Retrieved from https://arxiv.org/abs/2211.09110 (TMLR; arXiv:2211.09110) doi: https://doi.org/10.48550/arXiv.2211.09110

Liu, T., Luo, R., Chen, Q., Qin, Z., Sun, R., Yu, Y., & Zhang, C. (2024). Jailbreaking black-box large language models in twenty queries. In *33rd usenix security symposium (usenix security 24)*. Philadelphia, PA: USENIX Association. Retrieved from https://www.usenix.org/conference/usenixsecurity24/presentation/liu-tong (See also arXiv:2310.08419)

Liu, Y., Yang, T., Huang, S., Zhang, Z., Huang, H., Wei, F., ... Zhang, Q. (2023). *Calibrating llm-based evaluator*. Retrieved from https://arxiv.org/abs/2309.13308

May, C., Wang, A., Bordia, S., Bowman, S. R., & Rudinger, R. (2019). On measuring social biases in sentence encoders. In *Proceedings of naacl-hlt 2019* (pp. 622–628). doi: https://doi.org/10.18653/v1/n19-1063

Meade, N., Poole-Dayan, E., & Reddy, S. (2022). An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1878–1898). Dublin, Ire-

land: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.acl-long.132/ (Published at ACL 2022; widely cited in 2023 literature) doi: https://doi.org/10.18653/v1/2022.acl-long.132

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*(6), 1–35. doi: https://doi.org/10.1145/3457607

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the 2019 conference on fairness, accountability, and transparency (fat*)* (pp. 220–229). doi: https://doi.org/10.1145/3287560.3287596

Nadeem, M., Bethke, A., & Reddy, S. (2021). Stereoset: Measuring stereotypical bias in pretrained language models. In *Proceedings of acl 2021 (long papers)* (pp. 5356–5371). doi: https://doi.org/10.18653/v1/2021.acl-long.416

Nangia, N., Vania, C., Bhalerao, R., & Bowman, S. R. (2020). Crows-pairs: A challenge dataset for measuring social biases in masked language models. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 227–239). doi: https://doi.org/10.18653/v1/2020.emnlp-main.154

Panickssery, A., Bowman, S. R., & Feng, S. (2024). *Llm evaluators recognize and favor their own generations.* Retrieved from https://arxiv.org/abs/2404.13076 doi: https://doi.org/10.52202/079017-2197

Parrish, A., Chen, A., Nangia, N., Padmakumar, V., Phang, J., Thompson, J., ... Bowman, S. (2022). BBQ: A hand-built bias benchmark for question answering. In *Findings of the association for computational linguistics: Acl 2022* (pp. 2086–2105). doi: https://doi.org/10.18653/v1/2022.findings-acl.165

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., ... Irving, G. (2022). *Red teaming language models with language models.* Retrieved from https://arxiv.org/abs/2202.03286 doi: https://doi.org/10.18653/v1/2022.emnlp-main.225

Raji, I. D., Denton, E., Bender, E. M., Hanna, A., & Paullada, A. (2021). AI and the everything in the whole wide world benchmark: A critical analysis of the biggest benchmarks in AI. *arXiv*. Retrieved from https://arxiv.org/abs/2111.15366 (Metric validity discussion)

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M., & Goldberg, Y. (2020). Null it out: Debiasing text representations by iterative nullspace projection. In *Proceedings of the 58th annual meeting of the association for computational linguistics (acl)* (pp. 7237–7256).

Rozado, D. (2025). *Gender and positional biases in llm-based hiring decisions: Evidence from comparative cv/résumé evaluations.* Retrieved from https://arxiv.org/abs/2505.17049 doi: https://doi.org/10.7717/peerj-cs.3628

Rudinger, R., Naradowsky, J., Leonard, B., & Van Durme, B. (2018). Gender bias in coreference resolution. In *Proceedings of the 2018 conference of the north american chapter of the association for computational*

linguistics: Human language technologies, volume 2 (short papers) (pp. 8–14). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N18-2002/ doi: https://doi.org/10.18653/v1/N18-2002

Rupprecht, J., Ahnert, G., & Strohmaier, M. (2025). *Prompt perturbations reveal human-like biases in large language model survey responses.* Retrieved from https://arxiv.org/abs/2507.07188

Sheng, E., Chang, K., Natarajan, P., & Peng, N. (2019). The woman worked as a babysitter: On biases in language generation. In *Proceedings of emnlp-ijcnlp 2019* (pp. 3407–3412). doi: https://doi.org/10.18653/v1/d19-1339

Sherry, J. H. (1965). The civil rights act of 1964: Fair employment practices under title VII. *Cornell Hotel and Restaurant Administration Quarterly*, *6*(2), 3–6. doi: https://doi.org/10.1177/001088046500600202

Sim, J., & Reid, N. (1999). Statistical inference by confidence intervals: Issues of interpretation and utilization. *Physical Therapy*, *79*(2), 186–195. doi: https://doi.org/10.1093/ptj/79.2.186

Smith, E. M., Hall, M., Kambadur, M., Presani, E., & Williams, A. (2022). I'm sorry to hear that: Finding new biases in language models with a holistic descriptor dataset. *arXiv*. Retrieved from https://arxiv.org/abs/2201.11745 doi: https://doi.org/10.18653/v1/2022.emnlp-main.625

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., ... Wang, J. (2019). Release strategies and the social impacts of language models. *arXiv*. Retrieved from https://arxiv.org/abs/1908.09203

Suresh, H., & Guttag, J. V. (2021). A framework for understanding sources of harm throughout the machine learning life cycle. In *Proceedings of acm eaamo 2021.* (Article 7) doi: https://doi.org/10.1145/3465416.3483305

Tabassi, E. (2023). *Artificial intelligence risk management framework (ai rmf 1.0).* Retrieved from http://dx.doi.org/10.6028/NIST.AI.100-1 doi: https://doi.org/10.6028/nist.ai.100-1

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., & Shieber, S. (2020). Investigating gender bias in language models using causal mediation analysis. In *Advances in neural information processing systems 33 (neurips 2020).*

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., ... Li, B. (2024). Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *arXiv*. Retrieved from https://arxiv.org/abs/2306.11698

Zhang, Y., Huang, Y., Sun, Y., Liu, C., Zhao, Z., Fang, Z., ... Zhu, J. (2024). Multitrust: A comprehensive benchmark towards trustworthy multimodal large language models. *arXiv*. Retrieved from https://arxiv.org/abs/2406.07057 doi: https://doi.org/10.52202/079017-1561

Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., & Chang, K. (2019). Gender bias in contextualized word embeddings. *arXiv*. Retrieved from https://arxiv.org/abs/1904.03310 doi: https://doi.org/10.18653/v1/n19-1064

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. (2017a). Men

also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 conference on empirical methods in natural language processing (emnlp)*.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of naacl-hlt 2018* (pp. 15–20). doi: https://doi.org/10.18653/v1/n18-2003

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2017b). Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2979–2989). Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.18653/v1/D17-1323 doi: https://doi.org/10.18653/v1/d17-1323

Zollo, T. P., Morrill, T., Deng, Z., Snell, J. C., Pitassi, T., & Zemel, R. (2024). Prompt risk control: A rigorous framework for responsible deployment of large language models. *arXiv*. Retrieved from https://arxiv.org/abs/2311.13628

Journal of Behavioral Data Science, 2026, *6* (1), 136-151.
DOI:https://doi.org/10.35566/jbds/thakur

Computational Approaches to Diabetes Risk Assessment: A Review of Data-Driven Techniques

Agrimaa Singh Thakur[1⋆] and Amit Verma[2]

[1] Research Scholar, Department of Computer Science and Engineering, Maharaja Agrasen University, Himachal Pradesh, India
agrimaa26@gmail.com
[2] Associate Professor, Department of Computer Science and Engineering, Maharaja Agrasen University, Himachal Pradesh, India
verma0152@gmail.com

Abstract. Over 540 million people worldwide suffer from diabetes mellitus, making it a serious global health concern. The advancement of robust predictive models that surpass traditional risk assessment approaches has demonstrated significant potential due to machine learning techniques. This thorough analysis summarizes the state of the art in machine learning-based diabetes prediction systems by examining algorithmic approaches, dataset properties, and performance indicators. The analysis shows how advanced ensemble and deep learning techniques have replaced more conventional statistical methods in order to achieve better results. Critical drawbacks still exist, nonetheless, such as an excessive dependence on datasets with a restricted demographic, a lack of real-world validation, and inadequate model interpretability for clinical acceptability. Regulatory obstacles, population-specific dataset variability, and discrepancies between algorithmic performance and therapeutic impact are some of the main obstacles. In order to convert advancements into clinically useful systems, future priorities include creating representative datasets, putting explainable artificial intelligence (AI) into practice, and carrying out prospective clinical studies.

Keywords: Diabetes Prediction · Machine Learning · Healthcare Analytics · Predictive Modeling · Clinical Decision Support

1 Introduction

Diabetes mellitus is a chronic metabolic condition marked by persistent hyperglycemia brought on by insufficient insulin secretion, defective insulin action,

⋆ Corresponding author: agrimaa26@gmail.com

or a combination of the two. It has developed into one of the 21st century's most urgent worldwide health issues. The International Diabetes Federation's latest projections paint an alarming picture: from 540 million affected individuals in 2023, the global burden is expected to escalate to 783 million by 2045, with potential costs exceeding $1 trillion annually (International Diabetes Federation, 2023, 2025c). This exponential growth trajectory positions diabetes as not merely a medical condition but a socioeconomic crisis demanding immediate and innovative interventions.

