Assessing individual response to training in sport and exercise. A conceptual and statistical review.

Swinton, P.A.

Doi: 10.51224/SRXIV.288

SportRχiv hosted preprint version 1

24/04/2023

Contact details

Dr. Paul Swinton

School of Health Sciences, Robert Gordon University

Garthdee Road

Aberdeen, UK,

AB10 7QG

p.swinton@rgu.ac.uk, +44 (0) 1224 262 3361

Twitter Handle:

@PaulSwinton9

Please cite as: Swinton, PA. Assessing individual response to training in sport and exercise. A conceptual and statistical review. 2023. Pre-print available from SportRχiv. https://doi.org/10.51224/SRXIV.288.

Abstract

Researchers are increasingly exploring contexts where training causes meaningful differences in the changes experienced by participants across interventions. Where this occurs, the phenomenon is referred to as individual response or trainability and provides scope for personalising training to maximise improvements based on participant characteristics. The potential for training to cause individual response in a given population is commonly assessed by comparing the variability in observed change between an intervention and control group. Similarly, the most common statistic used to quantify the difference is the standard deviation of individual response (SD_{IR}). It has been recommended that preliminary studies estimate the SD_{IR} to identify areas where personalising training may provide substantive improvements over prescribing the same, usually standardised, training to all participants. The purpose of this review was to provide a detailed examination of the SD_{IR} including conceptual and statistical overviews. A series of different plausible data generating models were used to highlight where the SD_{IR} appropriately assesses individual response, and where the standard formulation may lead to erroneous conclusions. The review highlights the importance of expressing uncertainty in estimates, comparing three different approaches to creating confidence intervals. It is recommended that 'melded' confidence intervals be used, especially for studies investigating relatively small sample sizes. The review also shows how model misspecification in terms of different measurement error distributions between intervention and control, and variance heterogeneity in external factors may represent the most pressing threats to valid conclusions when estimating the SD_{IR}. It is recommended that future research assess the potential for model misspecification and variance heterogeneity. Repeated measurements pre- and post-training can be used to better estimate the SD_{IR} and account for differences in group measurement error. The existence of variance heterogeneity should be relatively simple to identify, however, it will be important for research teams to consider the best measures to capture the wide range of external factors that may influence observed change in outcomes included pre- and post-training.

Introduction

Concurrent with attempts to identify training interventions that are most effective for different populations, there is growing interest in personalised approaches to exercise prescription. The belief that personalised approaches will be more effective is based primarily on the assumption that within a specific population, there are participant characteristics that interact with the training stimulus causing effect modification (Hecksteden et al, 2015; Mills et al, 2021). Where this interaction exists, the related concepts of 'individual response' and 'trainability' are evoked and those participants with positive interactions are regarded as more trainable (Hecksteden et al, 2015). Additionally, with meaningful thresholds of improvement stated, participants can be identified as responders or non-responders, or potentially more appropriately, the expected proportion of response calculated (Bonafiglia et al, 2021; Bonafiglia et al, 2022). With knowledge of the most relevant participant characteristics and how they interact with different training stimuli, the best match could theoretically be selected for each individual. There are, however, several challenges in surmounting even the first hurdle in identifying where participant-by-training interactions occur. Reliable estimates will in general require large sample sizes (Mills et al, 2021), greatly exceeding those routinely used in sport and exercise research (Swinton et al, 2023). As a result, it is recommended that preliminary investigations be conducted where individual response is likely to be large and practically relevant to justify the resources required for subsequent study estimating participant-by-training interactions (Atkinson and Batterham 2015).

Preliminary investigations of individual response generally focus on analysis of variability in observed change across an intervention using a measurement outcome that reflects the domain of interest (Bonafiglia et al, 2021). This variability is referred to as gross response variability and is comprised of several sources (Ross et al, 2019). The three main sources of variability include measurement error, variation due to differences in external factors that affect the measurement outcome independently of training (e.g. sleep, nutrition and expectation), and participant-by-training interaction. To parse these

three different sources of variation, preliminary studies often include a control group to provide a measure of the variability caused by external factors, as those in the control do not engage in the training intervention. To illustrate how participant-by-training interactions can influence gross response variability, we examine one of the most likely factors creating individual response. It is generally proposed that a participant's baseline value is inversely related to improvements such that those with higher baselines experience reduced improvements (Swinton et al, 2023). To capture this and other phenomena throughout the review, we introduce the following notation and model framework: Participants unknown true values for the outcome of interest are captured by the random variable Y_{ijk}, where i is the i-th person in the j-th group {0=control, 1=intervention}, and k is the time point {0=pre-training, 1=post-training}. A data generating model for a participant-by-training interaction from baseline value can be expressed as:

$$Y_{ij1} = Y_{ij0} + \beta_0 + \beta_1 X_{1_{.j1}} + \beta_2 X_{2_{.j1}} Y_{ij0} + \zeta_{Ext_{i.1}}, \qquad eq.1$$

where β_0 is the mean change in the control group, $X_{1_{.j1}}$ and $X_{2_{.j1}}$ are indicator variables that take on the value 0 for a participant in the control, and 1 for a participant in the intervention, β_1 is required to set the expected change of the intervention relative to control, β_2 controls the direction and magnitude of the relationship between baseline value and change, and $\zeta_{Ext_{i.1}}$ is used to describe external factors that cause variation but not through interaction with the training stimulus and in this model applies equally to both groups. $\zeta_{Ext_{i.1}}$ is modelled as a random error term of the form $N(0, \tau^2_{Ext})$ and is viewed more appropriately as a model simplification representing as yet unexplained series of relationships involving external factors and post-training outcome variability (Hecksteden 2015). In practice, we cannot obtain a participant's true value and instead when measurements are conducted, we obtain the observed random variables $y_{ijk} = Y_{ijk} + \epsilon_{ijk}$, where $\epsilon_{ijk} \sim N(0, \delta^2)$. The measurement error ϵ_{ijk} includes instrumentation noise and short term biological variability (Swinton et al, 2018). In this initial model, the standard deviation of the measurement error (δ) is the same across both groups and time points.

Where there is as generally posited, a negative participant-by-training interaction such that those with the highest baseline values experience lower improvements (β_2<0), there is a relative fanning in effect such that the post-intervention variation is lower than would be obtained if there was no interaction (Supplementary B1). Based on this finding, it may be intuited that the variation in change values across the training is reduced for the intervention group. In contrast, the opposite occurs, and the participant-by-training interaction causes an increase in variation of change values (see Supplementary B1 for explanation), such that for the model presented in eq.1, the intervention group exhibits greater gross response variability compared with control. It can be shown that the sign of the relationship has no influence, only the magnitude of β_2 with greater values causing increased variability in observed change of the intervention group (Supplementary B1).

The comparison of variances in change values across training between an intervention and control group has commonly been made using what has been referred to as the standard deviation of individual response (SD_{IR}) (Hopkins 2015). As the number of studies using the SD_{IR} to identify training and population combinations that may exhibit participant-by-training interactions has increased, mixed findings have been generated with some studies reporting potentially large individual response, and others reporting negative SD_{IR} values or individual response that is unlikely to be clinically meaningful (Bonafiglia et al, 2021). Whilst these mixed findings may reflect the true underlying existence or not of participant-by-training interactions across different contexts, it is important to consider the SD_{IR} in detail. This includes how to estimate and interpret the SD_{IR}, and contexts where estimates may lead to faulty conclusions. The present review provides a detailed overview of the SD_{IR} and three areas of associated importance including: 1) expressing uncertainty in SD_{IR} estimates; 2) the influence of model misspecification; and 3) the influence of variance heterogeneity in external factors.

