# Supporting information for:

# “Measuring Distances in High Dimensional Spaces

## Why Average Group Vector Comparisons Exhibit Bias, And What to Do About it”

Breanna Green, William Hobbs, Sofia Avila, Pedro Rodriguez, Arthur Spirling, Brandon M. Stewart

## Table of Contents

# A Illustration of correction with R code

```
library(MASS)
library(tidyverse)
library(broom)
```

## Example debiasing function

This is a simplified illustration of our debiasing method. The `conText` package will be implemented more efficiently and robustly. **The debiasing step in the function below is highlighted with ####.**

```
debiased_estimates <- function(
    mod # e.g., model output from lm() -- R's main linear regression function
    # for independent observations
    # or the estimatr package's lm_robust() to calculate clustered standard errors
    # for non-independent observations
) {
  mod_df <- tidy(mod) # convert model summary to data frame
  #
  # (unbiased) beta hats to (biased) squared beta hats
  mod_df$biased_sqrd_beta <- mod_df$estimate^2
  #
  mod_df$beta_variance <- mod_df$std.error^2 # beta standard error to beta variance
  #
  ##### This is debiasing step: ####
  # subtract estimated beta variance from squared beta hats
  mod_df$debiased_sqrd_beta <- mod_df$biased_sqrd_beta - mod_df$beta_variance
  #
  return(mod_df)
}
```

## Simulate data

In this simulation, the true value (i.e., the expected value) of the squared Euclidean norm on the difference between embeddings vectors is 0 – because the groups have been randomly assigned. We demonstrate that the debiased estimator returns 0 in the section "Average of 10,000 estimates" below.

```
simulate_data_k2 <- function(n = 500, group_imbalance = c(0.9, 0.1)) {
  list(
    # a (random) dummy variable for group membership
    random_groups = sample(c(0, 1), size = n, replace = TRUE, prob = group_imbalance),
    # these k=2 "embeddings" are just the example in the help file of mvrnorm()
    embeddings =  mvrnorm(n = n, mu = rep(0, 2), matrix(c(10,3,3,2),2,2))
  )
}
```

## Single dimension illustration

```
set.seed(987654321)

simulated_data <- simulate_data_k2()
```

```r
embeddings <- simulated_data[["embeddings"]]
random_groups <- simulated_data[["random_groups"]]

mod_d1 <- lm(
  # run a regression with the group indicator as x and the first embedding dimension as y
  embeddings[,1] ~ random_groups
)
```

```r
debiased_estimates(mod_d1) |>
  select(term, biased_sqrd_beta, beta_variance, debiased_sqrd_beta) |>
  filter(term != "(Intercept)")
# we remove the intercept estimate for this illustration but it can be used in
# intercept only models, e.g., y ~ 1 and one of these intercept regressions for each
# compared group, to correct the denominator of a cosine similarity calculation
```

| term | biased_sqrd_beta | beta_variance | debiased_sqrd_beta |
|---|---:|---:|---:|
| random_groups | 0.24 | 0.21 | 0.03 |

## Multiple dimension illustration

```r
mod_d2 <- lm( # run a separate regression with the second embedding dimension as y
  embeddings[,2] ~ random_groups
)

all_debiased_sqrd_betas <- bind_rows(
  # stack estimates from models 1 and 2 for the squared Euclidean norm below
  debiased_estimates(mod_d1),
  debiased_estimates(mod_d2)
) |>
  filter(term != "(Intercept)")
```

```r
all_debiased_sqrd_betas |>
  # calculate the squared Euclidean norm for each x variable (here, only 1 of them)
  group_by(term) |>
  summarize(
    biased_sqrd_euclidean_norm = sum(biased_sqrd_beta),
    debiased_sqrd_euclidean_norm = sum(debiased_sqrd_beta)
  )
```

| term | biased_sqrd_euclidean_norm | debiased_sqrd_euclidean_norm |
|---|---:|---:|
| random_groups | 0.28 | 0.02 |

## Average of 10,000 estimates

Repeating the above code for 10,000 simulated samples, mean estimates are:

| term | biased_sqrd_euclidean_norm.mean | debiased_sqrd_euclidean_norm.mean |
|---|---:|---:|
| random_groups | 0.27 | 0 |

# B Twitter data tests

For the Twitter data tests, we use data from a panel of Twitter users, described in (Hughes et al., 2021). Users in this panel were linked to voter records, which included basic demographic information and vote histories. We down-sample the large panel to only users whose user IDs ended with eight, and analyzed tweets between January 2019 and February 2023 that contained the words "children" (illustration of bias in Figure 1), "people" (large sample illustration of bias and correction in Figure 3), or "racism" (Tables C.8 and C.9; Figures C.12 and C.13 – assessing correction for plausibly larger main effects and covariate effects).

Context-dependent word embeddings are drawn from the a la carte embedding approach described in (Rodriguez et al., 2023). This approach assigns context-dependent word embeddings (the 200d Twitter embeddings from GloVe (Pennington et al., 2014): `https://nlp.stanford.edu/projects/glove/`) for the word 'people' based on the words that appear near the word people in each tweet. Our analyses study the squared Euclidean norm of distances across groups for these embeddings.