The pathophysiological complexity of diabetes encompasses multiple types, each presenting distinct challenges. Type 1 diabetes, an autoimmune condition primarily affecting younger populations, results from pancreatic beta-cell destruction and necessitates ongoing insulin treatment. (American Diabetes Association, n.d.-b; Atkinson, Eisenbarth, & Michels, 2014; Knip & Simell, 2012). Type 2 diabetes, which makes up 90–95% of all cases, is primarily linked to obesity, a sedentary lifestyle, and genetic predisposition. It is caused by growing insulin resistance and relative insulin shortage (American Diabetes Association, n.d.-c; Defronzo, Ferrannini, Zimmet, & Alberti, 2015). Gestational diabetes mellitus affects pregnant women previously undiagnosed with diabetes, presenting risks to both maternal and fetal health while predicting future Type 2 diabetes development (American Diabetes Association, n.d.-a).

The clinical manifestations of diabetes extend far beyond elevated blood glucose levels. The disease precipitates a cascade of complications that significantly impact quality of life and mortality rates. The primary cause of death for diabetics is still cardiovascular disease, and their risk of heart attacks and strokes is significantly higher than that of non-diabetics. When diabetic nephropathy develops into end-stage renal disease, dialysis or kidney transplants are frequently required. Diabetic neuropathy, or damage to the nerves, raises the risk of foot ulcers, which in extreme circumstances can lead to amputation. Similarly, the main cause of vision impairment and blindness in adults is still diabetic eye problems (retinopathy). These complications not only devastate individual lives but impose enormous economic burdens on healthcare systems worldwide.

India's rise to prominence as the "diabetes capital of the world" is a prime example of the scope of the worldwide problem. Nearly 17% of the world's diabetes cases are in India, where there are over 90 million cases among people aged 20 to 79 (International Diabetes Federation, 2025a, 2025b). According to the World Health Organization, India will account for 58% of the rise in Type 2 diabetes incidence worldwide and will see a 90% increase in diabetes-related mortality by 2030 compared to 2017 levels (World Health Organization, 2016). These figures demonstrate the critical need for economically feasible, culturally appropriate, and population-specific prediction systems.

The combination of artificial intelligence and machine learning into diabetes prediction represents a paradigm shift toward personalized medicine and precision healthcare. By analyzing diverse datasets encompassing genetic profiles, lifestyle factors, medical histories, and real-time physiological data, these technologies enable the development of individualized treatment plans and risk as-

sessments (Rajkomar, Dean, & Kohane, 2019). The potential extends beyond prediction to encompass continuous monitoring, treatment optimization, and complication prevention, fundamentally transforming diabetes care from reactive to proactive. However, the translation of machine learning advances into clinical practice faces significant challenges. Issues of model interpretability, regulatory approval, healthcare professional acceptance, and integration with existing clinical workflows present substantial barriers. Additionally, concerns regarding data privacy, algorithmic bias, and generalizability across diverse populations require careful consideration. The over-reliance on limited datasets, particularly the PIMA Indian Diabetes Dataset, raises questions about model applicability across different ethnic groups, geographic regions, and healthcare systems.

2 Related Work

The use of machine learning techniques to predict diabetes has advanced significantly in recent years. From utilizing simple algorithm comparisons, researchers have progressed to applying sophisticated ensemble models and deep learning methods. This section provides a thorough analysis of important studies across time, with a focus on new research trends, performance improvements, and methodological advancements.

2.1 Foundational Studies

The early period of ML-based diabetes prediction was characterized by establishing baseline performance metrics and exploring the potential of traditional machine learning algorithms. Uloko et al. (2018) conducted one of the first comprehensive meta-analyses focusing on diabetes risk factors in Nigeria, examining 23 independent studies comprising 14,650 participants. Their work identified urbanization, lack of physical activity, aging, and poor dietary habits as primary risk factors, establishing the epidemiological foundation for subsequent predictive modeling efforts. Joshi, Pramila, and Chawan (2018) represented pioneering efforts in algorithmic comparison, incorporating Logistic Regression and Support Vector Machines for early diabetes prediction. Their study achieved 79% accuracy with SVM, demonstrating the potential of machine learning approaches while highlighting the need for performance improvements. This work established the benchmark for subsequent comparative studies and emphasized the importance of feature selection in predictive accuracy. Sneha and Gangil (2019) contributed to the field by thoroughly evaluating different algorithms on the PIMA Indian Diabetes Dataset using the WEKA software platform. Their comparison of Naive Bayes, Random Forest, and Decision Trees yielded significant insights: Random Forest achieved 98.20% specificity, Decision Tree reached 98.00% specificity, while Naive Bayes demonstrated 82.30% overall accuracy. The study's emphasis on feature selection techniques for early identification established important methodological precedents for optimal classification performance enhancement. Sonar and JayaMalini (2019) expanded the algorithmic

landscape by incorporating Artificial Neural Networks (ANN) alongside traditional approaches including Gaussian Naive Bayes, SVM, and Decision Trees. Their work demonstrated ANN's superiority over conventional algorithms when applied to the PIMA dataset, achieving improved precision, recall, accuracy, and F1-score metrics. This study marked the beginning of neural network applications in diabetes prediction, setting the stage for subsequent deep learning developments.

2.2 Methodological Innovations

The intermediate period witnessed significant methodological sophistication, with researchers focusing on ensemble methods, advanced preprocessing techniques, and novel feature engineering approaches. Mujumdar and Vaidehi (2019) achieved a breakthrough with their integrated approach combining genetic susceptibility, lifestyle habits, and clinical measures. Their AdaBoost pipeline achieved exceptional 98.8% accuracy, while the Logistic Regression model maintained 96% classification accuracy, demonstrating the power of ensemble techniques in diabetes prediction. Saru and Subashree (2019) contributed to the methodological foundation through medical bioinformatics analysis, comparing Naive Bayes, Decision Trees, and K-Nearest Neighbor algorithms using UCI repository data. Their work established KNN's superior accuracy performance while emphasizing the critical role of classifier selection for precise and timely diagnosis. The study highlighted machine learning's potential for enhancing diabetes prediction algorithms beyond traditional statistical approaches. Kopitar, Kocbek, Cilar Budler, Sheikh, and Stiglic (2020) conducted one of the most comprehensive algorithmic comparisons of the period, evaluating LightGBM, Glmnet, XGBoost, and Random Forest against traditional regression analysis. Using 100 bootstrap resamples to simulate recurring information arrival, their study revealed algorithm ranking: XGBoost > Random Forest > LightGBM > Glmnet > Simple Regression. Significantly, LightGBM demonstrated remarkable stability in variable selection over time, establishing its value for longitudinal predictive modeling. Syed and Khan (2020) advanced the clinical applicability through their cross-sectional survey approach, employing binary logistic regression and Chi-Squared tests for Type 2 diabetes prediction. Their Decision Forest model achieved superior performance with mean F1 score of 0.8453 ± 0.0268, validated across NHANES and PIDD datasets. The deployment of their calibrated model as an API web service represented significant progress toward clinical translation.

2.3 Advanced Ensemble and Deep Learning Approaches

In the past few years, diabetes prediction models have become more refined, as researchers have combined advanced computational techniques to enhance accuracy and ensure stronger clinical applicability. Zhang, Wang, Niu, et al. (2020) utilized data from the Henan rural cohort study to examine machine learning performance in rural Chinese populations. Their evaluation of six algorithms (SVM, Random Forest, ANN, Gradient Boosting Machine, Classification and

Regression Trees, and Logistic Regression) achieved moderate predictive performance with AUC values ranging 0.767-0.872, with GBM achieving the highest AUC. Importantly, their work identified novel risk factors including sweet taste preference and urinary symptoms that traditional models overlooked. Ahmed et al. (2022) demonstrated the power of comprehensive methodological integration, applying ensemble methods, deep learning, and feature engineering to health parameters and medical records. Their approach achieved remarkable 94.87% accuracy, representing substantial improvement over traditional methods through systematic integration of advanced techniques. Zhou, Xin, and Li (2023) achieved exceptional performance through their innovative combination of Boruta feature selection and ensemble learning techniques. Their systematic approach utilizing PIMA dataset with Boruta's statistical significance-based feature selection and K-Means++ clustering achieved 98% accuracy. The integration of stacking ensemble methods for classification demonstrated superior performance compared to related methods, indicating significant potential for practical diabetes prevention and management. Doğru, Buyrukoglu, and Arı (2023) introduced hybrid super ensemble learning, combining meta-learning models with SVM across multiple datasets. Their approach achieved outstanding accuracy rates: 99.6% for early-stage diabetes prediction, 92% for PIMA dataset, and 98% for hospital datasets. The Chi-square test emerged as the optimal feature selection method, with GridSearch optimization of hyperparameters contributing to exceptional performance across diverse datasets.

2.4 Clinical Integration and Real-World Applications

Recently, studies have shifted their attention toward bridging the gap between research and practice, emphasizing the implementation of predictive models in real clinical settings and real-world healthcare environments. Su, Huang, Zhu, Lyu, and Ji (2023) employed federated learning techniques to overcome important privacy concerns, enabling multi-institutional collaboration without compromising patient data privacy. Their secure protocols for regression and tree-based models demonstrated effectiveness across XGBoost, LightGBM, Neural Networks, and Logistic Regression, validated on both PIMA and local datasets. Hennebelle, Materwala, and Ismail (2023) introduced HealthEdge, representing comprehensive IoT edge and cloud computing-based predictive modeling. Using data from Sylhet Diabetes Hospital in Bangladesh and PIDD, their Random Forest algorithm achieved 97% accuracy with 6% average predictive improvement, demonstrating the promising future of integrated predictive healthcare frameworks. Qi, Song, Liu, Zhang, and Wong (2023) presented the sophisticated KFPredict Ensemble Model, incorporating Recursive Feature Elimination and correlation coefficient analysis with multi-input neural networks. Their final stacking approach combining KF_NN with SVM, Random Forest, and KNN achieved 93.5% accuracy, 85% sensitivity, and 98% specificity, representing up to 18.18% and 14.93% improvement over single prediction methods and previous models respectively.