Introduction to the SD_{IR}

First, we introduce the canonical model used to conceive the SD_{IR} and subsequent potential to identify the existence of individual response:

$$Y_{ij1} = Y_{ij0} + \beta_0 + \beta_1 X_{1\_j_1} + \zeta_{Train_{i11}} + \zeta_{Ext_{i.1}}. \qquad eq.2$$

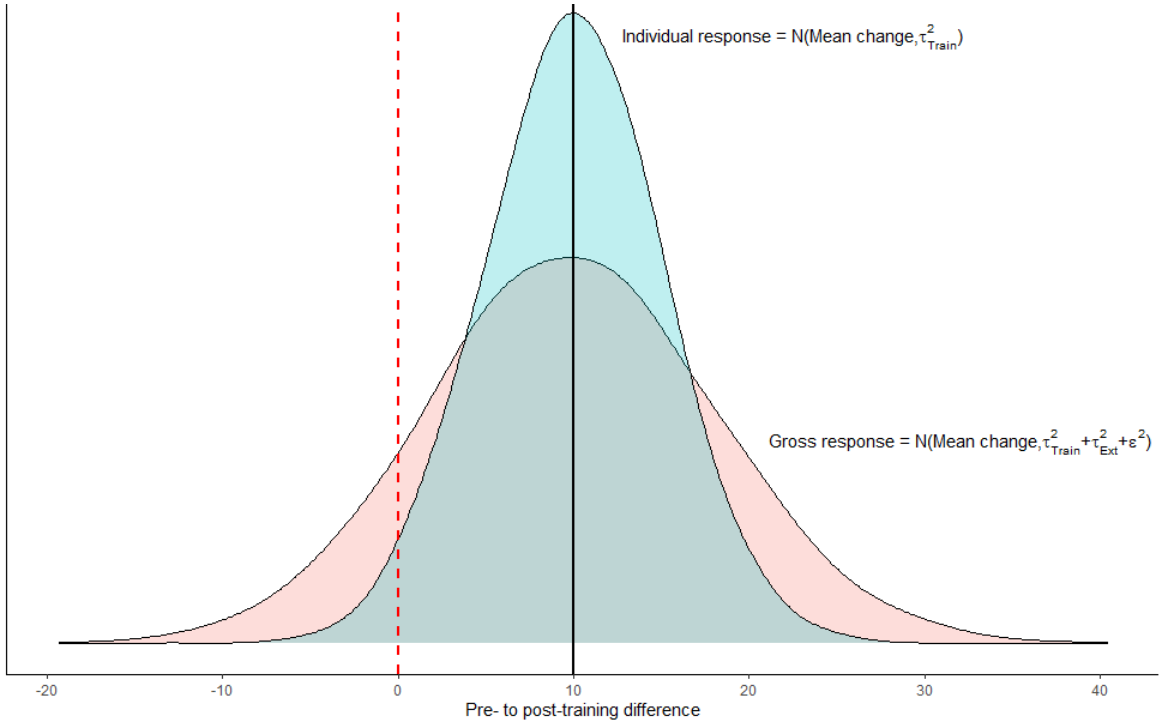
Here we follow the same format as eq.1, except we model the interaction of participant baseline values (and any other potential participant-by-training interactions) with the random effect $\zeta_{Train_{i11}}$, where $\zeta_{Train_{i11}} \sim N(0, \tau^2_{Train})$. The subscripts denote that participants in both groups are exposed to the same distribution of the random effect term ζ_{Ext} describing the influence of factors that act by mechanisms external to the training stimulus, but only the intervention group is exposed to ζ_{Train}. Note, with this model it is possible that a factor such as age could contribute to ζ_{Train} by interacting with the training stimulus, and to ζ_{Ext} through other mechanisms.

The SD_{IR} is defined as the standard deviation of the variance in observed change values of the intervention group minus the variance in observed change values of the control:

$$SD_{IR} = \sqrt{\mathrm{Var}(y_{.11} - y_{.10}) - \mathrm{Var}(y_{.01} - y_{.00})} = \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.})}, \qquad eq.3$$

where Δ denotes the change in observed values across the training. As shown in Supplementary B2, the SD_{IR} based on the data generating model in eq2 is equal to τ_{Train}, which was the intended target. Combining the expected change across the intervention and τ_{Train} provides a quantitative description of individual response through a normal distribution, with the proportion of the curve exceeding a given threshold denoting the proportion of response (Figure 1).

Figure 1: Schematic illustrating individual and gross response based on pre- to post-training change values.



Broader distribution represents the gross response distribution obtained by creating a normal distribution from the mean and standard deviation of the observed change values from the intervention group. The narrower distribution represents the individual response distribution obtained by creating a normal distribution from the mean observed change from the intervention group and the SD_{IR} calculated from observed change from both intervention and control. Red line represents a simple zero threshold, such that the proportion of the individual response distribution that lies beyond the line equals the proportion of response. Alternative thresholds can be selected.

Uncertainty in estimating the SD_{IR}

In practice, the SD_{IR} is unknown and requires estimation with a sample statistic. This estimate is denoted $\widehat{SD}_{IR}$ and calculated using sample variances:

$$\widehat{SD}_{IR} = \sqrt{S^2_{\Delta_1} - S^2_{\Delta_0}}, \qquad eq.4$$

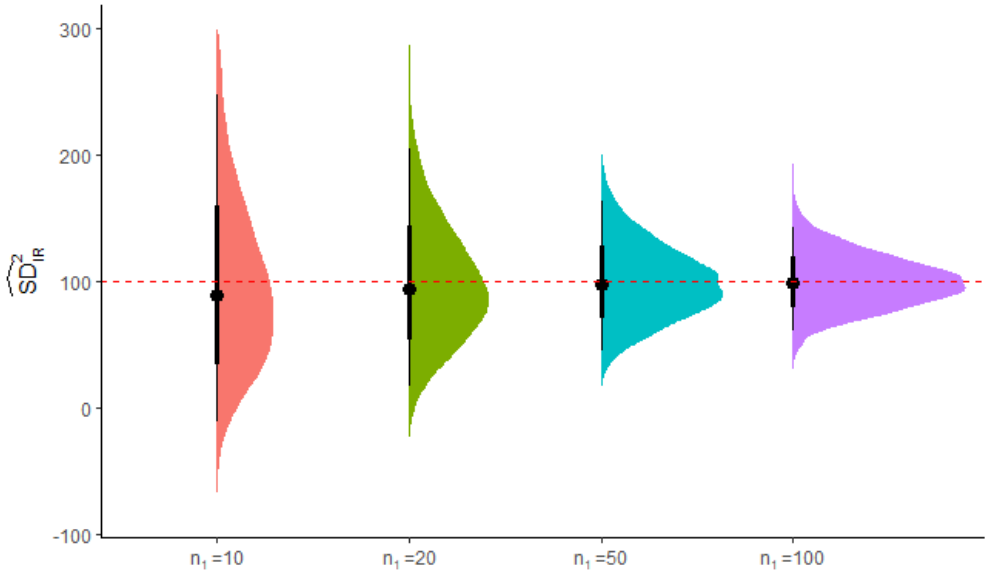
where $S^2_{\Delta_j} = \frac{1}{n_j - 1}\sum_{r=1}^{n_j}\left(\Delta_{rj.} - \bar{\Delta}_{.j.}\right)^2$. The statistic $\widehat{SD}_{IR}$ comprises sampling variance, potentially with an easily described distribution depending on the data generating model. Where the estimate is made with relatively small sample sizes, it is plausible that the value calculated will be far from the actual SD_{IR}. It is possible therefore, that when a study does not obtain an $\widehat{SD}_{IR}$ deemed indicative of meaningful participant-by-training interaction (or even positive), that the phenomena does occur (with the opposing

argument also a possibility). Within a frequentist framework, the typical approach to express uncertainty is to calculate a confidence interval for the parameter of interest. Currently, two popular methods are used to calculate confidence intervals for the SD_{IR} (Hopkins 2015, Hecksteden et al, 2018). Both methods start by creating a confidence interval for SD_{IR}^2 based on an assumed distribution for the estimate, and then square the limits. The first and most common method (Hopkins 2015) assumes a normal distribution for $\widehat{SD}_{IR}^2$ and estimates the standard error (see Supplementary B3) to provide the following (100-α)% confidence interval:

$$\widehat{SD}_{IR}^2 + F_z^{-1}(\alpha/2)\sqrt{2\left(\frac{S_{\Delta_0}^4}{n_0-1}+\frac{S_{\Delta_1}^4}{n_1-1}\right)} < SD_{IR}^2 < \widehat{SD}_{IR}^2 + F_z^{-1}(1-\alpha/2)\sqrt{2\left(\frac{S_{\Delta_0}^4}{n_0-1}+\frac{S_{\Delta_1}^4}{n_1-1}\right)}, \qquad eq.5$$

where F_z^{-1} is the inverse of the standard normal cumulative distribution function. As show in Figure 2, the distribution for $\widehat{SD}_{IR}^2$ does approach a normal distribution with increasing sample size, but for small samples there can be substantial asymmetry.