# C Supplementary figures and tables

## C.1 Illustration of bias from folding

If, across samples or noisy measurements, the values that we estimate for $\beta$ (the unobserved sampling distribution of $\hat{\beta}$) are sometimes greater than their true values and sometimes less than, our distances are nonetheless always positive – and so, in expectation, greater than the true value of $\beta$.

We illustrate this folding effect in Figure C.1.

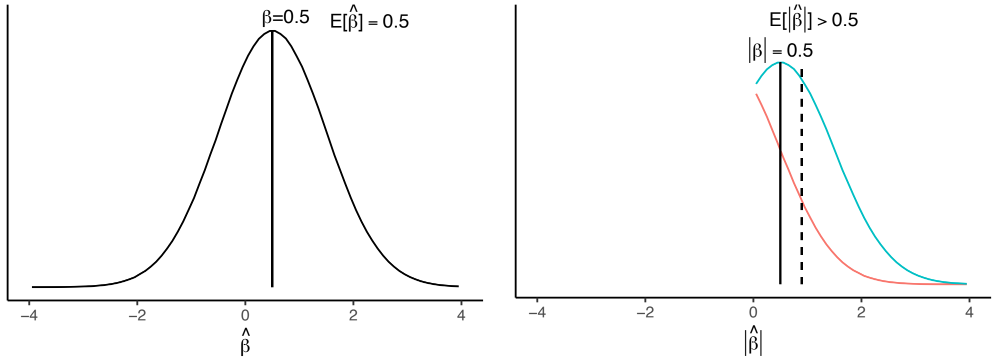

Figure C.1: Illustration of bias from folding. In the right panel, negative values (in red) for $\hat{\beta}$ become positive after squaring and then taking the square root (equivalent to $|\hat{\beta}|$ in this uni-dimensional illustration).

Less intuitively, *squared* Euclidean distance estimates are biased even when the sampling (or measurement error) distribution (unrealistically) never spans 0. If we split a positively or negatively bounded (and non-constant) distribution in exactly half at its expected value, the expected value of the half further from 0 will increase *more* (or, if originally less than 1, decrease less) after squaring than the expected value of the half closer to it will.

## C.2 (Unsquared) corrected Euclidean distance

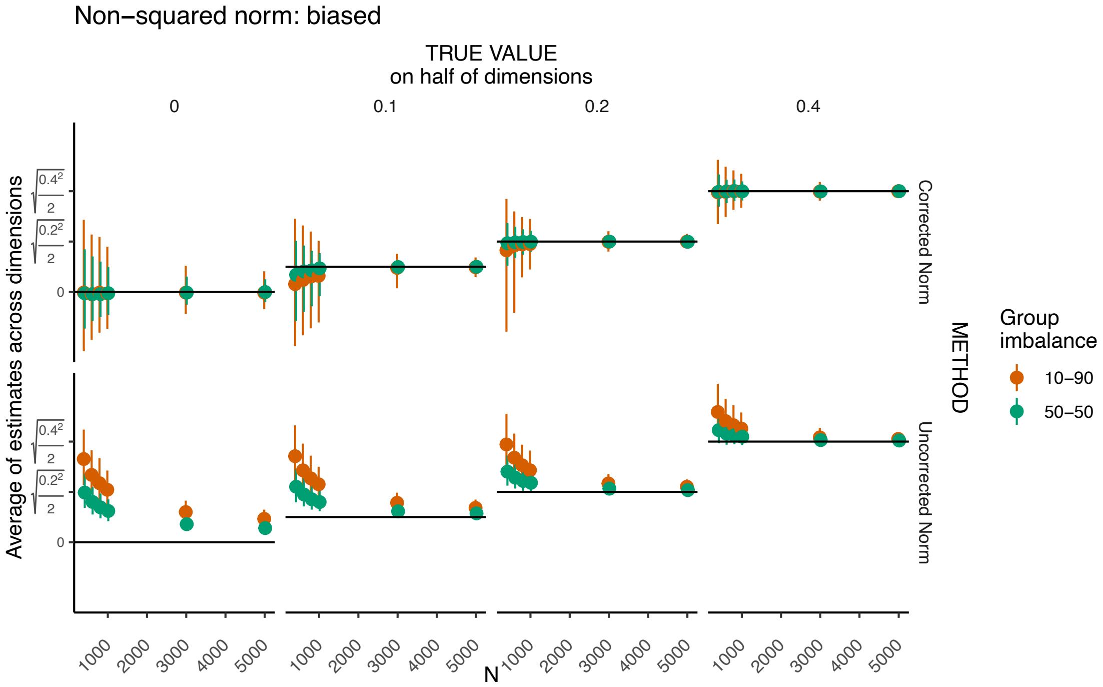

Figure C.2: This figure shows simulation results for the ordinary (unsquared) Euclidean norm. The horizontal black lines represent the true Euclidean norm, divided by the number of dimensions (50). Points represent average of the simulations and intervals are the 2.5% to 97.5% quantiles of the sampling distribution.