2.5 Emerging Trends and Specialized Applications

Current research is increasingly directed toward specialized domains and the integration of advanced technologies to improve diabetes prediction. These efforts emphasize addressing specific clinical challenges, enhancing diagnostic accuracy, and promoting more personalized and effective healthcare solutions. Aslan and Sabanci (2023) proposed novel deep learning approaches by converting numerical attributes in PIMA dataset to image data, enabling CNN models like ResNet18 and ResNet50 for diabetes prediction. Their investigation of fusion strategies combining deep features with SVM classification demonstrated the efficiency of image-based representations for early detection enhancement. Butunoi, Stolojescu-Crisan, and Negru (2024) developed sophisticated algorithms for Type 1 diabetes management, focusing on macrovascular complications and severe hyperglycemia/hypoglycemia episode prediction. Their analysis of GRU, LSTM, RNN architectures, and regression models using Dexcom G6 continuous glucose monitoring data highlighted the importance of accurate forecast models for daily diabetes management and long-term outcome improvement. Kokkorakis et al. (2023) addressed critical generalizability challenges through predictive model development across various ethnicities using UK Biobank data. The researchers developed logistic regression classifiers using training data exclusively from White participants, subsequently validating model performance across five additional ethnic populations and the Lifelines cohort. The models demonstrated robust discriminative capacity, yielding area under the receiver operating characteristic curve (AUROC) values of 0.901 for cross-sectional prevalence prediction and 0.873 for prospective eight-year incidence forecasting. These metrics indicate strong generalizability of the predictive framework across ethnically heterogeneous populations.

2.6 Comparative Analysis

A thorough comparison of the examined methods spanning datasets, algorithms, performance metrics, significant advancements, and constraints is given in Table 1. The development of diabetes prediction research in recent years is shown by this thorough analysis.

With accuracy rising from 79% in early research to 99.6% in more current hybrid ensemble techniques, the comparison shows notable performance gains. In terms of methodology, the discipline has advanced from basic classifiers to deep learning, federated approaches, and complex ensemble techniques. Even while the PIMA dataset is still the most popular, contemporary research is using multi-ethnic and population-specific datasets more and more to improve generalizability. Notwithstanding these developments, significant obstacles still exist, including deployment complexity, model interpretability issues, inadequate clinical validation, and demographic restrictions in training data. To convert computational advancements into therapeutic impact, these gaps must yet be filled.

Table 1. Comprehensive Comparison of Diabetes Prediction Studies

Authors & Year	Dataset(s)	Algorithms	Best Performance	Key Innovation / Contribution
Joshi et al. (2018)	PIMA	Logistic Regression, SVM	79% accuracy (SVM)	Early algorithmic comparison
Sneha and Gangil (2019)	PIMA (WEKA)	Naive Bayes, Random Forest, Decision Tree	RF: 98.20% Specificity; NB: 82.30% Accuracy	Feature selection for early identification
Sonar and JayaMalini (2019)	PIMA	ANN, Gaussian NB, SVM, Decision Trees	ANN superior (Precision, Recall, F1)	First neural network application
Mujumdar and Vaidehi (2019)	PIMA	AdaBoost, Logistic Regression	AdaBoost: 98.8%; LR: 96%	Integrated genetic, lifestyle, clinical measures
Saru and Subashree (2019)	UCI Repository	Naive Bayes, Decision Trees, KNN	KNN superior	Medical bioinformatics approach
Kopitar et al. (2020)	Clinical (100 bootstrap resamples)	LightGBM, Glmnet, XGBoost, RF	XGBoost > RF > LightGBM > Glmnet	LightGBM stability in variable selection
Syed and Khan (2020)	NHANES, PIDD	Decision Forest, Binary LR, Chi-Square	F1: 0.8453 ± 0.0268	Deployed as API web service
Zhang et al. (2020)	Henan Rural Cohort (China)	SVM, RF, ANN, GBM, CART, LR	GBM: AUC 0.872	Novel risk factors (sweet taste, urinary symptoms)
Alanazi and Mezher (2020)	Saudi Arabia Healthcare	Random Forest	AUROC 0.99	Population-specific risk factors
Ahmed et al. (2022)	Health parameters and records	Ensemble, Deep Learning, Feature Eng.	94.87% accuracy	Comprehensive methodological integration
Bhat, Selvam, Ansari, Ansari, and Rahman (2022)	North Kashmir (>1,000 records)	Random Forest	98% Accuracy	Local demographic characteristics
Zhou et al. (2023)	PIMA	Boruta, K-Means++, Stacking	98% Accuracy	Statistical significance-based features
Doğru et al. (2023)	Multiple datasets	Hybrid super ensemble, SVM, GridSearch	99.6% (early-stage); 92% (PIMA)	Hybrid ensemble across diverse datasets
Su et al. (2023)	PIMA, Local dataset	Federated Learning: XGBoost, LightGBM, NN, LR	Effective across algorithms	Privacy-preserving multi-institutional
Hennebelle et al. (2023)	Sylhet Hospital (Bangladesh), PIDD	RF (IoT edge + cloud)	97% Accuracy (6% improvement)	HealthEdge: IoT-edge-cloud framework
Qi et al. (2023)	PIMA	KFPredict: RFE, multi-input NN, stacking	93.5% Accuracy; 85% Sensitivity; 98% Specificity	18.18% improvement over single methods
Aslan and Sabanci (2023)	PIMA (as images)	CNN (ResNet18/50), Deep features + SVM	Efficient early detection	Numerical-to-image conversion for CNN
Kokkorakis et al. (2023)	UK Biobank (631,748 prev; 67,083 inc)	Logistic Regression (cross-ethnic)	AUC 0.901 (prev); 0.873 (inc)	Multi-ethnic validation (5 groups)
Butunoi et al. (2024)	Dexcom G6 CGM (Type 1)	GRU, LSTM, RNN, Regression	Varies by episode	Type 1 focus; complications prediction

3 Datasets in Diabetes Prediction: A Comprehensive Analysis

The foundation of any successful machine learning application lies in the quality, diversity, and representativeness of the underlying datasets. In diabetes prediction research, dataset characteristics significantly influence model performance, generalizability, and clinical applicability. This section provides an exhaustive analysis of datasets commonly employed in diabetes prediction studies, examining their strengths, limitations, and impact on model development.

3.1 Primary Datasets in Diabetes Prediction Research

PIMA Indian Diabetes Dataset (PIDD) The PIMA Indian Diabetes Dataset stands as the most frequently utilized resource in diabetes prediction research, appearing in over 60% of published studies (UCI Machine Learning Repository, n.d.). These data were originally collected by the National Institute of Diabetes and Digestive and Kidney Diseases and include the medical records of 768 female patients of Pima Indian ethnicity who were 21 years of age or older. The dataset comprises eight numerical attributes across 768 instances with binary classification outcomes for diabetic versus non-diabetic status. The class distribution shows approximately 35% positive cases representing diabetic patients, while the population consists exclusively of Pima Indian women ranging in age from 21 to 81 years. The feature characteristics include pregnancies ranging from zero to seventeen occurrences, plasma glucose concentration measured at two hours during oral glucose tolerance testing with values spanning zero to 199 mg/dL, diastolic blood pressure measurements in mmHg ranging from zero to 122, and triceps skinfold thickness measurements in millimeters with values from zero to 99. The dataset provides several advantages including its establishment as a well-recognized benchmark enables direct comparison across studies, clean and preprocessed data with minimal missing values, balanced feature representation covering key diabetes risk factors, extensive validation across multiple machine learning algorithms, and open-source availability facilitating reproducible research. However, significant limitations affect the dataset's applicability. The demographic diversity remains severely restricted through single ethnic group representation and gender-specific sampling, while the small sample size may limit model robustness and generalizability. Potential genetic homogeneity reduces applicability across broader populations, temporal constraints from data collection within specific time periods affect relevance, and geographic specificity limits global applicability of developed models.

Framingham Heart Study Dataset The Framingham Heart Study represents one of the most comprehensive longitudinal cardiovascular research initiatives, initiated in 1948 and continuing today. For diabetes prediction applica-

tions, researchers utilize subsets of this extensive database containing approximately 4,240 records with multiple clinical, demographic, and lifestyle variables collected through prospective longitudinal cohort design (Framingham Heart Study, n.d.).

The population consists predominantly of Caucasian residents of Framingham, Massachusetts, with data collection spanning multiple decades enabling temporal relationship analysis. Demographic characteristics like age and gender, cardiovascular risk factors like blood pressure and cholesterol, lifestyle factors like smoking and physical activity, anthropometric measurements like body mass index (BMI) and waist circumference, biochemical parameters like glucose and lipid profiles, and comprehensive family history data are all important factors in predicting diabetes.

Limitations include predominantly Caucasian population composition limiting ethnic diversity and generalizability, geographic specificity from single US location restricting broader applicability, potential cohort effects from long-term study design affecting temporal validity, complex data structure requiring sophisticated preprocessing techniques, and limited representation of contemporary lifestyle factors affecting current relevance.

CDC BRFSS Diabetes Health Indicators Dataset The Behavioral Risk Factor Surveillance System represents the world's largest continuously conducted health survey system, providing population-level diabetes risk factor data through annual cross-sectional telephone surveys. The dataset contains over 253,680 records with 21 or more health behavior and demographic variables, covering all 50 US states, District of Columbia, and territories with representative adult US population sampling (Centers for Disease Control and Prevention, n.d.-a).