Figure 2: Sampling variance of estimate $\widehat{SD}_{IR}^2$ obtained with Monte Carlo simulation across increasing sample sizes with $n_0 = n_1$.



Red line represents true parameter value for $SD_{IR}^2 = \tau_{Train}^2$. Values used for simulations: $Y_{ij0} \sim N(100, 15^2)$, $\beta_0 = 5, \beta_1 = 15, \tau_{Ext}^2 = 36, \delta^2 = 4$; Number of iterations per simulation = 10,0000.

Given the limitation of assuming a normal distribution, Hecksteden et al (2017) suggested the use of a chi-squared distribution for $\widehat{SD}_{IR}^2$ and creating a (100-α)% confidence interval with:

$$\frac{(n_1-1)\widehat{SD}_{IR}^2}{F^{-1}_{\chi^2_{n_1-1}}(1-\alpha/2)} < SD_{IR}^2 < \frac{(n_1-1)\widehat{SD}_{IR}^2}{F^{-1}_{\chi^2_{n_1-1}}(\alpha/2)}, \qquad eq.6$$

where $F^{-1}_{\chi^2_{n_1-1}}$ is the inverse of the cumulative distribution function for a chi-squared random variable with $n_1 - 1$ degrees of freedom. Using a specific data set, Hecksteden et al (2017) compared the use of a normal versus a chi-squared distribution and showed much shorter intervals with the chi-squared approach. However, whilst $(n_j - 1)S_{\Delta_j}^2/\sigma_j^2$ for $j = 0,1$ both follow a chi-squared distribution (Cochran 1934), the difference does not, with the chi-squared distribution defined only for positive values. The chi-squared distribution is a special case of the gamma distribution (Supplementary B3), and the difference between two chi-squared random variables is described by a VarianceGamma distribution when the degrees of freedom are equal (Ferrari 2019), and a gamma difference distribution when they are not (Klar 2015) (Supplementary B3). Instead, a relatively simple and elegant method to create a confidence interval for SD_{IR}^2 is to 'meld' the sample confidence intervals with $S_{\Delta_1}^2$ and $S_{\Delta_0}^2$ (see Supplementary B3). The method presented by Fay et al. (2015) guarantees nominal coverage and the melded $100(1-\alpha)\%$ lower ($L_{SD_{IR}^2}(\alpha)$) and upper ($U_{SD_{IR}^2}(\alpha)$) one-sided melded confidence limits for SD_{IR}^2 are obtained with:

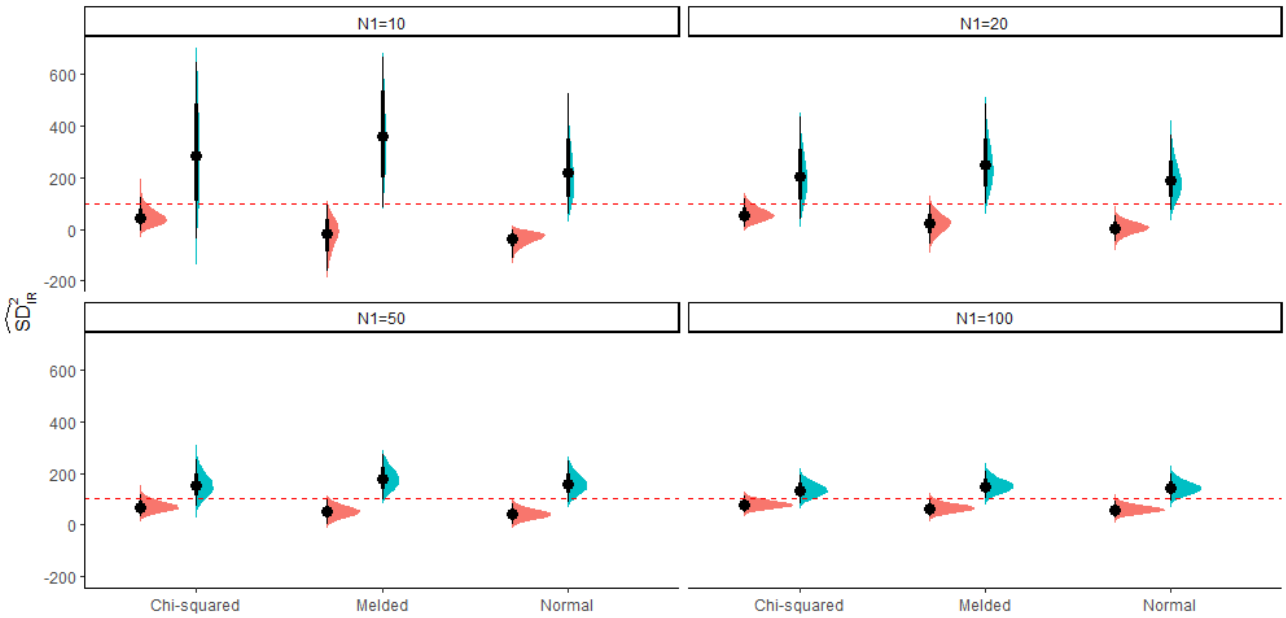
$L_{SD_{IR}^2}(\alpha) =$ the α-th quantile of $(n_1 - 1)S_{\Delta_1}^2/F^{-1}_{\chi^2_{n_1-1}}(A) - (n_0 - 1)S_{\Delta_0}^2/F^{-1}_{\chi^2_{n_0-1}}(B)$

$U_{SD_{IR}^2}(\alpha) =$ the $(1-\alpha)$-th quantile of $(n_1 - 1)S_{\Delta_1}^2/F^{-1}_{\chi^2_{n_1-1}}(A) - (n_0 - 1)S_{\Delta_0}^2/F^{-1}_{\chi^2_{n_0-1}}(B).$ $\qquad eq.7$

where A and B are uniform random variables, such that the melded interval can be calculated using Monte Carlo simulation or numeric integration (see Supplementary B3 for more detail).

In Figure 3 we illustrate the performance of 95% confidence intervals generated using the three different approaches (normal distribution, chi-squared distribution, and melded interval) by simulating data for different sample sizes and calculating the percentage of times the intervals included the true value. For all sample sizes the melded confidence intervals included the true value on 95% of occasions, and as can be seen in Figure 3, both upper and lower bounds were close to the true value. Performance improved for greater sample sizes when assuming a normal distribution (n=10: 89%; n=20: 92%; n=50: 94%; n=100: 95%), with the lower bound moving closer to the true value as the sample size increased (Figure 3). Finally, performance of confidence intervals assuming a chi-squared distribution remained poor as sample sizes increased (n=10: 78%; n=20: 80%; n=50: 80%; n=100: 81%), with too much overlap of both upper and lower bounds (Figure 3).

Figure 3: Assessment of 95% confidence interval performance using the normal distribution assumption, chi-squared distribution assumption, and melded intervals across increasing sample sizes with $n_0 = n_1$.



Red distributions represent distribution of lower bound limits. Blue distributions represent distribution of upper bound limits. Red line represents true parameter value for τ^2_{Train}. Values used for simulations: $Y_{ij0} \sim N(100, 15^2)$, $\beta_0 = 5, \beta_1 = 15, \tau^2_{Ext} = 36, \delta^2 = 4$, number of iterations per simulation = 10,0000. Number of uniform random variables for melded interval = 1000.

Model misspecification

Previous authors have highlighted that a key assumption in the use of the SD_{IR} to assess individual response is that measurement errors are normally distributed, and perhaps most importantly, equal for both groups across the training (Ross et al, 2019; Mills et al, 2021). Research in sport and exercise shows that measurement errors are often heteroscedastic, such that those that produce the highest values exhibit greater variability and therefore larger measurement errors (Atkinson and Nevil 1998). As research shows that even short training intervention can results in relatively large average improvements (Swinton et al, 2022), it is possible that measurement errors could increase for the intervention group across training. In contrast, in many training interventions measurement outcomes are the same, or similar to, the activities performed in training. Additionally, many training interventions are performed in the same laboratory setting with the same researchers that conduct the measurement outcomes. The increased familiarity and reduced learning effects that these conditions are likely to induce in the intervention group compared with the control may result in greater consistency and therefore reduced measurement error following training. Model misspecification in terms of different group measurement error distributions post-training would cause the data generating model in eq.2 to be updated such that $\epsilon_{ij1} \sim N(0, \delta^2)$ is replaced with $\epsilon_{ij1} \sim N(0, \delta_{.j1}^2)$. The potential for this change to alter findings will depend primarily on the relative variances caused by measurement error and $\zeta_{Train_{i.1}}$. As shown in Supplementary B4, with this model misspecification, the original definition of the SD_{IR} leads to $SD_{IR} = \sqrt{\tau_{Train}^2 + \delta_{.11}^2 - \delta_{.01}^2}$, such that the quantity will be inflated if measurement error is greater in the intervention compared with control post-training, and reduced if measurement error is lower.