### C.2.1 (Unsquared) Euclidean distance bias

An expression of the bias for the Euclidean norm must, to our knowledge, be distribution dependent. For example, for the case of $k = 1$, we can use the properties of the half normal distribution (for a $\hat{\beta}$ that is normally distributed for large $N$, by the central limit theorem) to get an expression for the expected value of the absolute value of $\hat{\beta}$: $E[|\hat{\beta}|] = \sigma\sqrt{\frac{2}{\pi}}e^{\frac{-\beta^2}{2\sigma^2}} + \beta\text{erf}\left(\frac{\beta}{\sqrt{2\sigma^2}}\right)$, where erf indicates the error function and $\sigma$ the standard deviation of $\hat{\beta}$ (i.e., the standard error). $E[\hat{\beta}] = \beta$. This reduces to $E[|\hat{\beta}|] = \sigma\sqrt{\frac{2}{\pi}}$ when $\beta = 0$.

## C.3 Bimodality of corrected Euclidean distance

Unlike the squared Euclidean distance estimator, the ordinary Euclidean distance estimator is strongly bimodal. We suspect the bimodality in particular may make this estimator somewhat difficult for many readers to interpret. We illustrate this bimodality and potential interpretation problem in Figures C.3 and C.4, where we show the *same* estimates with and without squaring. In the squared version, we think that the distribution resembles what an average reader would expect to see for estimates of no difference. In the unsquared version, some readers may interpret estimates further from 0 as being more distinct from 0 than they really are – they are far from 0 only because the distribution of this estimator has low density close to zero.

Given this, and while we think it is reasonable to prefer the ordinary Euclidean distance, authors who use this corrected distance measure may need to be careful to fully explain its bimodal distribution – and the reason for heavily skewed confidence intervals.

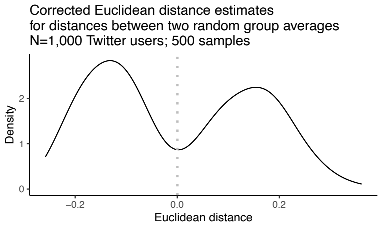

Figure C.3: Distribution of corrected Euclidean distance estimates for N=1,000 across 500 samples from Twitter data for term ‘people’.

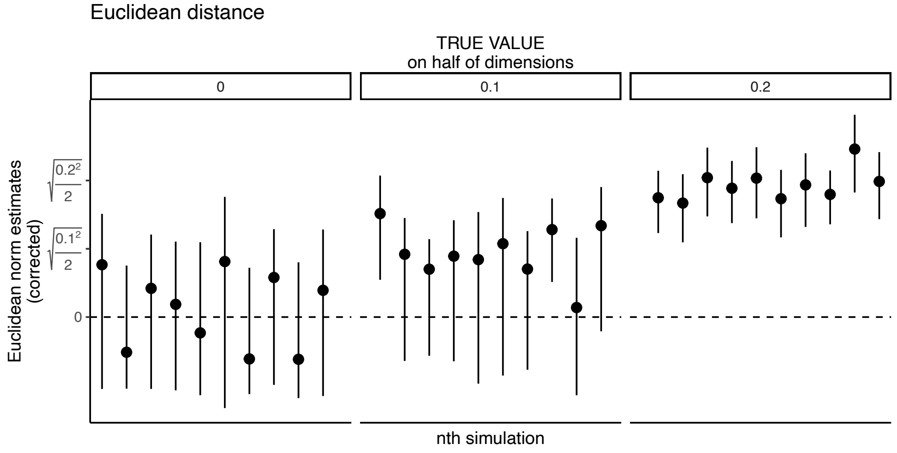

Figure C.4: 10 corrected Euclidean distance estimates for N=1,000, equal group comparisons, and different effect sizes – from simulations shown in Figure C.2.

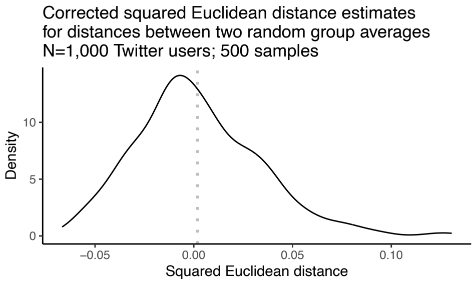

Figure C.5: Distribution of corrected *squared* Euclidean distance estimates for N=1,000 across 500 samples from Twitter data for term ‘people’.