Key features encompass general health status indicators, BMI and physical activity measures, healthcare access and utilization patterns, demographics including age, education, and income levels, behavioral risk factors such as smoking and alcohol consumption, chronic disease indicators, and preventive health behaviors. The massive sample size enables population-level analysis and subgroup investigations, while geographic diversity across US states provides regional variation analysis capabilities. Standardized data collection protocols ensure consistency and comparability, contemporary data reflects current health trends and behaviors, and complex survey design accounts for population representation through appropriate weighting procedures. However, limitations include self-reported data with potential recall and social desirability bias, limited clinical laboratory values reducing diagnostic precision, US- specific population characteristics limiting global generalizability, complex survey weights requiring specialized statistical analysis techniques, and potential temporal inconsistencies across different survey years.

NHANES Dataset (National Health and Nutrition Examination Survey) NHANES provides comprehensive health and nutritional status information for the US population through cross-sectional surveys with continuous data

collection. The dataset varies by cycle with approximately 5,000 participants per year, incorporating interview, examination, and laboratory components representing the US civilian population (Centers for Disease Control and Prevention, n.d.-b).

Key features for diabetes prediction include laboratory glucose and insulin measurements, HbA1c levels and other biomarkers, anthropometric measurements such as BMI and waist circumference, blood pressure and cardiovascular indicators, dietary assessment data, socioeconomic and demographic variables, and physical activity and lifestyle factors. The comprehensive clinical and laboratory data provides high-quality biomarker measurements, while standardized examination protocols ensure data consistency and reliability. Representative population sampling enables generalization to broader US population, and continuous data collection enables trend analysis over time. However, limitations include complex survey design requiring weighted analysis techniques, limited sample size compared to BRFSS reducing power for subgroup analyses, US-specific population characteristics limiting international applicability, costly data collection procedures limiting global replication, and potential selection bias in examination participation affecting representativeness. Figure 1 reveals a significant imbalance in dataset usage across diabetes prediction research. The analysis shows that studies overwhelmingly favor the PIMA Indian Diabetes Dataset, despite the availability of larger datasets with more diverse ethnic representation and longitudinal follow-up data. This narrow focus on a single dataset raises important questions about whether predictive models can reliably generalize to broader populations and clinical settings.

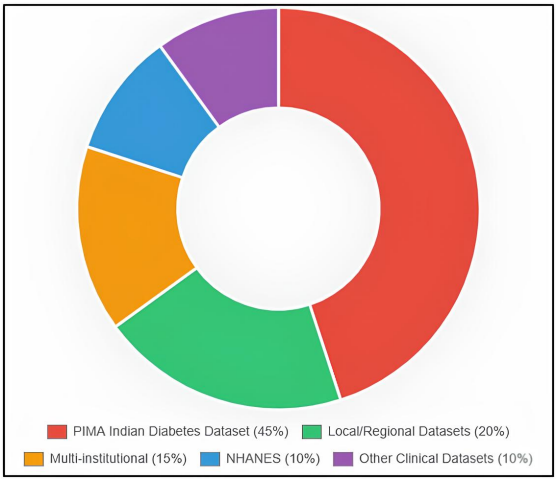


Figure 1. Dataset distribution in diabetes prediction studies.

3.2 Emerging and Specialized Datasets

Local clinical datasets have gained increasing prominence in recent studies to improve regional applicability and clinical relevance. The North Kashmir District Dataset utilized by Bhat et al. (2022) contains over 1,000 patient records from Bandipora district, incorporating local demographic and clinical characteristics that achieved 98% accuracy with Random Forest algorithms, demonstrating community-level prediction effectiveness and regional specificity advantages. The Saudi Arabia Healthcare Dataset employed by Alanazi and Mezher (2020) draws from Security Force Primary Health Care Centre data, incorporating population-specific risk factors, regional genetic and lifestyle considerations, and achieved superior performance with Random Forest algorithms showing AUROC values of 0.99. Multicultural and international datasets address generalizability challenges across diverse populations. The UK Biobank Diabetes Data utilized by Kokkorakis et al. (2023) encompasses 631,748 participants for prevalence prediction and 67,083 for incidence prediction, providing multiple ethnic group representation through questionnaire-based feature collection and cross-ethnic validation capabilities. The Chinese Rural Cohort employed by Zhang et al. (2020) draws from the Henan Rural Cohort Study with 252,176 follow-up records, representing rural Chinese population characteristics while enabling novel risk factor identification and cultural and dietary factor integration in predictive modeling.

3.3 Data Quality and Preprocessing Challenges

Diabetes prediction datasets consistently exhibit missing data patterns requiring sophisticated handling strategies. The PIMA Dataset demonstrates characteristic missing value patterns with glucose showing 0.6% true zeros that are physiologically implausible, blood pressure exhibiting 4.6% zero values, skin thickness displaying 29.6% zero values, insulin measurements containing 48.7% zero values, and BMI showing 1.4% zero values.

Common imputation strategies include mean or median imputation providing simple but potentially distribution-distorting solutions, K-Nearest Neighbors imputation preserving local data structure, multiple imputations accounting for uncertainty in missing values, and domain-specific rules applying clinical knowledge-based value assignment. Each approach presents trade-offs between computational complexity, statistical validity, and clinical interpretability.

Class imbalance issues affect most diabetes datasets with diabetic cases typically representing 20-35% of samples, requiring specialized handling techniques. Synthetic Minority Oversampling Technique generates synthetic examples of minority class instances, Adaptive Synthetic Sampling focuses on difficult-to-learn minority examples, random under sampling reduces majority class representation, cost- sensitive learning approaches adjust misclassification penalties, and ensemble methods with balanced sampling combine multiple models with different class distributions.

Feature standardization and normalization require various preprocessing approaches depending on dataset characteristics and algorithmic requirements. Numerical feature scaling includes Min-Max normalization scaling features to zero-one range, Z-score standardization creating zero mean and unit variance distributions, robust scaling using median and interquartile range to handle outliers, and quantile transformation mapping to uniform or normal distributions.

Categorical variable encoding encompasses multiple strategies including one-hot encoding, which transforms nominal variables into binary indicator variables; label encoding, which assigns numerical values to ordinal variables while preserving ranking structure; target encoding, which replaces high-cardinality categories with corresponding target variable statistics; and embedding techniques for deep learning applications, which create dense vector representations capturing complex categorical relationships while reducing dimensionality.

3.4 Dataset Limitations and Generalizability Challenges

Extensive missing values (especially in the PIMA dataset, which has 48.7% missing insulin values and 29.6% missing skin thickness), class imbalance (diabetic cases typically make up only 20–35% of samples), and the requirement for suitable feature scaling and categorical encoding strategies are some of the major preprocessing challenges faced by diabetes prediction datasets. Demographic bias plagues current diabetes prediction research, with gaps in socioeconomic variety and ethnic representation resulting from a preponderance of Western datasets. Model generalizability across various populations and healthcare contexts is further limited by temporal considerations, such as changing treatment procedures and lifestyles, as well as regional differences in healthcare systems and cultural views.

4 Dataset Design and Real-World Clinical Utility

The nature and composition of training data fundamentally shape how well predictive models perform in actual healthcare settings. Studies relying on limited or narrow datasets—such as the frequently cited PIMA Indian Diabetes Database—often demonstrate performance levels that don't translate to broader populations. These smaller datasets, while useful for testing algorithmic approaches, suffer from insufficient variation in ethnicity, sex, and geographical origin, which undermines their practical value in diverse clinical environments.

More promising results emerge from models trained on expansive, heterogeneous data sources like NHANES, BRFSS, UK Biobank, and multi-site hospital registries. These platforms offer several advantages: they capture wider demographic ranges, incorporate key diagnostic markers including HbA1c and glucose measurements, and in some cases provide longitudinal tracking that supports early risk identification rather than after-the-fact classification.

The evidence points clearly toward a design imperative: models destined for clinical implementation must be built on datasets that reflect actual patient

diversity, include medically relevant biomarkers, and ideally capture health trajectories over time rather than single snapshots.

5 Evaluation Framework for Diabetes Risk Prediction: Methodological Rigor and Clinical Relevance

Despite frequently impressive performance claims, comparing diabetes prediction models across different studies proves difficult due to varied assessment approaches and incomplete metric reporting. Studies often highlight singular measures like accuracy or area under the ROC curve while neglecting critical aspects such as validation methodology, outcome distribution imbalances, and the relative costs of different error types—factors that substantially affect practical utility.

Ensuring model reliability demands validation approaches that guard against overoptimistic estimates and information leakage. Proper temporal or random data partitioning, population-representative cross-validation techniques, and testing on entirely separate datasets from different institutions or regions serve as fundamental safeguards. Testing performance on external populations—particularly those from distinct demographic or healthcare contexts—remains surprisingly rare despite being crucial for deployment readiness.

Given the typical scarcity of diabetes cases in screening populations and the differential consequences of false predictions, single-metric assessments prove inadequate. Comprehensive evaluation should encompass multiple dimensions: the ability to correctly identify true cases (sensitivity), positive prediction accuracy (precision), harmonic performance balance (F1-score), and probability calibration. The alignment between predicted probabilities and actual outcomes becomes especially vital for screening programs and clinical decision systems, where miscalibrated estimates may trigger inappropriate treatment decisions.

Performance metrics alone cannot determine clinical value or fairness. Decision curve methodology quantifies net benefit across various probability thresholds for intervention, while disaggregated performance analysis across patient subgroups reveals potential inequities tied to age, gender, or racial background. These assessments prove indispensable for ensuring models serve diverse populations equitably.

Synthesizing these requirements yields a recommended evaluation framework for future research:

- Implement temporally and methodologically sound data partitioning with population-appropriate validation
- Validate findings using datasets from external sources or multiple healthcare systems
- Document comprehensive performance indicators relevant to clinical decision-making, emphasizing both discrimination and calibration
- Quantify practical value through decision-analytic frameworks
- Examine performance consistency and potential bias across patient demographic categories

Widespread adoption of these evaluation principles would enhance methodological transparency and accelerate the development of diabetes prediction systems suitable for real-world healthcare implementation.