To address the model misspecification identified here, additional post-training measurements are required to estimate the differences in measurement error caused by the experimental process. We begin with the simple case where in addition to the pre- and post-training data, we conduct two separate measurements

post-training (y_{ij1_A}, y_{ij1_B}) to estimate reliability. Under these conditions we update the SD_{IR} to $\widetilde{SD}_{IR}$ and use the difference of these additional post-training values for the intervention ($\Delta_{i11_{AB}}$) and control ($\Delta_{i01_{AB}}$):

$$\widetilde{SD}_{IR} = \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.}) + \frac{1}{2}\left(\mathrm{Var}\left(\Delta_{i01_{AB}}\right) - \mathrm{Var}\left(\Delta_{i11_{AB}}\right)\right)}. \qquad eq.8$$

As shown in Supplementary B4, this updated quantity returns τ_{Train} under the model misspecification identified. As was done previously, we can estimate this quantity ($\widehat{\widetilde{SD}}_{IR}$) and create confidence intervals for the estimate. Provided in Supplementary B4 are calculations using sample standard deviations including the difference in the additional post-training values to create confidence intervals assuming a normal distribution for $\widehat{\widetilde{SD}}_{IR}^2$. Performance of the 95% confidence intervals was improved for the updated process (n=10: 92%; n=20: 94%; n=50: 95%; n=100: 95%) compared with the original (n=10: 89%; n=20: 91%; n=50: 90%; n=100: 89%), highlighting that under model misspecification the original process would not converge on the correct proportion for a (100-α)% confidence interval.

In practice, where researchers expect model misspecification and wish to account for differences in measurement reliability, it is likely that a research design including two sets of measurements pre- and post-training would be implemented. The average of the two pre-and post-training measurements can be used to reduce uncertainty and width of confidence intervals (Swinton et al, 2023), and the difference between the post-training measurements used to adjust the SD_{IR} estimate for changes in measurement error. Note, with this design an update is required to both the estimator: $\widehat{\widetilde{SD}}_{IR} = \sqrt{S^2_{\bar{\Delta}_1} - S^2_{\bar{\Delta}_0} + \frac{1}{4}\left(S^2_{\Delta_{0_{AB}}} - S^2_{\Delta_{1_{AB}}}\right)}$ and standard error: $\sqrt{2\left(\frac{S^4_{\bar{\Delta}_0}}{n_0-1} + \frac{S^4_{\bar{\Delta}_1}}{n_1-1} + \frac{S^4_{\Delta_{0_{AB}}}}{16(n_0-1)} + \frac{S^4_{\Delta_{1_{AB}}}}{16(n_1-1)}\right)}$, (Supplementary B4) to obtain appropriate confidence intervals.

Variance heterogeneity in external factors

Another key assumption in the use of the SD_{IR} to assess the potential for individual response is that the variance in external factors that influence change remains consistent for both groups across training (Ross et al, 2019). One of the most studied outcomes in research investigating individual response is body mass (Williamson et al, 2018; Bonafiglia et al, 2022). Here, nutrition plays a key role in influencing change and whilst the effect of nutrition may be the same for both intervention and control, it is possible that engaging in exercise alters the variation in whichever outcome is used to summarise nutritional factors, leading to variance heterogeneity. To explore this potential further, we introduce the final data generating model:

$$Y_{ij1} = Y_{ij0} + \beta_0 + \beta_1 X_{1_{.j1}} + \beta_2 X_{2_{ij1}} + \zeta_{Train_{i11}} + \zeta_{Ext_{i.1}}. \qquad eq.9$$

where $X_{2_{ij1}}$ is a covariate measuring an external factor such as nutritional intake, that alters the post-training value. β_2 quantifies the magnitude and direction of the effect and is assumed to be constant across individuals and groups. In this variance heterogeneity model, we have $X_{2_{ij1}} \sim N\left(\psi_{2.j1}, \tau_{2.j1}\right)$, where it is possible that both the mean, but importantly, the variance of the covariate is different across groups. Note, that $\zeta_{Ext_{i.1}}$ is different from the quantity expressed in eq.2, as at least some of the influence of nutrition has been accounted for in $\beta_2 X_{2_{ij1}}$, so we may expect τ^2_{Ext} to decrease. We may also expect that $\tau_{2.01} > \tau_{2.11}$, such that the SD_{IR} from eq.3 would not return τ_{Train}, but instead return $\sqrt{\tau^2_{Train} + \beta_2^2\left(\tau^2_{2.11} - \tau^2_{2.01}\right)}$ (Supplementary B5), and so underestimate individual response.

Extensive study is required to systematically investigate different external factors and how they combine with participant-by-training interaction terms to explain observed variation across different contexts and populations. Extensive study includes development of valid and reliable measurements that can summarise the aggregate effects of external factors on individuals across training so that they can be

included in models such as that presented in eq.9. In the meantime, however, there are sufficient measurement scales regarding a range of external factors such as sleep quality, nutritional intake, and life stress, to assess whether variance heterogeneity is common, and incorporate findings into future study of individual response.

Conclusion

Investigating individual response is a challenging area of research given typical constraints including the use of relatively small sample sizes and short training durations (Swinton et al, 2022). Calculation of the SD_{IR} and use of meta-analyses to combine estimates across different contexts in sport and exercise (Steele et al, 2022) represent important steps to provide an overall assessment of the scope of individual response prior to investing substantial resources focusing on specific areas. The purpose of this review was to provide a detailed examination of the SD_{IR} dealing with important topics such as expressing uncertainty and identifying issues that may cause interpretations to be erroneous. When conducting research with relatively small sample sizes, it is recommended that melded confidence intervals be used to express uncertainty in the SD_{IR}. It is also recommended that future research assess the potential for model misspecification and variance heterogeneity. Repeated measurements pre- and post-training can be used to estimate the SD_{IR} and account for model misspecification including differences in group measurement error. It is likely that multidisciplinary teams will be required to identify relevant external factors that influence chosen outcomes including the most appropriate, valid, and reliable measuring tools. With selection of appropriate external factors and measurement tools, variance heterogeneity can easily be assessed between those in the intervention and control. Where substantive variance heterogeneity exists, this should be accounted for when assessing individual response.

References

Atkinson G, Nevill AM. Statistical methods for assessing measurement error (reliability) in variables relevant to sports medicine. Sports medicine. 1998;26:217-38. Doi:10.2165/00007256-199826040-00002.

Atkinson G, Batterham AM. True and false interindividual differences in the physiological response to an intervention. Experimental physiology. 2015 Jun 1;100(6):577-88. Doi:10.1113/EP085070.

Bonafiglia JT, Preobrazenski N, Gurd BJ. A systematic review examining the approaches used to estimate interindividual differences in trainability and classify individual responses to exercise training. Frontiers in Physiology. 2021:1881. Doi: 10.3389/fphys.2021.665044.

Bonafiglia JT, Swinton PA, Ross R, Johannsen NM, Martin CK, Church TS, Slentz CA, Ross LM, Kraus WE, Walsh JJ, Kenny GP. Interindividual Differences in Trainability and Moderators of Cardiorespiratory Fitness, Waist Circumference, and Body Mass Responses: A Large-Scale Individual Participant Data Meta-analysis. Sports Medicine. 2022;52(12):2837-51. Doi: 10.1007/s40279-022-01725-9.

Cochran WG. The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. Mathematical Proceedings of the Cambridge Philosophical Society. 1934;30(2):178-191. Doi:10.1017/S0305004100016595.

Fay MP, Proschan MA, Brittain E. Combining one-sample confidence procedures for inference in the two-sample case. Biometrics. 2015;71(1):146-56. Doi: 10.1111/biom.12231.