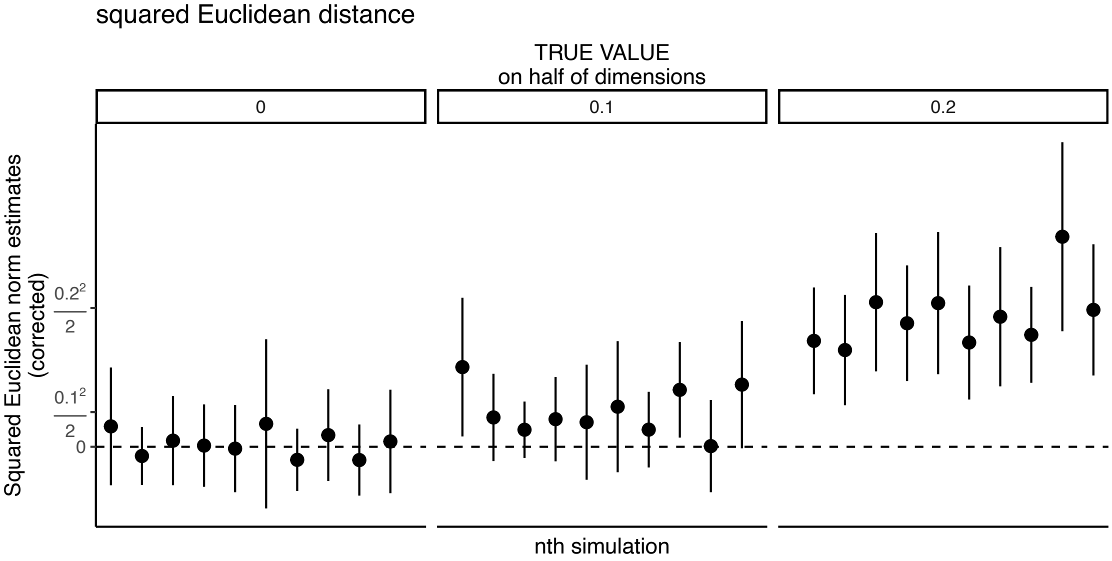

Figure C.6: 10 corrected squared Euclidean distance estimates for N=1,000, equal group comparisons, and different effect sizes – from simulations shown in Figure 2.

## C.4 Bootstrapping: challenges, coverage of confidence intervals

We assess whether bootstrapping and/or the jackknife can be used to construct confidence intervals for the squared Euclidean norm.

For the bootstrap, we calculate the coverage of a bootstrapped confidence interval with 500 replicates for our main simulations described in the main text (see Figure 2) for the case of N=1,000. Meaning, we calculate the fraction of (true) squared Euclidean norms that fall within the range of 2.5% to 97.5% quantiles of the bootstrap distribution (after subtracting double the calculated variance of each estimate – since, as we show in Figure C.7, the mean of the bootstrap distribution is biased by double the variance).

For the jackknife, we use the leave-one-out method to construct standard errors and confidence intervals.

These results are shown in Figures C.8 and C.9. For effect size values less than around 0.5, "95%" confidence intervals contain more than 95% of the true/assigned effect size. The jackknife appears to have closer to nominal coverage because for effect sizes less than 0.1 it has coverage of around 98% for a "95%" confidence interval while the bootstrap is around 100%.

In Figure C.10, we show similar coverage for the method in Hyodo et al. (2018).

We also test the jackknife using Congressional Record data from Sessions 111-114 (Gentzkow et al., 2018). To do this, we select target words with varying degrees of gender and partisan differences and obtain locally trained embeddings with context window size six. We fit an embedding regression with party or gender as a covariate and define the (non-deflated) squared Euclidean norm of the coefficients as the true parameter. We simulate sampling distributions from this 'population' of embeddings by taking sub-samples of varying sizes ($n = 100, n = 500, n = 1000$) and estimate the same regression, using the jackknife to calculate confidence intervals. For each target word and sub-sample size, we replicate the simulation process 1000 times and calculate the jackknife coverage as described above. Coverage results for each embedding regression specification are shown in Table C.1. Similar to the coverage we obtain using simulated data, the jackknife has a coverage of around 98% for a "95%" confidence interval for effect sizes close to 0, but has closer to nominal coverage for larger effect sizes.

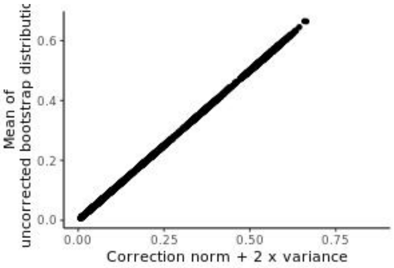

Figure C.7: Bootstrapping doubles the variance bias (from simulations in main text Figure 2).

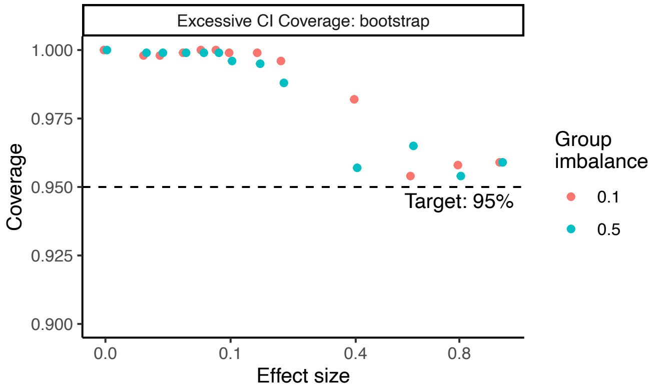

Figure C.8: Coverage of bootstrapped and doubly corrected norm for N=1,000.

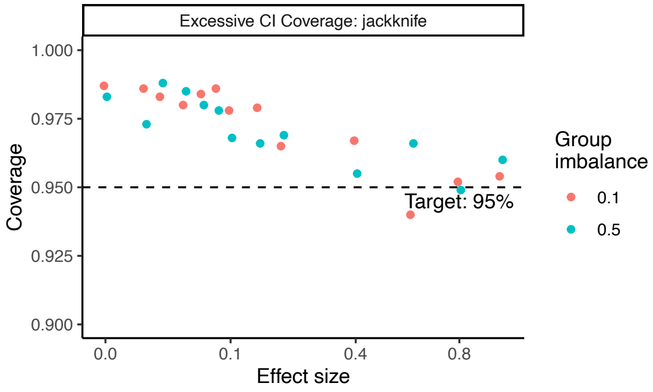

Figure C.9: Coverage of jackknife for N=1,000.