6 Conclusion

With ensemble methods, deep learning models, and standard algorithms routinely surpassing established risk assessment tools, machine learning technologies have shown great potential for diabetes prediction. Opportunities for improved early diagnosis and individualized risk classification are presented by these developments. Critical obstacles, such as a lack of uniform evaluation frameworks, a lack of dataset diversity across global populations, and limited model interpretability that prevents clinical acceptance, restrict clinical translation.

Future research priorities must address these gaps through development of globally representative datasets, implementation of explainable AI methodologies, and rigorous prospective clinical validation studies. Success will ultimately depend on interdisciplinary collaboration to ensure algorithmic innovations translate into meaningful improvements in patient care and population health outcomes.

References

Ahmed, U., Issa, G., Aftab, S., Farhan Khan, M., Said, R., Ghazal, T., ... Khan, M. (2022). Prediction of diabetes empowered with fused machine learning. *IEEE Access*. doi: https://doi.org/10.1109/ACCESS.2022.3142097

Alanazi, A., & Mezher, M. (2020). Using machine learning algorithms for prediction of diabetes mellitus. In *Proceedings of iccit* (pp. 1–3). doi: https://doi.org/10.1109/ICCIT-144147971.2020.9213708

American Diabetes Association. (n.d.-a). *Gestational diabetes.* https://www.diabetes.org/diabetes/gestational-diabetes.

American Diabetes Association. (n.d.-b). *Type 1 diabetes.* https://www.diabetes.org/diabetes/type-1.

American Diabetes Association. (n.d.-c). *Type 2 diabetes.* https://www.diabetes.org/diabetes/type-2.

Aslan, M. F., & Sabanci, K. (2023). A novel proposal for deep learning-based diabetes prediction: Converting clinical data to image data. *Diagnostics*, *13*(4), 796. doi: https://doi.org/10.3390/diagnostics13040796

Atkinson, M. A., Eisenbarth, G. S., & Michels, A. W. (2014). Type 1 diabetes. *Lancet*, *383*(9911), 69–82. doi: https://doi.org/10.1016/S0140-6736(13)60591-7

Bhat, S. S., Selvam, V., Ansari, G. A., Ansari, M. D., & Rahman, M. H. (2022). Prevalence and early prediction of diabetes using machine learning in north kashmir: A case study of district bandipora. *Computational Intelligence and Neuroscience*, *2022*, 2789760. doi: https://doi.org/10.1155/2022/2789760

Butunoi, B.-P., Stolojescu-Crisan, C., & Negru, V. (2024). Blood glucose prediction in type 1 diabetes based on long short-term memory. In *Recent advances in artificial intelligence* (pp. 1–10). doi: https://doi.org/10.1007/978-3-031-70259-4_35

Centers for Disease Control and Prevention. (n.d.-a). *Behavioral risk factor surveillance system (brfss).* https://www.cdc.gov/brfss/annual_data/annual_data.htm.

Centers for Disease Control and Prevention. (n.d.-b). *National health and nutrition examination survey (nhanes).* https://wwwn.cdc.gov/nchs/nhanes/.

Defronzo, R. A., Ferrannini, E., Zimmet, P., & Alberti, G. (2015). *International textbook of diabetes mellitus.* John Wiley & Sons.

Doğru, A., Buyrukoglu, S., & Arı, M. (2023). A hybrid super ensemble learning model for the early-stage prediction of diabetes risk. *Medical & Biological Engineering & Computing*, *61*. doi: https://doi.org/10.1007/s11517-022-02749-z

Framingham Heart Study. (n.d.). *Framingham heart study dataset.* https://www.kaggle.com/datasets/aasheesh200/framingham-heart-study-dataset.

Hennebelle, A., Materwala, H., & Ismail, L. (2023). Healthedge: A machine learning-based smart healthcare framework for prediction of type 2 diabetes in an integrated iot, edge, and cloud computing system. *Procedia Computer Science*, *220*, 331–338. doi: https://doi.org/10.1016/j.procs.2023.03.043

International Diabetes Federation. (2023). *Diabetes facts and figures.* Retrieved from https://idf.org/ (Accessed 2025)

International Diabetes Federation. (2025a). *Country data: India.* Retrieved from https://diabetesatlas.org/ (Accessed 2025)

International Diabetes Federation. (2025b). *Data by location.* Retrieved from https://diabetesatlas.org/ (Accessed 2025)

International Diabetes Federation. (2025c). *Idf diabetes atlas reports.* Retrieved from https://diabetesatlas.org/ (Accessed 2025)

Joshi, T. N., Pramila, M., & Chawan, P. (2018). Logistic regression and svm based diabetes prediction system. *International Journal of Computer Applications*, *180*(20), 1–5.

Knip, M., & Simell, O. (2012). Environmental triggers of type 1 diabetes. *Cold Spring Harbor Perspectives in Medicine*, *2*(7), a007690. doi: https://doi.org/10.1101/cshperspect.a007690

Kokkorakis, M., et al. (2023). Effective questionnaire-based prediction models for type 2 diabetes across several ethnicities: a model development and validation study. *EClinicalMedicine*, *64*, 102235. doi: https://doi.org/10.1016/j.eclinm.2023.102235

Kopitar, L., Kocbek, P., Cilar Budler, L., Sheikh, A., & Stiglic, G. (2020). Early detection of type 2 diabetes mellitus using machine learning-based prediction models. *Scientific Reports*, *10*. doi: https://doi.org/10.1038/s41598-

020-68771-z

Mujumdar, A., & Vaidehi, V. (2019). Diabetes prediction using machine learning algorithms. *Procedia Computer Science*, *165*, 292–299. doi: https://doi.org/10.1016/j.procs.2020.01.047

Qi, H., Song, X., Liu, S., Zhang, Y., & Wong, K. K. L. (2023). Kfpredict: An ensemble learning prediction framework for diabetes based on fusion of key features. *Computer Methods and Programs in Biomedicine*, *231*, 107378. doi: https://doi.org/10.1016/j.cmpb.2023.107378

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, *380*(14), 1347–1358. doi: https://doi.org/10.1056/NEJMra1814259

Saru, S., & Subashree, S. (2019). Analysis and prediction of diabetes using machine learning. *International Journal of Emerging Technology and Innovative Engineering*, *5*(4).

Sneha, N., & Gangil, T. (2019). Analysis of diabetes mellitus for early prediction using optimal features selection. *Journal of Big Data*, *6*, 13. doi: https://doi.org/10.1186/s40537-019-0175-6

Sonar, P., & JayaMalini, K. (2019). Diabetes prediction using different machine learning approaches. In *Proceedings of iccmc* (pp. 367–371). doi: https://doi.org/10.1109/ICCMC.2019.8819841

Su, Y., Huang, C., Zhu, W., Lyu, X., & Ji, F. (2023). Multi-party diabetes mellitus risk prediction based on secure federated learning. *Biomedical Signal Processing and Control*, *85*, 104881. doi: https://doi.org/10.1016/j.bspc.2023.104881

Syed, A. H., & Khan, T. (2020). Machine learning-based application for predicting risk of type 2 diabetes mellitus (t2dm) in saudi arabia: A retrospective cross-sectional study. *IEEE Access*, *8*, 199539–199561. doi: https://doi.org/10.1109/ACCESS.2020.3035026

UCI Machine Learning Repository. (n.d.). *Pima indians diabetes database.* `https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database`.

Uloko, A. E., Musa, B. M., Ramalan, M. A., Gezawa, I. D., Puepet, F. H., Uloko, A. T., ... Sada, K. B. (2018). Prevalence and risk factors for diabetes mellitus in nigeria: A systematic review and meta-analysis. *Diabetes Therapy*, *9*(3), 1307–1316. doi: https://doi.org/10.1007/s13300-018-0441-1

World Health Organization. (2016). *Global report on diabetes.*

Zhang, L., Wang, Y., Niu, M., et al. (2020). Machine learning for characterizing risk of type 2 diabetes mellitus in a rural chinese population: the henan rural cohort study. *Scientific Reports*, *10*, 4406. doi: https://doi.org/10.1038/s41598-020-61123-x

Zhou, H., Xin, Y., & Li, S. (2023). A diabetes prediction model based on boruta feature selection and ensemble learning. *BMC Bioinformatics*, *24*, 224. doi: https://doi.org/10.1186/s12859-023-05300-5

Journal of Behavioral Data Science, 2026, *6* (1), 152–164.
DOI: https://doi.org/10.35566/jbds/tonghl

EmojiSentR: An R Package for Integrated Text and Emoji Sentiment Analysis

Logan Hanson[0009−0008−4669−0162], Elias Lahrim, and Xin Tong*[0000−0003−3050−1554]

Department of Psychology, University of Virginia, Charlottesville, VA 22904, USA
loganh11702@gmail.com; elahrim@gmail.com; xt8b@virginia.edu

Abstract.

Although emojis are used by 92% of the world's online population, appear in over one quarter of social media posts, and carry affect beyond words, most software packages for sentiment analysis ignore emoji semantics. Our newly developed R package, `EmojiSentR`, addresses this gap by merging emoji valence with word-level sentiment in a tidy R workflow. The package bundles an internal lexicon derived from the Novak-1200 dataset (1,200 glyphs × 8 emotions) and a pipeline function, `sentiment_analysis()`, that (1) extracts emojis, (2) cleans residual text, (3) scores each channel separately, (4) leverages VADER's (Valence Aware Dictionary and sEntiment Reasoner) simple negation, and (5) returns a weighted composite score. Furthermore, by adding Unicode-cleanup and Poppler-based PDF-to-text utilities, the package also broadens data-source coverage. In an illustrative example, on a corpus of 17,996 English tweets, `EmojiSentR` changed the predicted polarity in 20.7% of messages while adding only 769.4 ms of computation time per 1,000 posts. For instance, polarity reversals appeared in sarcastic laughter ("Great… 😂") and uncertainty cues ("Ok 😬"). The `EmojiSentR` package makes it simple and convenient to treat emojis as first-class sentiment carriers within R. Its modular design supports user-tunable weights, transparent lexicon updates, and forthcoming multilingual extensions, offering researchers a reproducible tool for analyzing emoji-rich text as well as PDF-based data sources.