Ferrari A. A note on sum and difference of correlated chi-squared variables. 2019. Pre-print available from arXiv. Doi:10.48550/arXiv.1906.09982.

Hecksteden A, Kraushaar J, Scharhag-Rosenberger F, Theisen D, Senn S, Meyer T. Individual response to exercise training-a statistical perspective. Journal of applied physiology. 2015;118(12):1450-9. Doi: 10.1152/japplphysiol.00714.2014.

Hecksteden A, Pitsch W, Rosenberger F, Meyer T. Repeated testing for the assessment of individual response to exercise training. Journal of Applied Physiology. 2018;124(6):1567-79. Doi: 10.1152/japplphysiol.00896.2017.

Hopkins WG. Individual responses made easy. Journal of applied physiology. 2015;118(12):1444-6. Doi:10.1152/japplphysiol.00098.2015.

Klar B. A note on gamma difference distributions. Journal of Statistical Computation and Simulation. 2015;85(18):3708-15. Doi:10.1080/00949655.2014.996566.

Mills HL, Higgins JP, Morris RW, Kessler D, Heron J, Wiles N, Smith GD, Tilling K. Detecting heterogeneity of intervention effects using analysis and meta-analysis of differences in variance between trial arms. Epidemiology. 2021;32(6):846. Doi: 10.1097/EDE.0000000000001401.

Ross R, Goodpaster BH, Koch LG, Sarzynski MA, Kohrt WM, Johannsen NM, Skinner JS, Castro A, Irving BA, Noland RC, Sparks LM. Precision exercise medicine: understanding exercise response variability. British journal of sports medicine. 2019;53(18):1141-53. Doi:10.1136/bjsports-2018-100328.

Steele J, Fisher J, Smith D, Schoenfeld B, Yang Y, Nakagawa S. Meta-Analysis of Variation in Sport and Exercise Science: Examples of Application Within Resistance Training Research. 2022. Pre-print available from SportRχiv. Doi:10.51224/SRXIV.214.

Swinton PA, Hemingway BS, Saunders B, Gualano B, Dolan E. A statistical framework to interpret individual response to intervention: paving the way for personalized nutrition and exercise prescription. Frontiers in nutrition. 2018;5:41. Doi:10.3389/fnut.2018.00041.

Swinton PA, Burgess K, Hall A, Greig L, Psyllas J, Aspe R, Maughan P, Murphy A. Interpreting magnitude of change in strength and conditioning: Effect size selection, threshold values and Bayesian updating. Journal of sports sciences. 2022;40(18):2047-54. Doi:10.1080/02640414.2022.2128548.

Swinton, PA. The influence of baseline capability on intervention effects in strength and conditioning: A review of concepts and methods with meta-analysis. 2023. Pre-print available from SportRχiv. Doi:10.51224/SRXIV.285.

Swinton P, Hemingway BS, Gallagher I, Dolan E. Statistical Methods to Reduce the Effects of Measurement Error in Sport and Exercise: A Guide for Practitioners and Applied Researchers. 2023. Pre-print available from SportRχiv. Doi:10.51224/SRXIV.247.

Williamson PJ, Atkinson G, Batterham AM. Inter-individual differences in weight change following exercise interventions: a systematic review and meta-analysis of randomized controlled trials. Obesity Reviews. 2018;19(7):960-75. Doi:10.1111/obr.12682.

Doi:10.51224/SRXIV.288 SportRχiv Preprint version 1

Quantifying interindividual variability of training in sport and exercise. A conceptual and statistical review.

Swinton, P.A.

Supplementary files:

The following supplementary file derives the results presented in the main paper and provides R code to illustrate and provide checks.

Supplementary A: Properties of statistical models

In this section basic properties of statistical models are outlined that will be used to derive subsequent results.

Property 1 (P1): Jointly Normal random variables: Two random variables X, Y are said to be jointly normal if they can be expressed in the form $X = aU + bV$; $Y = cU + dV$ where U and V are independent normal random variables.

Property 2 (P2): Population mean $E(X) = \mu$ and the linearity of expectation: $E(aX + bY) = aE(X) + bE(Y)$, where a and b are constants.

Property 3 (P3): Expectation of an independent product: if X and Y are independent then $E(XY) = E(X)E(Y)$.

Property 4 (P4): Population variance and expectation: $\mathrm{Var}(X) = E(X^2) - \mu^2$.

Property 5 (P5): Variance of a linear combination: $\mathrm{Var}(aX + by) = a^2\mathrm{Var}(X) + 2ab\mathrm{Cov}(X, Y) + b^2\mathrm{Var}(Y)$.

Property 6 (P6): Covariance and expectation: $\mathrm{Cov}(X, Y) = E(XY) - \mu_X\mu_Y$.

Supplementary B: Results derived in the main paper

Supplementary B1 – Variances in baseline-by-training interaction model

The following data generating model was presented in the main paper to describe a participant-by-training interaction based on the baseline values of participants in the intervention group:

$$Y_{ij1} = Y_{ij0} + \beta_0 + \beta_1 X_{1_{.j1}} + \beta_2 X_{2_{.j1}} Y_{ij0} + \zeta_{Ext_{i.1}}, \qquad eq.1$$

Y_{ijk}, where i is the i-th person in the j-th group {0=control, 1=intervention}, and k is the time point {0=pre-training, 1=post-training}. We have $Y_{.j0} \sim N(\mu_0, \varphi^2)$, β_0 is the mean change in the control group, $X_{1_{.j1}}$ and $X_{2_{.j1}}$ are binary indicator variables that take on the value 0 for a participant in the control, and 1 for a participant in the intervention, β_1 is required to set the expected change of the intervention relative to control, β_2 controls the direction and magnitude of the relationship between baseline performance and change, $\zeta_{Ext_{i.1}} \sim N(0, \tau_{Ext}^2)$ is used to describe external factors that cause variation independent of any interaction with the training stimulus. We also have the observed scores $y_{ijk} = Y_{ijk} + \epsilon_{ijk}$, where $\epsilon_{ijk} \sim N(0, \delta^2)$ and is independent of $\zeta_{Ext_{i.1}}$. For the model presented in eq.1 we have the following variances:

$$\mathrm{Var}(Y_{.j0}) = \varphi^2$$

$$\mathrm{Var}(Y_{.01}) = \varphi^2 + \tau_{Ext}^2$$

$$\mathrm{Var}(Y_{.11}) = (1 + \beta_2)^2 \varphi^2 + \tau_{Ext}^2.$$

For variances of the observed random variables y_{ijk}, we simply add δ^2.

We can see that the variance for the post training scores will be greater for the intervention group compared to control if $\beta_2 > 0$, and will be less if $-1 < \beta_2 < 0$. In general, we expect the latter.

We now derive the variances of the change values $\Delta_{ij.} = y_{ij1} - y_{ij0}$. For the control we have:

$$\begin{aligned}\mathrm{Var}(\Delta_{.0.}) &= \mathrm{Var}(y_{.01}) + \mathrm{Var}(y_{.00}) - 2\mathrm{Cov}(y_{.01}, y_{.00}) \\ &= \varphi^2 + \delta^2 + \varphi^2 + \tau_{Ext}^2 + \delta^2 - 2\mathrm{Cov}(y_{.01}, y_{.00}),\end{aligned}$$

$$\begin{aligned}\mathrm{Cov}(y_{.01}, y_{.00}) &= E\left((Y_{.00} + \epsilon_{.00})(Y_{.00} + \beta_0 + \zeta_{Ext_{i11}} + \epsilon_{.01})\right) - \mu_{y_{.01}} \mu_{y_{.00}} \\ &= E(Y_{.00}^2) + \beta_0 E(Y_{.00}) - \mu_0(\mu_0 + \beta_0) \\ &= \mathrm{Var}(Y_{.00}) + \mu_0^2 + \beta_0 \mu_0 - \mu_0^2 - \beta_0 \mu_0 \\ &= \varphi^2,\end{aligned}$$

Hence:

$$\mathrm{Var}(\Delta_{.0.}) = \tau_{Ext}^2 + 2\delta^2. \qquad \text{Result 1}$$