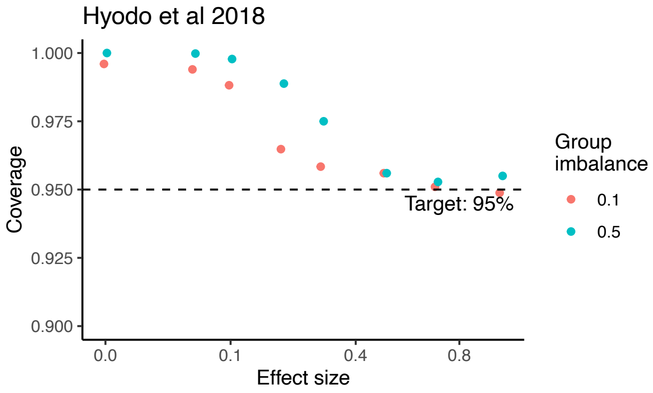

Figure C.10: Coverage of Hyodo et al. (2018) confidence intervals for N=1,000.

| Embedding Regression | Squared Norm | Total observations | Coverage by Sub-sample Size | | |
|---|---|---|---|---|---|
| | Full Sample Estimate | | n = 100 | n = 500 | n = 1000 |
| children∼gender | 0.62 | 50,191 | 0.989 | 0.982 | 0.969 |
| nation∼party | 0.82 | 49,777 | 0.986 | 0.977 | 0.968 |
| president∼party | 3.08 | 220,944 | 0.975 | 0.938 | 0.928 |
| health∼party | 3.16 | 13,3797 | 0.976 | 0.953 | 0.964 |
| women∼gender | 5.01 | 46,802 | 0.938 | 0.936 | 0.944 |
| abortion∼party | 6.57 | 6,670 | 0.974 | 0.947 | 0.932 |
| climate∼party | 6.98 | 12,641 | 0.939 | 0.92 | 0.937 |
| hispanic∼party | 7.45 | 1,565 | 0.959 | 0.91 | 0.871 |
| black∼party | 11.77 | 6,945 | 0.973 | 0.957 | 0.948 |
| unemployment∼party | 12.01 | 21,398 | 0.935 | 0.95 | 0.951 |
| wage∼party | 21.75 | 6,471 | 0.915 | 0.95 | 0.951 |
| gun∼party | 22.64 | 10,446 | 0.96 | 0.956 | 0.96 |
| immigrants∼party | 24.99 | 4,677 | 0.938 | 0.951 | 0.968 |

Table C.1: Coverage of jackknife on full Congressional Record data for N=100, N=500, and N=1000.

## C.5 Effects of whitening embeddings

Our method corrects bias related to the variance of an estimated $\hat{\beta}$ rather than variance in the data itself. If we equalize variance in the data, like by whitening a matrix and *then* calculating Euclidean distance (i.e., Mahalanobis distance), this re-introduces bias. Intuitively, this introduces bias because (large) differences between groups increase the variance of the data without altering the variance of an estimator. Further, whitening the embeddings of groups separately prior to comparing them would place them into different and incomparable embedding spaces.

In Figure C.11, we re-run our main simulation shown in Figure 2 but whiten the matrix prior to calculating distances. This whitening step equalizes the variance of every embedding dimension and removes covariance across embedding dimensions.