Keywords: Emoji · Sentiment analysis · Text mining · R package · PDF extraction

1 Introduction

Digitally mediated communication has made text one of the most abundant forms of data in research (Aggarwal, 2015). News articles, social media posts, policy documents, customer reviews, electronic health records, psychotherapy

transcripts, scientific literature, and open-ended survey responses capture attitudes, behaviors, and events at scale and in near real time. Harnessing these sources enables researchers to track public opinion, monitor health and education outcomes, study misinformation, evaluate programs, and generate hypotheses that complement traditional quantitative measures (e.g., Humphreys & Wang, 2018; Khaiser et al., 2023; Kozik et al., 2022; Machová et al., 2023; Thakur, 2023; Westgate et al., 2015).

Text analysis, or text mining, is the process of converting unstructured texts into structured information (e.g., numerical data) so it can be searched, summarized, modeled, and used for inference or prediction. Text analysis methods range from simple lexicons to modern deep learning (e.g., Hotho et al., 2005; Minaee et al., 2021; Wankhade et al., 2022). As an important category of text analysis, sentiment and emotion mining has been broadly applied across many fields to infer affective tone and evaluative stance expressed in text (Itani et al., 2017; Liu et al., 2025). Sentiment typically measures polarity and intensity (positive, negative, or neutral), while emotion targets specific feelings (e.g., happiness, anger, sadness). Despite this difference, one key facet of sentiment and emotion mining is polarity detection, which employs deep learning-based, machine learning-based, or lexicon-based approaches to discern the polarities in a text, as highlighted in Nandwani and Verma (2021). Other categories of text analysis include topic modeling, which can be used to detect latent topics in text data (Blei & Lafferty, 2007; Blei et al., 2003); LLM-based classification, which uses a prompted large language model to label or categorize texts without task-specific training (Hassani et al., 2025); etc.

Dedicated toolkits have made the text analysis pipeline virtually turnkey. In R software, packages such as `tm`, `quanteda`, and `tidytext` provide tools for corpus management, tokenization, and document-feature matrix construction (Gagolewski, 2024; Wickham, 2024); `sentimentr`, `syuzhet`, and `VADER` implement lexicon- and rule-based polarity scoring (Hutto & Gilbert, 2014; Jockers, 2020; Rinker, 2022); and topic-model frameworks such as `stm` (Roberts et al., 2019) and `text2vec` (Selivanov et al., 2024) enable estimation of latent thematic structures. More recently, R interfaces to large language model APIs have added embedding extraction and generative capabilities. In short, analysts can move from messy text to predictive or explanatory models with only a few lines of code.

Yet these workflows still treat emojis (now a core element of online speech) as either noise to be stripped or crude word-like tokens. They are anything but fringe: Emojipedia reported that more than 26% of all tweets in 2023 contained at least one emoji (Broni, 2023) and Adobe's global survey found that 92% of internet users rely on emojis to communicate across language barriers (Adobe Inc., 2022). Despite this prevalence, most text analysis toolkits (e.g., R sentiment libraries) ignore emoji semantics or rely on small, static lookup tables that lack polarity scores and context handling. Analyses that omit pictographic affect therefore risk under-estimating emotion, misclassifying polarity, and missing subtle cues such as sarcasm conveyed by 🤨 or the tonal shift from 😊 to 😬.

To fill this gap, we focus on the lexicon-based sentiment analysis, which is simple but commonly used in practice, develop and introduce `EmojiSentR`. `EmojiSentR` is an R package that fuses a purpose-built emoji-sentiment lexicon with conventional text-sentiment engines, yielding a single weighted score that captures both modalities. The package (i) expands the publicly available Novak-1200 ratings into an internal dataset; (ii) offers pipe-friendly functions for extracting, scoring, and context-adjusting emojis; (iii) provides Unicode-cleanup and Poppler-based PDF-to-text utilities; and (iv) blends emoji and textual sentiment through user-tunable weights, enabling richer analyses of social-media and other text-form communication.

The remainder of this article is organized as follows. We first review existing work in the literature and discuss the rationale for `EmojiSentR`. The next section explains how the emoji lexicon was derived, how text- and emoji-level polarity are combined, and how the package is installed and invoked from R. Then we present a brief demonstration on a small tweet corpus, reporting the proportion of polarity flips and the additional run-time required. At the end, we interpret these findings, outline practical applications, recognize current limitations, and sketch directions for future expansion of the lexicon and software.

2 Related Work and Rationale for `EmojiSentR`

This section explains why emojis matter for sentiment analysis, outlines the limits of text-only approaches, summarizes prior emoji research, surveys relevant R tooling, and closes with the specific gap that our package fills.

2.1 Emojis

An emoji is a standardized pictographic character (e.g., 😂, 😍, or 🚀) encoded in Unicode and displayed on digital devices to convey emotion, objects, concepts, or actions within text. Originating from Japanese mobile messaging in the late 1990s, emojis are now supported across smartphones, computers, and web platforms, and have become a routine part of digital communication, allowing writers to convey tone and visual nuance (Broni, 2023). Key benefits of using emojis include simplifying communication by conveying emotion quickly, solving miscommunication caused by ambiguous messaging, and fostering a greater sense of connection in digital interactions. For example, although "This is fine" is technically an affirmative response, it can also serve as a signal of the sender's dissatisfaction. An added emoji can easily clarify the intended tone, e.g., "This is fine 😊" conveys genuine approval, whereas "This is fine 🤨" reveals underlying frustration.

Therefore, when sentiment analyses ignore emojis, polarity estimates can be inaccurate or biased, e.g., missing sarcasm flagged by an emoji or positive reinforcement added via emoji intensifiers. Because downstream tasks (e.g., monitoring attitudes, auditing content) depend on accurate affect estimation, incorporating emoji signals is practically important rather than decorative.

2.2 Limits of text-only sentiment analysis

Previous studies have shown that emojis encode sentiment and can improve affect modeling when integrated appropriately (Boia et al., 2013; Kralj Novak et al., 2015). However, commonly used lexicon-based tools such as `VADER`, `syuzhet`, and `sentimentr` principally score tokens from plain text (Hutto & Gilbert, 2014; Jockers, 2020; Rinker, 2022). While these methods provide fast, interpretable baselines, they may struggle when emojis flip or amplify sentiment, especially when negation scopes across text-emoji mixes (e.g., "not bad 😍"), or when tone is conveyed primarily by pictographs. In such cases, the text-only pipelines can under- or over-estimate polarity. These open issues motivate practical tooling that treats emojis as first-class affective signals rather than post-hoc decorations or annotations.

Within the R software, a robust suite of libraries already supports many core text-mining tasks. Widely used R packages, such as `stringi`, `stringr`, `textclean`, `utf8`, and `text2vec` support tokenization, string handling, and sentiment scoring (Gagolewski, 2024; Rinker, 2023; Selivanov et al., 2024; Urbanek, 2023; Wickham, 2024). Topic modeling and downstream exploration are often implemented with `stm` (Roberts et al., 2019). For emojis, packages such as `emoji`, `emo`, and `emojifont` expose Unicode metadata, aliases, and glyph support (Rudis & Robinson, 2024; Wickham et al., 2022; Yu, 2022). Yet, despite this rich ecosystem, there is still no tidy, end-to-end workflow that simultaneously extracts emojis, scores them with a dedicated emoji lexicon, and fuses text and emoji sentiment with transparent weighting.

2.3 Rationale for `EmojiSentR`

In this study, we propose that implementing sentiment analysis in R based on the following sequence: (i) prepare and normalize text (tokenization, string handling, cleaning), (ii) extract emojis from the same text, (iii) score words with a standard text-lexicon method, (iv) score emojis with an emoji-specific lexicon, and (v) combine both channels with a transparent weighting scheme calibrated to the task. This "text-plus-emoji" storyline is the through line of our work, which is operationalized in the next section.

The developed `EmojiSentR` package bundles the above sequence into a tidy, end-to-end workflow that extracts emojis, scores text and emojis in dedicated channels and returns a tunable weighted composite. The package delivers a lexicon of 1,200 emojis (derived from the Novak-1200 ratings and aligned with Unicode 15) and provides a pipe-friendly function leveraging `VADER's` negation returning a tidy tibble ready for widely used R packages such as `dplyr`, `ggplot2`, or `tidymodels`. By integrating emoji semantics alongside text, rather than treating them as noise, the approach reduces polarity errors in emoji-heavy content while remaining simple enough for routine analysis.

In addition, text data for sentiment analysis are usually ingested into R as plain-text files (.txt or .csv) because these formats are straightforward to

parse and tokenize. But much of the material that researchers use, such as policy briefs, historical documents, court opinions, and annual reports, circulates only as PDFs. Extracting clean text from PDFs in R typically requires juggling separate utilities such as `Poppler`, post-processing errant line breaks, and troubleshooting encoding issues, all of which create friction and discourage analysts from incorporating valuable sources. Our `EmojiSentR` package treats PDF input as first-class data, with which users can directly import PDF files, automatically handle Unicode cleanup, and receive a tidy tibble ready for sentiment analysis. The package makes sentiment workflows as seamless for PDFs as they are for traditional text files, broadening the empirical reach of text mining in R.