For the intervention we have:

$$\begin{aligned}\mathrm{Var}(\Delta_{.1.}) &= \mathrm{Var}(y_{.11}) + \mathrm{Var}(y_{.10}) - 2\mathrm{Cov}(y_{.11}, y_{.10}) \\ &= \varphi^2 + \delta^2 + (1+\beta_2)^2\varphi^2 + \tau_{Ext}^2 + \delta^2 - 2\mathrm{Cov}(y_{.11}, y_{.10}) \\ \mathrm{Cov}(y_{.11}, y_{.10}) &= E\left((Y_{.10} + \epsilon_{.10})\left(Y_{.10} + \beta_0 + \beta_1 + \beta_2 Y_{.10} + \zeta_{Ext_{i11}} + \epsilon_{.11}\right)\right) - \mu_{y_{.11}}\mu_{y_{.10}} \\ &= E(Y_{.10}^2) + (\beta_0 + \beta_1)E(Y_{.10}) + \beta_2 E(Y_{.10}^2) - \mu_0(\mu_0 + \beta_0 + \beta_1 + \beta_2\mu_0) \\ &= \mathrm{Var}(Y_{.10}) + \mu_0^2 + (\beta_0 + \beta_1)\mu_0 + \beta_2(\mathrm{Var}(Y_{.10}) + \mu_0^2) - \left((\beta_2 + 1)\mu_0^2 + (\beta_0 + \beta_1)\mu_0\right) \\ &= (1+\beta_2)\varphi^2.\end{aligned}$$

Hence:

$$\begin{aligned}\mathrm{Var}(\Delta_{.1.}) &= (1+\beta_2)^2\varphi^2 - 2(1+\beta_2)\varphi^2 + \varphi^2 + \tau_{Ext}^2 + 2\delta^2 \\ &= (\beta_2^2 + 2\beta_2 + 1)\varphi^2 - 2\varphi^2 - 2\beta_2\varphi^2 + \varphi^2 + \tau_{Ext}^2 + 2\delta^2 \\ &= \beta_2^2\varphi^2 + \tau_{Ext}^2 + 2\delta^2. \end{aligned}$$ Result 2

The results above show that for the data generating mechanism presented in eq.1, the variance of the change values for the intervention is greater compared to control, and that the relative increase is based on the value of β_2 regardless of sign.

Supplementary B2 – Results of the SD_{IR}

The following data generating model was presented in the main paper as the canonical model for the standard deviation of individual response (SD_{IR}):

$$Y_{ij1} = Y_{ij0} + \beta_0 + \beta_1 X_{1_{.j1}} + \zeta_{Train_{i11}} + \zeta_{Ext_{i.1}}. \qquad eq.2$$

where $\zeta_{Train_{i11}} \sim N(0, \tau_{Train}^2)$ is experienced only by the intervention group and the two random effect terms are independent for those in the intervention.

$$\begin{aligned}SD_{IR} &= \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.})} \qquad eq.3 \\ &= \sqrt{\sigma_{\Delta_1}^2 - \sigma_{\Delta_0}^2},\end{aligned}$$

where $\sigma_{\Delta_j}^2$ are the population parameters describing the variance of the change values for group j.

From the previous section $\mathrm{Var}(\Delta_{.0.}) = \tau_{Ext}^2 + 2\delta^2$.

For $\Delta_{i1.}$ and the data generating mechanism in eq.2 we have

$$\begin{aligned}\mathrm{Var}(\Delta_{.1.}) &= \mathrm{Var}(y_{.11}) + \mathrm{Var}(y_{.10}) - 2\mathrm{Cov}(y_{.11}, y_{.10}) \\ &= \varphi^2 + \delta^2 + \varphi^2 + \tau_{Train}^2 + \tau_{Ext}^2 + \delta^2 - 2\mathrm{Cov}(y_{.11}, y_{.10})\end{aligned}$$

$$\begin{aligned}\mathrm{Cov}(y_{.11}, y_{.10}) &= E\left((Y_{.10} + \epsilon_{.10})(Y_{.10} + \beta_0 + \beta_1 + \zeta_{Train_{i11}} + \zeta_{Ext_{i.1}} + \epsilon_{.11})\right) - \mu_{y_{.11}}\mu_{y_{.10}} \\ &= E(Y_{.10}^2) + (\beta_0 + \beta_1)E(Y_{.10}) - \mu_0(\mu_0 + \beta_0 + \beta_1) \\ &= \mathrm{Var}(Y_{.10}) + \mu_0^2 + (\beta_0 + \beta_1)\mu_0 - \mu_0^2 - (\beta_0 + \beta_1)\mu_0 \\ &= \varphi^2.\end{aligned}$$

Hence:

$$\begin{aligned}\mathrm{Var}(\Delta_{.1.}) &= \varphi^2 + \delta^2 + \varphi^2 + \tau_{Train}^2 + \tau_{Ext}^2 + \delta^2 - 2\varphi^2 \\ &= \tau_{Train}^2 + \tau_{Ext}^2 + 2\delta^2.\end{aligned}$$

With the variance in change values for both groups, we show that:

$$\begin{aligned}SD_{IR} &= \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.})} \\ &= \sqrt{\tau_{Train}^2 + \tau_{Ext}^2 + 2\delta^2 - \tau_{Ext}^2 - 2\delta^2} \\ &= \tau_{Train.}\end{aligned}$$ Result 3

Supplementary B3 – SD_{IR} estimate and confidence intervals

Let $S_{\Delta_j}^2$ be the sampling variance of the change values in group j, such that:

$$S_{\Delta_j}^2 = \frac{1}{n_j - 1}\sum_{r=1}^{n_j}\left(\Delta_{rj.} - \overline{\Delta}_{.j.}\right)^2.$$

Our sample statistic to estimate SD_{IR} is denoted $\widehat{SD}_{IR}$, and we have that:

$$\widehat{SD}_{IR} = \sqrt{S_{\Delta_1}^2 - S_{\Delta_0}^2}. \qquad eq.4$$

The most popular method to calculate confidence intervals for SD_{IR} uses the following steps: 1) calculate $\widehat{SD}_{IR}^2$; 2) estimate the standard error of $\widehat{SD}_{IR}^2$; 3) assume a normal distribution for $\widehat{SD}_{IR}^2$; 4) calculate a $100(1-\alpha)\%$ confidence interval for SD_{IR}^2 by adding and subtracting the required multiple of the standard error from $\widehat{SD}_{IR}^2$ using the inverse function of the standard normal cumulative distribution function; and 5) take the square root of the confidence limits to obtain a $100(1-\alpha)\%$ confidence interval for SD_{IR}. Note, that if any of the limits for SD_{IR}^2 are negative, then the square root of the absolute value of the limit should be used and then the negative reintroduced.

Given $\widehat{SD}_{IR}^2 = S_{\Delta_1}^2 - S_{\Delta_0}^2$, the standard error is expressed as $\sqrt{\mathrm{Var}\left(S_{\Delta_1}^2 - S_{\Delta_0}^2\right)}$. Since $S_{\Delta_1}^2$ and $S_{\Delta_0}^2$ are independent, we have that $\sqrt{\mathrm{Var}\left(S_{\Delta_1}^2 - S_{\Delta_0}^2\right)} = \sqrt{\mathrm{Var}\left(S_{\Delta_1}^2\right) + \mathrm{Var}\left(S_{\Delta_0}^2\right)}$ Using Cochran's theorem (1934) we have that $(n_j - 1)S_{\Delta_j}^2 \sim \sigma_{\Delta_j}^2 \chi_{n_j-1}^2$. Since the chi-squared distribution with $n_j - 1$ degrees of freedom has variance $2(n_j - 1)$ we note that:

$$\begin{aligned}\mathrm{Var}\left(S_{\Delta_j}^2\right) &= \mathrm{Var}\left(\frac{\sigma_{\Delta_j}^2}{n_j-1}\chi_{n_j-1}^2\right)\\ &= \left(\frac{\sigma_{\Delta_j}^2}{n_j-1}\right)^2 \mathrm{Var}\left(\chi_{n_j-1}^2\right)\\ &= \left(\frac{\sigma_{\Delta_j}^2}{n_j-1}\right)^2 2(n_j-1)\\ &= \frac{2\sigma_{\Delta_j}^4}{n_j-1}.\end{aligned}$$

This quantity is then estimated by replacing $\sigma_{\Delta_j}^4$ with $S_{\Delta_j}^4$to give $\mathrm{Var}\left(S_{\Delta_j}^2\right) \approx \frac{2S_{\Delta_j}^4}{n_j-1}$. Therefore, we estimate the standard error of $\widehat{SD}_{IR}^2$ with:

$$\sqrt{\mathrm{Var}(S_{\Delta_1}^2) + \mathrm{Var}(S_{\Delta_0}^2)} \approx \sqrt{2\left(\frac{S_{\Delta_0}^4}{n_0-1} + \frac{S_{\Delta_1}^4}{n_1-1}\right)}.$$

Given the assumption of a normal distribution, a realisation of a $100(1-\alpha)\%$ confidence interval can then be created with:

$$\widehat{SD}_{IR}^2 + F_z^{-1}(\alpha/2)\sqrt{2\left(\frac{S_{\Delta_0}^4}{n_0-1} + \frac{S_{\Delta_1}^4}{n_1-1}\right)} < SD_{IR}^2 < \widehat{SD}_{IR}^2 + F_z^{-1}(1-\alpha/2)\sqrt{2\left(\frac{S_{\Delta_0}^4}{n_0-1} + \frac{S_{\Delta_1}^4}{n_1-1}\right)}. \qquad eq.5$$

where F_z^{-1} is the inverse of the standard normal cumulative distribution function.