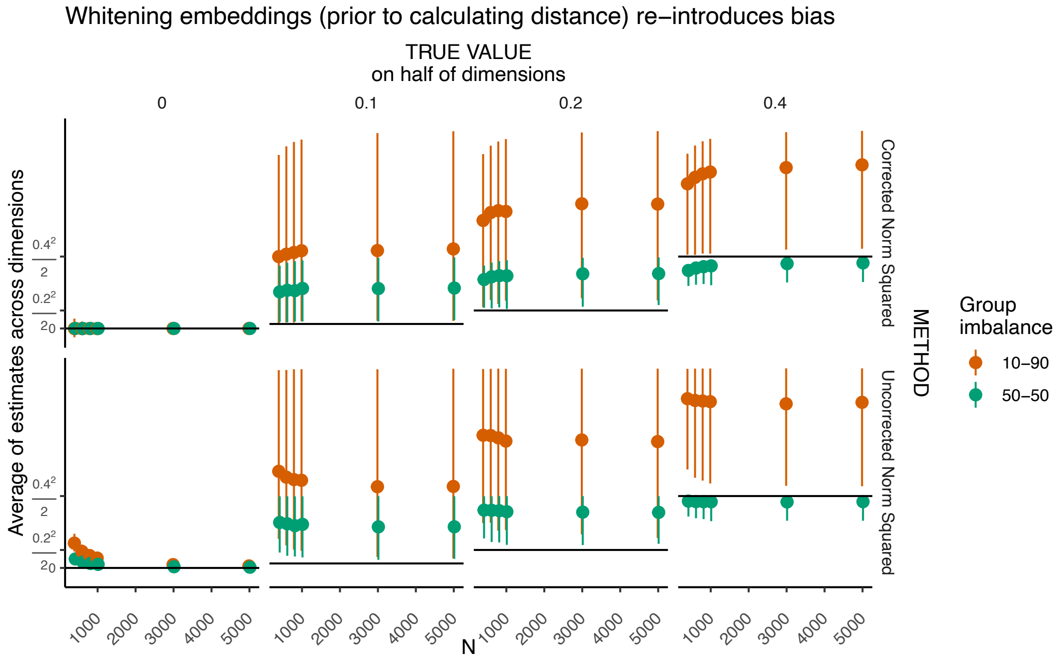

Figure C.11: Estimated squared Euclidean distance on a whitened embedding matrix. In this analysis, the simulated embeddings are whitened prior to calculating squared Euclidean distance. Whitening the embeddings of groups separately prior to comparing them would place them into different and incomparable embedding spaces.

## C.6 Covariates and clustering: corrections and simulations

### Clustering

If responses are not independent, then we can under-estimate the variance of our $\hat{\beta}$, just as in ordinary linear regression. The solution for this is straightforward – we can cluster our standard errors using standard practices. We demonstrate in Table C.2 that a) not accounting for clustering biases estimates and b) we can fix that bias through the approaches just described. For estimating clustered standard errors, we use the 'estimatr' R package (Blair et al., 2024) and "stata" (CR1) type cluster-robust standard errors.

Further, we must also permute our outcomes at the cluster level to return valid p-values. Without accounting for clustering, we will tend to over-reject the null due to a permutation distribution that is too narrow and that also has a downward bias. These problems and their fix in simulations is are shown in Tables C.4 and C.7.

Expanding our main text derivation to

$$\begin{aligned} E\left[\|\hat{\theta} - \hat{\phi}\|_2^2\right] &= \|\theta - \phi\|_2^2 + \sum_{k=1}^{K} V[\hat{\theta}_k - \hat{\phi}_k] && (5) \\ &= \|\theta - \phi\|_2^2 + \sum_{k=1}^{K} \left(V[\hat{\theta}_k] + V[\hat{\phi}_k] - 2Cov[\hat{\theta}_k, \hat{\phi}_k]\right) && (6) \end{aligned}$$

our clustered standard error approach on the *difference* corrects for both inaccurately estimated variances *and* the covariance term. This covariance term can be non-zero when, for example, the same author's text embeddings are included in the averages of both compared vectors. We demonstrate the efficacy of this covariance correction in the "non-independent contrast" results in SI Table C.4, where the same errors are included in both compared vectors in a cross-over design.

### Multiple regression

If we use a regression approach, like (Rodriguez et al., 2023), then we need to account for the possibility that highly predictive variables will reduce the variability of other estimates. In permutation tests that permute our outcomes, we remove the effect of that increased precision (setting all associations to 0 on average) and, if we do have predictive variables, then over-estimate the variance of our $\hat{\beta}$'s.

The primary solution to this issue is a) to use standard errors from the regression rather than using permutation to estimate variance and b) permuting the residuals from our regression rather than the outcome. We demonstrate that full model residual permutation produces accurate estimates and valid p-values in Tables C.3 and C.6.

Below, we conduct simulations that are the same as those in the main paper, but we restrict our sample size to 1,000, use 1,000 replicates (rather than 500), and also for:

- (maximum) clustering: duplicate each observation (each observation appears twice)
- (strong) covariate: assign a covariate with $c = 10$ (a very large effect size)
- non-independent contrasts (a crossover design): duplicate each observation – but with the duplicate observation in the opposite group as the original

In reporting estimates, we calculate the average estimates using the squared norm before taking the pseudo square root. Meaning, the estimates below are for the unbiased correction – we have only applied a pseudo square root so that we can still see bias (and lack of bias after correction) in the uncorrected squared norm for effect sizes equal to 0.

### C.6.1 Simulation estimates

| True value | Uncorrected estimate | Subtract regression variances | **Subtract clustered regression variances** |
|---|---|---|---|
| $0.00^2$ | $0.09^2$ | $0.06^2$ | $0.01^2$ |
| $0.71^2$ | $0.71^2$ | $0.71^2$ | $0.71^2$ |

Table C.2: Normed estimates: (maximum) clustering only

| True value | Uncorrected estimate | Subtract regression variances | **Subtract clustered regression variances** |
|---|---|---|---|
| $0.00^2$ | $0.09^2$ | $0.06^2$ | $0.01^2$ |
| $0.71^2$ | $0.71^2$ | $0.71^2$ | $0.71^2$ |

Table C.3: Normed estimates: (maximum) clustering and (strong) covariate

| | True value | Uncorrected estimate | Subtract regression variances | **Subtract clustered regression variances** |
|---|---|---|---|---|
| no covariate | $0.00^2$ | $0.04^2$ | $-(0.06^2)$ | $-(0.00^2)$ |
| strong covariate | $0.00^2$ | $0.04^2$ | $-(0.06^2)$ | $0.00^2$ |
| no covariate | $0.71^2$ | $0.71^2$ | $0.70^2$ | $0.71^2$ |
| strong covariate | $0.71^2$ | $0.71^2$ | $0.70^2$ | $0.71^2$ |