3 The `EmojiSentR` package

In this section, we detail how the emoji lexicon is constructed and how the pipeline implements proposed steps with R code.

3.1 Installation, package overview, and intended uses

Installation. The `EmojiSentR` package is hosted on GitHub and can be installed by typing the following code into R:

```
install.packages("devtools")
devtools::install_github("eliaslah/EmojiSentR", subdir
    =  "unicode")
library(EmojiSentR)
```

Users who plan to extract PDFs must also have Poppler (`pdftotext`) installed (Windows: Oschwartz build; macOS: brew install poppler; Linux: apt-get install poppler-utils).

If installation errors occur, users may alternatively download the package source from the GitHub repository, place it in a local directory, and install from source using the appropriate `install.packages()` path. Troubleshooting steps can be performed in parallel with any preferred large language model to identify and resolve dependency or configuration issues. Be sure to check the functions within the scripts on GitHub to leverage `?function_name` for all package applications.

Package overview. Loading the library exposes two public objects:

- `emoji_lexicon`, a tibble that maps emoji code points to sentiment scores, and
- `sentiment_analysis()`, an end-to-end helper that extracts emojis, scores text and emojis separately, and returns a single weighted sentiment index.

The NAMESPACE also exports four utility functions, `remove_unicode`, `mutate_unicode_cols`, `clean_with_poppler`, and `save_pdf_as_text`, which support Unicode cleaning and PDF ingestion. These aspects are pipe-friendly and have minimal additional dependencies.

A quick start example is provided below.

```
txt <- c("I love this 😍",
         "Great... 😂",
         "Not impressed 🤨")
sentiment_analysis(txt)
```

Table 1. Example outputs from `sentiment_analysis()` with emoji rendering.

clean_text	emojis	text_sentiment	emoji_sentiment	combined_sentiment
I love this 😍	😍	0.60	0.75	0.63
Great... 😂	😂	0.05	-0.12	-0.03
Not impressed 🤨	🤨	-0.45	-0.18	-0.41

By default, the weights for text sentiment and emoji sentiment are 0.7 and 0.3, respectively, when calculating the combined sentiment. Namely, `sentiment_analysis(txt)` and `sentiment_analysis(txt, text_weight = 0.7, emoji_weight = 0.3)` give the same output. The defaults are retained for backward compatibility. Custom weights could also be specified. For example, for emoji-heavy text, users can assign a higher weight for emojis:

```
sentiment_analysis(txt, text_weight = 0.5, emoji_weight
    = 0.5)
```

Note that negation handling is often challenging in text analysis, as negation ("not happy," "never went") can flip or dampen the polarity and meaning of entire phrases. With emojis, the problem of negation could be reduced. For example, `sentiment_analysis("not bad")` provided a text sentiment score of 0.431 while `sentiment_analysis("not bad 😊")` provides an emoji sentiment of 0.644 and a combined sentiment score of 0.495.

The `EmojiSentR` package also provides optional Unicode and PDF utilities. To normalize or remove Unicode characters from input strings, use `remove_unicode(txt)`. For PDF documents, use `clean_with_poppler("path/to/file.pdf")` to extract UTF-8 text (requires Poppler's `pdftotext`).

Intended uses. The package is designed for researchers and analysts who need to incorporate emojis into sentiment analysis pipelines without wrestling with ad-hoc preprocessing. It ingests plain text or PDF-extracted content, tokenizes words and emojis in a single step, and enables users to quantify affect in sources where emojis carry crucial tonal cues, including social media dashboards, chat transcripts, psychology studies of diary entries, and classroom demonstrations of feature engineering. In short, it turns mixed text-and-emoji streams into ready-to-model data, enabling more accurate, interpretable sentiment estimates in modern, emoji-rich communication.

In the next subsection, we explain, in plain language, how the emoji lexicon is built and stored, then outline the end-to-end pipeline.

3.2 Emoji lexicon construction

The setup script reads the Novak-1200 emoji table, computes one scalar sentiment per emoji as the simple mean of its eight Plutchik emotion ratings and saves the result as an internal package object (`load_novak_table`). This build happens once at development time, so users do not need to regenerate the lexicon; it loads automatically when the package is attached.

The lexicon is built by the script `sentiment_helpers.R` as shown below.

```
load_novak_table <- function() {

  fn <- system.file("extdata",
            "emoji_sentiment_novak_unicode.rds",
             package = utils::packageName())

  if (fn == "")
    stop("Bundled Novak emoji table not found in
       package.")

  readRDS(fn)
}
```

- Source dataset – Novak-1200 with eight Plutchik emotion ratings per glyph.
- Aggregation rule – arithmetic mean of the eight columns; no scaling or polarity weighting.
- Storage – the resulting tibble is saved from an internal data object `emoji_sentiment_novak_unicode.rds` and loads automatically when the package is attached.

With the lexicon in place, the pipeline can combine emoji-level and text-level scores for any input string.

3.3 Sentiment pipeline

The `sentiment_analysis()` function executes a five-step workflow:

(i) Emoji extraction - We first extract all emoji code-points from each message (e.g., via a Unicode-aware regex (regular expression)) and store them as a per-message list. The regex targets the primary pictograph blocks and is conservative by design; it can be extended if a dataset uses additional ranges. For example,

```
[\U0001F600-\U0001F64F\U0001F300-\U0001F5FF\U0001F680-\
   U0001F6FF]
```

isolates all pictographs in each string.

(ii) Text cleansing – We remove the extracted emojis from the text and apply light normalization (e.g., lowercasing, punctuation/HTML cleanup). Removing

emojis before text scoring prevents double-counting the same affective cue across channels.

(iii) Word-level sentiment – We compute a baseline text polarity using a standard lexicon-based method (with simple handling of negators and intensifiers; e.g., the VADER negation detector ("not", "no", "never") downweights the score). This yields a text-only polarity that we will merge with the emoji channel.

(iv) Emoji-level sentiment – each extracted emoji is matched in emoji_lexicon and averaged.

(v) Weighted fusion – the final score is a convex combination of the two channels:

```
FinalScore = (text_weight*TextSentiment) + (emoji_
    weight*EmojiSentiment)
```

For example, with default weights text_weight = 0.7 and emoji_weight = 0.3, FinalScore = (0.7 * TextSentiment) + (0.3 * EmojiSentiment). If defaults are not desired, weights are user-settable, constrained to [0,1], and must sum to 1. Edge cases are handled gracefully: if a message contains no emojis, the text score is returned; if it is emoji-only, the emoji score is returned; if neither is present, the result is NA.

With the weight tuning parameters, researchers may perform a sensitivity analysis, using weight pairs (e.g., 0.9/0.1, 0.7/0.3, 0.5/0.5, 0.3/0.7, and 0.1/0.9) on a small labeled set, checking the robustness of the analysis, and picking the best-performing pair.

3.4 Unicode-cleaning and PDF-ingestion utilities

These utilities normalize Unicode and preserve emoji code-points so downstream tokenization and scoring behave consistently across platforms. The functions `remove_unicode()` and `mutate_unicode_cols()` strip emojis, pictographs, and control characters from individual strings or entire data-frame columns, enabling analysts to create parallel text-only datasets. The functions `clean_with_poppler()` and `save_pdf_as_text()` wrap the Poppler tool `pdftotext`, converting PDF survey responses or social-media screenshots to UTF-8 text for downstream sentiment analysis.

3.5 Scope, current status, and limitations

Before turning to the results, we briefly note the scope and limitations of the current release. The `EmojiSentR` package uses base-emoji entries; skin-tone and gender modifiers do not change the base emoji's score, and platform-specific renderings are not modeled. The emoji-extraction regex is conservative but extensible if a dataset requires additional Unicode ranges. Negation handling is

intentionally simple for transparency, and the default weights reflect typical text-first sentiment but are user-tunable. Finally, the implementation assumes English-centric tokenization; multilingual or domain-specific slang may require custom tweaks.

The package compiles and passes R CMD check on all major platforms. The core pipeline is fully functional. Planned enhancements (e.g., variant collapsing, and expanded benchmarks) are described in Section 5.3.

4 Applications

This section provides an illustrative example, including a simple weight override to show how tuning the weight affects the final polarity. See the runnable code in section 3.1 for defaults, weight overrides, negation, and Unicode/PDF helpers, for implementation.

We tested `EmojiSentR` on a convenient sample of 17,996 English tweets collected from X/Twitter, of which 15,337 contained at least one emoji. Scoring the entire set required only an additional 769.4 ms of computation time per 1,000 posts (Apple M2, 8 GB RAM). Compared with the text-only baseline (using `sentimentr`), `EmojiSentR` changed the predicted polarity in 20.7% of cases.

To further examine the package's PDF workflow, we compared polarity classifications obtained from plain-text inputs with those obtained after converting the same content from PDF using `clean_with_poppler()`. Across a sampled set of tweets, PDF-derived polarity matched the plain-text result in 63.3% of cases and differed in 36.7% of cases. These findings suggest that the PDF helper is useful for document-based text ingestion, while also indicating that PDF-to-text extraction artifacts can affect downstream sentiment scoring in a subset of cases. Table 2 lists illustrative examples.

Table 2. Illustrative polarity shifts introduced by the emoji channel in plain text and PDF-derived text.

Text / File (excerpt)	text-only polarity	EmojiSentR polarity	effect
Dose 2! (Moderna)... 😊	neutral	positive	flip
@ctraywick ... 😍	negative	positive	flip

On a 300-tweet subset manually labeled as positive, neutral, or negative, the text-only baseline achieved a macro-F1 of 0.427, whereas `EmojiSentR` achieved 0.437. This result indicates that incorporating EmojiSentR can improve classification performance on emoji-rich social media text. In particular, the emoji channel helps capture affective emphasis, stance, and tone cues that are not always reflected in text alone, as illustrated by the examples in Table 2.