A second method proposed by Hecksteden et al, 2018 is to use a chi-squared distribution for $\widehat{SD}_{IR}^2$ and then create a $100(1-\alpha)\%$ confidence interval with:

$$\frac{(n_1-1)\widehat{SD}_{IR}^2}{F_{\chi_{n_1-1}^2}^{-1}(1-\alpha/2)} < SD_{IR}^2 < \frac{(n_1-1)\widehat{SD}_{IR}^2}{F_{\chi_{n_1-1}^2}^{-1}(\alpha/2)}, \qquad eq.6$$

where $F_{\chi_{n_1-1}^2}^{-1}$ is the inverse of the cumulative distribution function for a chi-squared random variable with n_1-1 degrees of freedom. As with the previous method, $100(1-\alpha)\%$ confidence limits are obtained for SD_{IR} by taking the square root of the initial limits and accounting for negative values if required. It is important to note, however, that whilst $\frac{(n_j-1)S_{\Delta_j}^2}{\sigma_{\Delta_j}^2}$ follows a chi-squared distribution, the difference between two random variables with a chi-squared distribution follows either a VarianceGamma distribution (if the degrees of freedom are equal; Ferarri 2019) or a Gamma difference distribution (where the degrees of freedom are not equal; Klar 2015). These findings are based on noting that the chi-squared distribution is a special case of a Gamma distribution. Using the shape k and scale θ parameterization for the Gamma distribution $\Gamma(k,\theta)$, we have the following probability distribution function:

$$f(x;k,\theta) = \frac{x^{k-1}e^{-x/\theta}}{\theta^k\Gamma(k)}.$$

For the Chi-squared distribution with ν degrees of freedom we have:

$f(x;\nu) = \frac{x^{(\nu/2)-1}e^{-x/2}}{2^{\nu/2}\Gamma(\nu/2)}$.

We can see that the Chi-squared is $\Gamma\left(\frac{\nu}{2}, 2\right)$, and from Ferrari (2019) the case where we have two chi-squared random variables X_1, X_2 with the same degrees of freedom ν, then their difference follows a VarianceGamma (VG) distribution:

$X = X_1 - X_2 \sim VG\left(0, 2\sqrt{\nu}, 0, \frac{2}{\nu}\right)$, with location, spread, asymmetry and shape parameters. When degrees of freedom are not equal, the probability distribution function for the Gamma difference distribution is provided in Klar (2015).

Given the $\widehat{SD}_{IR}^2$ does not follow a chi-squared distribution, as identified in the simulations documented in the supplementary R code for this review, the confidence intervals proposed by Hecksteden et al, 2018 are not likely to map to the proportions required.

Finally, we introduce a third method for creating confidence intervals that maps to the $\widehat{SD}_{IR}^2$ sampling distribution across all sample sizes. The method is referred to as "melded confidence intervals" and can be used when we have confidence intervals for two parameters θ_1 and θ_2 from separate samples and wish to create a confidence interval for $g(\theta_1, \theta_2)$ such as $\theta_2 - \theta_1$. Given the data obtained from the two samples $(\boldsymbol{x_1}, \boldsymbol{x_2})$, the approach requires the $100\alpha\%$ one-sided lower and upper confidence limits $L_{\theta_i}(\boldsymbol{x_i}, \alpha)$ and $U_{\theta_i}(\boldsymbol{x_i}\alpha)$, respectively. The $100(1-\alpha)\%$ lower and upper one-sided melded confidence limits for $\beta = g(\theta_1, \theta_2)$ are then given with:

$L_\beta(\boldsymbol{x}, 1-\alpha) =$ the α-th quantile of $g\{U_{\theta_1}(\boldsymbol{x_1}, A), L_{\theta_2}(\boldsymbol{x_1}, B),\}$

$U_\beta(\boldsymbol{x}, 1-\alpha) =$ the $(1-\alpha)$-th quantile of $g\{L_{\theta_1}(\boldsymbol{x_1}, A), U_{\theta_2}(\boldsymbol{x_1}, B),\}$,

where A and B are independent and uniform random variables, such that the melded intervals can be calculated using Monte Carlo simulation or numeric integration.

For $SD_{IR}^2 = \sigma_{\Delta_1}^2 - \sigma_{\Delta_0}^2$ and its estimate $\widehat{SD}_{IR}^2 = S_{\Delta_1}^2 - S_{\Delta_0}^2$, we have $(n_j - 1)S_{\Delta_j}^2 \sim \sigma_{\Delta_j}^2 \chi_{n_j-1}^2$, hence to create a confidence interval for $\sigma_{\Delta_j}^2$ we take our pivot $\frac{(n_j-1)S_{\Delta_j}^2}{\sigma_{\Delta_j}^2}$ and note that:

$$P\left\{F^{-1}_{\chi^2_{n_j-1}}(\alpha/2) < \frac{(n_j-1)S_{\Delta_j}^2}{\sigma_{\Delta_j}^2} < F^{-1}_{\chi^2_{n_j-1}}(1-\alpha/2)\right\} = 1-\alpha \;\rightarrow$$

$$P\left\{\frac{F^{-1}_{\chi^2_{n_j-1}}(\alpha/2)}{(n_j-1)S_{\Delta_j}^2} < \frac{1}{\sigma_{\Delta_j}^2} < \frac{F^{-1}_{\chi^2_{n_j-1}}(1-\alpha/2)}{(n_j-1)S_{\Delta_j}^2}\right\} = 1-\alpha \;\rightarrow$$

$$P\left\{\frac{(n_j-1)S_{\Delta_j}^2}{F^{-1}_{\chi^2_{n_j-1}}(\alpha/2)} > \sigma_{\Delta_j}^2 > \frac{(n_j-1)S_{\Delta_j}^2}{F^{-1}_{\chi^2_{n_j-1}}(1-\alpha/2)}\right\} = 1-\alpha \;\rightarrow$$

$$P\left\{\frac{(n_j-1)S^2_{\Delta_j}}{F^{-1}_{\chi^2_{n_j-1}}(1-\alpha/2)} < \sigma^2_{\Delta_j} < \frac{(n_j-1)S^2_{\Delta_j}}{F^{-1}_{\chi^2_{n_j-1}}(\alpha/2)}\right\} = 1-\alpha.$$

The $100(1-\alpha)\%$ lower and upper one-sided melded confidence limits for SD^2_{IR} are thus obtained with:

$L_{SD^2_{IR}}(\boldsymbol{x_i}, \alpha) =$ the α-th quantile of $(n_1-1)S^2_{\Delta_1}/F^{-1}_{\chi^2_{n_1-1}}(A) - (n_0-1)S^2_{\Delta_0}/F^{-1}_{\chi^2_{n_0-1}}(B)$

$U_{SD^2_{IR}}(\boldsymbol{x_i}, \alpha) =$ the $(1-\alpha)$-th quantile of $(n_1-1)S^2_{\Delta_1}/F^{-1}_{\chi^2_{n_1-1}}(A) - (n_0-1)S^2_{\Delta_0}/F^{-1}_{\chi^2_{n_0-1}}(B)$ $\quad eq.7$

The above shows that for Monte Carlo simulation we would obtain two separate samples (A and B) from a Uniform(0,1); e.g. in R using runif(#sample). We would then plug these samples into Result 6; e.g. in R using qchisq(A, n_1-1) and take the α-th and $(1-\alpha)$-th quantiles. As with the previous methods, $100(1-\alpha)\%$ confidence limits are obtained for SD_{IR} by taking the square root of the initial limits and accounting for negative values if required.