Table C.4: Normed estimates: (maximum) clustering and non-independent contrasts (i.e., a crossover design)

### C.6.2 Simulation p-values

| True fraction < 0.05 | Permutation test | **Clustered permutation test** | **Clustered residuals permutation test** |
|---|---|---|---|
| 0.05 | 0.73 | 0.04 | 0.04 |

Table C.5: P-values: (maximum) clustering only

| True fraction < 0.05 | Permutation test | Clustered permutation test | **Clustered residuals permutation test** |
|---|---|---|---|
| 0.05 | 0.00 | 0.00 | 0.05 |

Table C.6: P-values: (maximum) clustering only and (strong) covariate

| | True fraction < 0.05 | Permutation test | **Clustered permutation test** | **Clustered residuals permutation test** |
|---|---|---|---|---|
| no covariate | 0.05 | 0.00 | 0.05 | 0.05 |
| strong covariate | 0.05 | 0.00 | 0.06 | 0.05 |

Table C.7: P-values: (maximum) clustering only and non-independent contrasts (i.e., a crossover design)

### C.6.3 Twitter p-values

We further assessed the performance of the clustered permutations on the Twitter data using the same sampling procedure as used in Figure 3 for the term 'racism'. 'Racism' is less common than the terms 'people' and 'children' and it is also more likely to be strongly associated with covariates (which can affect the performance of permutation tests). In these tests, each tweet is weighted inversely proportional to the number of tweets that a user posted in the sample overall (e.g., each observation of a user who posted twice will receive a weight of $\frac{1}{2}$). Embeddings are permuted at the user level, whether or not a user has posted the same number of tweets, and each tweet is then re-weighted using a user's new number of tweets after permutation.

In that data, clustered permutation appropriately controls type I error and ordinary permutation slightly over-rejects the null. Clustered residual permutation slightly over-rejects, though it over-rejects less than non-clustered residual permutation. Based on this, we suspect that the ordinary permutation test may perform relatively well on most data sets – except for cases where there is substantial duplication in the embedding observations (e.g., many observations drawn from one very short document), which would more closely resemble the extreme correlation across observations considered in the simulations in Section C.6.2.

A Hotelling $T^2$ test as well as an estimator (Chen and Qin, 2010) for settings where the number of embedding dimensions exceeds the number of observations can also be used for *simple design* significance tests (Chen and Qin, 2010; Hyodo et al., 2018), though with potentially restrictive assumptions. We are unaware of any such estimator for complex designs, however, and we see below that it performs poorly with non-independent observations.

| | True fraction < 0.05 | clustered permutation | non-clustered permutation | Hotelling $T^2$ test | Hotelling $T^2$ test (on single tweet per author) |
|---|---|---|---|---|---|
| Random group: 1%-99% | 0.05 | 0.05 | 0.07 | 0.59 | 0.17 |
| Random group: 10%-90% | 0.05 | 0.05 | 0.08 | 0.94 | 0.05 |
| Random group: 50%-50% | 0.05 | 0.05 | 0.09 | 1.00 | 0.04 |

Table C.8: P-values: Twitter sampling (500 samples of 1,000 users) and distances calculated between random groups.

|  | True fraction $< 0.05$ | clustered residual permutation | non-clustered residual permutation |
|---|---|---|---|
| Random group: 10%-90% | 0.05 | 0.07 | 0.10 |
| Random group: 50%-50% | 0.05 | 0.06 | 0.09 |

Table C.9: P-values: Twitter sampling (500 samples of 1,000 users) and distances calculated between random groups. Controls: age group, race, gender, party.

## C.7 Cosine similarity correction

We use the same down-sampling procedure as in Section C.6.3 (1,000 users' uses of the word 'racism' on Twitter) to analyze the performance of a corrected cosine similarity estimator, but without covariates. In this, we corrected the Euclidean distance in the denominator of the cosine similarity calculations (the Euclidean norm of each group's average embedding vector), and left the numerator untouched (assuming independence across the compared groups). Figure C.12 and C.13 display these results. Although there is a small upward bias in the corrected estimator, that bias is far smaller than the uncorrected estimator bias. Note, too, that this figure illustrates that group differences in embeddings surrounding the same terms may tend to be relatively small.