5 Discussion

In sum, on a pilot set of 17,996 English tweets, `EmojiSentR` shifted predicted sentiment in 20.7 % of messages, most often correcting cases of sarcasm or emoji-heavy enthusiasm (e.g., Table 1). On a 300-tweet manually labeled subset, the text-only baseline achieved a macro-F1 of 0.427 versus 0.437 for `EmojiSentR`, demonstrating a better performance of `EmojiSentR`. The additional processing time was modest, 769.4 ms per 1,000 posts, making the approach practical for real-time analysis.

5.1 Implications for sentiment analysis

Methodologically, the results confirm that pictographic affect can be integrated into classic lexicon pipelines without resorting to deep-learning infrastructure. The findings also support theories that view emojis as affective amplifiers. For quantitative psychological work, such as monitoring social media expressions, quantifying public stigma, and tracking shifts in sentiment, `EmojiSentR` offers a reproducible way to capture emotional nuance that would otherwise be lost.

Because `EmojiSentR` outputs tidy tibbles, analysts can drop sentiment scores directly into social-media dashboards, mixed-method studies that triangulate survey data with textual affect, or classroom demonstrations of feature engineering in R.

5.2 Limitations

The current lexicon is English-centric and may not capture cultural or temporal drift in emoji meaning. The negation module handles only simple cues ("not", "no", "never") and misses sarcasm or multimodal context (GIFs, images). Validation to date is limited to 17,996 English tweets collected from X/Twitter, so generalizability to long-form or multilingual corpora is uncertain. Finally, the lexicon will need periodic updates as Unicode expands, and a preliminary PDF test revealed a 36.7% of score mismatches, indicating that further refinement of the Poppler wrapper is necessary for full cross-format consistency. In conclusion, common evaluation metrics may under-reward the nuanced effect that emojis contribute, suggesting the need for richer annotation schemes in future benchmarks.

5.3 Future Work

Planned extensions include crowd-sourcing multilingual emoji ratings, handling skin-tone and gender variants, exploring a lightweight embedding back-end, and releasing a Shiny GUI for drag-and-drop sentiment scoring.

In summary, `EmojiSentR` fills a long-standing methodological gap by treating emojis as first-class sentiment carriers within tidy R workflows. Preliminary tests show that adding an emoji channel alters polarity predictions in a substantial

subset of social-media posts, demonstrating the analytical cost of ignoring pictographic affect. The package's lightweight, modular design, coupled with fully open-source code, enables straightforward replication, extension, and integration into dashboards or experimental pipelines. Although the present release relies on an English-only lexicon and exhibits occasional PDF-processing mismatches, the architecture is engineered for rapid Unicode updates and forthcoming multilingual tables. Collectively, `EmojiSentR` offers behavioral researchers a reproducible, practical tool for capturing the full emotional nuance of contemporary digital text while laying a clear path for future accuracy benchmarks and feature growth.

Author Notes

Correspondence should be sent to Xin Tong, Department of Psychology, University of Virginia, Charlottesville, VA 22904; Email: `xt8b@virginia.edu`

Author Contributions

Conceptualization, L.H. (Logan Hanson) and E.L. (Elias Lahrim); Methodology, L.H. and E.L.; Software, E.L.; Validation, L.H. and E.L.; Data curation, E.L. and L.H.; Writing—original draft, L.H.; Writing—review & editing, L.H. and X.T. (Xin Tong); Visualization, L.H. and E.L.; Project administration, X.T.; Supervision, X.T. L.H. and E.L. contributed equally to this work.

Data Availability Statement

All code and data are openly hosted at `https://github.com/eliaslah/EmojiSentR`, including the package source, the 17,996 tweet evaluation corpus, the 300 tweet labeled subset, the internal Novak 1200 emoji lexicon, and scripts to reproduce the benchmarks.

References

Adobe Inc. (2022). *Future of creativity: 2022 global emoji trend report.* (Retrieved from https://www.adobe.com/)

Aggarwal, C. C. (2015). *Data mining: The textbook*. Cham: Springer. doi: https://doi.org/10.1007/978-3-319-14142-8

Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, 17–35. doi: https://doi.org/10.1214/07-aoas114

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022. doi: https://doi.org/10.7551/mitpress/1120.003.0082

Boia, M., Faltings, B., Musat, C., & Pu, P. (2013). A study of user assessment of the emoji sentiment lexicon. *Proceedings of the International Conference on Social Informatics*, 24–33.

Broni, K. (2023). *What's new on the 10th annual world emoji day.* (Retrieved from https://blog.emojipedia.org/)

Gagolewski, M. (2024). stringi: Fast and portable character string processing in r [Computer software manual]. Retrieved from https://cran.r-project.org/package=stringi (R package version 1.8.4)

Hassani, S., Sabetzadeh, M., & Amyot, D. (2025). An empirical study on llm-based classification of requirements-related provisions in food-safety regulations. *Empirical Software Engineering*, *30*(3), 72. doi: https://doi.org/10.1007/s10664-025-10619-z

Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. *LDV Forum*, *20*(1), 19–62. doi: https://doi.org/10.21248/jlcl.20.2005.68

Humphreys, A., & Wang, R. J. H. (2018). Automated text analysis for consumer research. *Journal of Consumer Research*, *44*(6), 1274–1306. doi: https://doi.org/10.1093/jcr/ucx104

Hutto, C. J., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international aaai conference on web and social media* (Vol. 8, pp. 216–225). doi: https://doi.org/10.1609/icwsm.v8i1.14550

Itani, M., Roast, C., & Al-Kjayatt, S. (2017). Developing resources for sentiment analysis of informal arabic text in social media. *Procedia Computer Science*, *117*, 129–136. doi: https://doi.org/10.1016/j.procs.2017.10.101

Jockers, M. L. (2020). syuzhet: Extract sentiment and plot arcs in novels [Computer software manual]. Retrieved from https://cran.r-project.org/package=syuzhet (R package version 1.0.6)

Khaiser, F. K., Saad, A., & Mason, C. (2023). Sentiment analysis of students' feedback using text-based classification and nlp. *Journal of Language and Communication*, *10*(1), 101–111. doi: https://doi.org/10.47836/jlc.10.01.06

Kozik, R., Kula, S., Choraś, M., & Woźniak, M. (2022). Technical solution to counter potential crime: Text analysis to detect fake news and disinformation. *Journal of Computational Science*, *60*, 101576. doi: https://doi.org/10.1016/j.jocs.2022.101576

Kralj Novak, P., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of emojis. *PLOS ONE*, *10*(12), e0144296. doi: https://doi.org/10.1371/journal.pone.0144296

Liu, H., Tsang, S., Wood, A., & Tong, X. (2025). Longitudinal sentiment analysis with conversation textual data. *Fudan Journal of the Humanities and Social Sciences*, *18*(1), 193–214. doi: https://doi.org/10.1007/s40647-024-00417-0

Machová, K., Szabóová, M., Paralič, J., & Mičko, J. (2023). Detection of emotion by text analysis using machine learning. *Frontiers in Psychology*, *14*, 1190326. doi: https://doi.org/10.3389/fpsyg.2023.1190326

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, *54*(3), 1–40. doi: https://doi.org/10.1145/3439726

Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, *11*(1), 81. doi: https://doi.org/10.1007/s13278-021-00776-6

Rinker, T. W. (2022). sentimentr: Calculate text polarity sentiment [Computer software manual]. Retrieved from https://cran.r-project.org/package=sentimentr (R package version 2.9.0)

Rinker, T. W. (2023). textclean: Text cleaning tools [Computer software manual]. Retrieved from https://cran.r-project.org/package=textclean (R package version 0.9.3)

Roberts, M. E., Stewart, B. M., & Tingley, D. (2019). stm: An r package for structural topic models. *Journal of Statistical Software*, *91*(2), 1–40. doi: https://doi.org/10.18637/jss.v091.i02

Rudis, B., & Robinson, D. (2024). emoji: Data and functions to work with emojis [Computer software manual]. Retrieved from https://cran.r-project.org/package=emoji (R package version 0.2.0)

Selivanov, D., Wang, Q., & Tang, Y. (2024). text2vec: Modern text mining framework for r [Computer software manual]. Retrieved from https://cran.r-project.org/package=text2vec (R package version 0.6)

Thakur, N. (2023). Sentiment and text analysis of public discourse on twitter about covid-19 and mpox. *Big Data and Cognitive Computing*, *7*(2), 116. doi: https://doi.org/10.3390/bdcc7020116

Urbanek, S. (2023). utf8: Unicode text processing [Computer software manual]. Retrieved from https://cran.r-project.org/package=utf8 (R package version 1.2.3)

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*(7), 5731–5780. doi: https://doi.org/10.1007/s10462-022-10144-1

Westgate, M. J., Barton, P. S., Pierson, J. C., & Lindenmayer, D. B. (2015). Text analysis tools for identification of emerging topics and research gaps in conservation science. *Conservation Biology*, *29*(6), 1606–1614. doi: https://doi.org/10.1111/cobi.12605

Wickham, H. (2024). stringr: Simple, consistent wrappers for common string operations [Computer software manual]. Retrieved from https://cran.r-project.org/package=stringr (R package version 1.6.3)

Wickham, H., Francois, R., & D'Agostino McGowan, L. (2022). emo: Easily insert emojis into r documents [Computer software manual]. Retrieved from https://github.com/hadley/emo (R package version 0.0.0.9000)

Yu, G. (2022). emojifont: Emoji and font awesome in graphics [Computer software manual]. Retrieved from https://cran.r-project.org/package=emojifont (R package version 0.5.6)