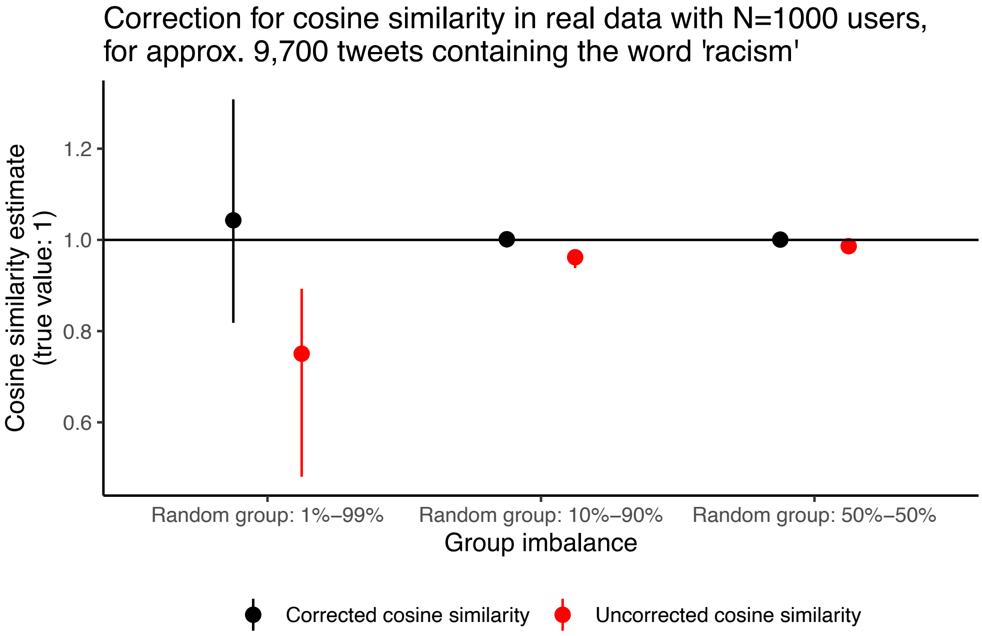

Figure C.12: Cosine similarity estimator performance on sub-samples of Twitter data set: random groups. For this data, we use sandwich-style standard errors to estimate the variance under clustering, to account for clustering at the user level given multiple tweets from the same users.

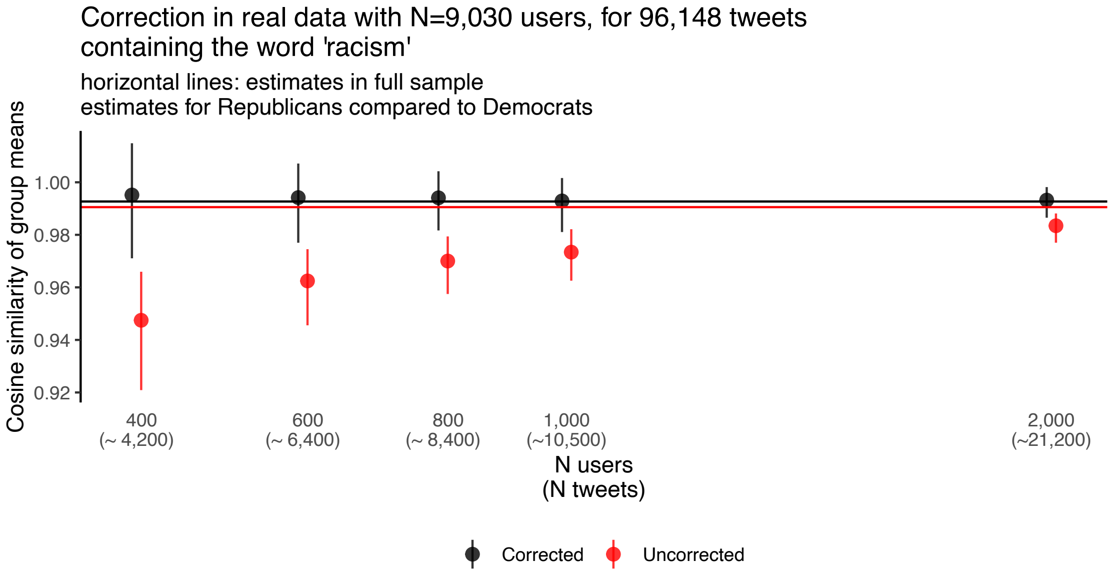

Figure C.13: Cosine similarity estimator performance on sub-samples of Twitter data set: Republican average versus Democrat average. For this data, we use sandwich-style standard errors to estimate the variance under clustering, to account for clustering at the user level given multiple tweets from the same users.

# References

Blair, G., J. Cooper, A. Coppock, M. Humphreys, and L. Sonnet (2024). *Estimatr: Fast Estimators for Design-Based Inference.*

Chen, S. X. and Y.-L. Qin (2010, April). A two-sample test for high-dimensional data with applications to gene-set testing. *The Annals of Statistics 38*(2).

Gentzkow, M., J. Shapiro, and M. Taddy (2018). Congressional record for the 43rd-114th congresses: Parsed speeches and phrase counts. *Stanford Libraries*.

Hughes, A. G., S. D. McCabe, W. R. Hobbs, E. Remy, S. Shah, and D. M. J. Lazer (2021, September). Using Administrative Records and Survey Data to Construct Samples of Tweeters and Tweets. *Public Opinion Quarterly 85*(S1), 323–346.

Hyodo, M., H. Watanabe, and T. Seo (2018, November). On simultaneous confidence interval estimation for the difference of paired mean vectors in high-dimensional settings. *Journal of Multivariate Analysis 168*, 160–173.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global Vectors for Word Representation. *EMNLP 14*, 1532–1543.

Rodriguez, P. L., A. Spirling, and B. M. Stewart (2023, January). Embedding Regression: Models for Context-Specific Description and Inference. *American Political Science Review*, 1–20.