Supplementary B4 – Model misspecification, post training error magnitude

Model misspecification in terms of differential measurement error for groups post-intervention would cause the data generating model in eq.2 to be updated such that $\epsilon_{ij1} \sim N(0, \delta^2)$ is replaced with $\epsilon_{ij1} \sim N(0, \delta^2_{.j1})$. First, we examine what effect this may have on the quantity SD_{IR} to represent τ_{Train}. With this new data generating model we have:

$$\mathrm{Var}(\Delta_{.0.}) = \tau^2_{Ext} + \delta^2 + \delta^2_{.01}.$$

$$\mathrm{Var}(\Delta_{.1.}) = \tau^2_{Train} + \tau^2_{Ext} + \delta^2 + \delta^2_{.11}.$$

$$SD_{IR} = \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.})}$$

$$= \sqrt{\tau^2_{Train} + \tau^2_{Ext} + \delta^2 + \delta^2_{.11} - \tau^2_{Ext} - \delta^2 - \delta^2_{.01}}$$

$$= \sqrt{\tau^2_{Train} + \delta^2_{.11} - \delta^2_{.01}}. \qquad \text{Result 3}$$

Assume in addition to y_{ij1} we have additional post training measurements $y_{ij1_A} = Y_{ij1} + \epsilon_{ij1_A}$, and $y_{ij1_B} = Y_{ij1} + \epsilon_{ij1_B}$. We will use the difference in these measurements which we define $\Delta_{ij1_{AB}} = y_{ij1_B} - y_{ij1_A}$. Across the two groups we have:

$$\mathrm{Var}(\Delta_{i01_{AB}}) = \mathrm{Var}(y_{.01B}) + \mathrm{Var}(y_{.01A}) - 2\mathrm{Cov}(y_{.01B}, y_{.01A})$$

$$= \varphi^2 + \tau^2_{Ext} + \delta^2_{.01} + \varphi^2 + \tau^2_{Ext} + \delta^2_{.01} - 2\mathrm{Cov}(y_{.01B}, y_{.01A})$$

$$= \varphi^2 + \tau^2_{Ext} + \delta^2_{.01} + \varphi^2 + \tau^2_{Ext} + \delta^2_{.01} - 2\varphi^2 - 2\tau^2_{Ext}$$

$$= 2\delta^2_{.01}.$$

$$\begin{aligned}\mathrm{Var}\left(\Delta_{i11_{AB}}\right) &= \mathrm{Var}(y_{.11B}) + \mathrm{Var}(y_{.11A}) - 2\mathrm{Cov}(y_{.11B}, y_{.11A}) \\ &= \varphi^2 + \tau^2_{Train} + \tau^2_{Ext} + \delta^2_{.11} + \varphi^2 + \tau^2_{Train} + \tau^2_{Ext} + \delta^2_{.11} - 2\mathrm{Cov}(y_{.11B}, y_{.11A}) \\ &= \varphi^2 + \tau^2_{Train} + \tau^2_{Ext} + \delta^2_{.11} + \varphi^2 + \tau^2_{Train} + \tau^2_{Ext} + \delta^2_{.11} - 2\varphi^2 - 2\tau^2_{Train} - 2\tau^2_{Ext} \\ &= 2\delta^2_{.11}.\end{aligned}$$

We now wish to use an updated SD_{IR} denoted $\widetilde{SD}_{IR}$, to account for the updated data generating model such that $\widetilde{SD}_{IR} = \tau_{Train}$. To obtain this we use:

$$\widetilde{SD}_{IR} = \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.}) + \frac{1}{2}\left(\mathrm{Var}\left(\Delta_{i01_{AB}}\right) - \mathrm{Var}\left(\Delta_{i11_{AB}}\right)\right)} \qquad eq.8$$

$$= \sqrt{\tau^2_{Train} + \delta^2_{.11} - \delta^2_{.01} + \frac{1}{2}(2\delta^2_{.01} - 2\delta^2_{.11})}$$

$$= \tau_{Train} \qquad \text{Result 4}$$

We estimate $\widetilde{SD}_{IR}$ with $\widehat{\widetilde{SD}}_{IR}$, where:

$$\widehat{\widetilde{SD}}_{IR} = \sqrt{S^2_{\Delta_1} - S^2_{\Delta_0} + \frac{1}{2}\left(S^2_{\Delta_{0_{AB}}} - S^2_{\Delta_{1_{AB}}}\right)},$$

where $S^2_{\Delta_{j_{AB}}}$ is the sample variance of the difference scores in the two post-training measurements used to estimate reliability in group j. To create confidence intervals for $\widetilde{SD}_{IR}$, we assume a normal distribution for $\widehat{\widetilde{SD}}^2_{IR}$, estimate the standard error, calculate the $100(1-\alpha)\%$ confidence interval and take the square of the limits as done previously. Using a similar process as previous, we estimate the standard error of $\widehat{\widetilde{SD}}^2_{IR}$ as:

$$\sqrt{\mathrm{Var}\left(\widehat{\widetilde{SD}}^2_{IR}\right)} = \sqrt{\mathrm{Var}\left(S^2_{\Delta_1} - S^2_{\Delta_0} + \frac{1}{2}\left(S^2_{\Delta_{0_{AB}}} - S^2_{\Delta_{1_{AB}}}\right)\right)}.$$

Expanding the different terms we can see that the sample variances are independent of each other such that:

$$\mathrm{Var}\left(S^2_{\Delta_1} - S^2_{\Delta_0} + \frac{1}{2}\left(S^2_{\Delta_{0_{AB}}} - S^2_{\Delta_{1_{AB}}}\right)\right) = \mathrm{Var}\left(S^2_{\Delta_1}\right) + \mathrm{Var}\left(S^2_{\Delta_0}\right) + \frac{1}{4}\left(\mathrm{Var}\left(S^2_{\Delta_{0_{AB}}}\right) + \mathrm{Var}\left(S^2_{\Delta_{1_{AB}}}\right)\right).$$

Based on the same reasoning with the chi-squared distribution as previous, we then have

$$\sqrt{\mathrm{Var}\left(S^2_{\Delta_1}\right) + \mathrm{Var}\left(S^2_{\Delta_0}\right) + \frac{1}{4}\left(\mathrm{Var}\left(S^2_{\Delta_{0_{AB}}}\right) + \mathrm{Var}\left(S^2_{\Delta_{1_{AB}}}\right)\right)} \approx \sqrt{2\left(\frac{S^4_{\Delta_0}}{n_0-1} + \frac{S^4_{\Delta_1}}{n_1-1} + \frac{S^4_{\Delta_{0_{AB}}}}{4(n_0-1)} + \frac{S^4_{\Delta_{1_{AB}}}}{4(n_1-1)}\right)},$$

with a $100(1-\alpha)\%$ confidence interval obtained with:

$$\widehat{\widetilde{SD}}^2_{IR} + F_z^{-1}(\alpha/2)\sqrt{2\left(\frac{S^4_{\Delta_0}}{n_0-1} + \frac{S^4_{\Delta_1}}{n_1-1} + \frac{S^4_{\Delta_{0_{AB}}}}{4(n_0-1)} + \frac{S^4_{\Delta_{1_{AB}}}}{4(n_1-1)}\right)} < \widetilde{SD}^2_{IR} < \widehat{\widetilde{SD}}^2_{IR} + F_z^{-1}(1-\alpha/2)\sqrt{2\left(\frac{S^4_{\Delta_0}}{n_0-1} + \frac{S^4_{\Delta_1}}{n_1-1} + \frac{S^4_{\Delta_{0_{AB}}}}{4(n_0-1)} + \frac{S^4_{\Delta_{1_{AB}}}}{4(n_1-1)}\right)},$$

where F_z^{-1} is the inverse of the standard normal cumulative distribution function.

We now examine the calculations required when instead of conducting two separate post-training measures to account for model misspecification, we include the more likely research design where two measurements are conducted both pre- $\left(y_{ij0_A}, y_{ij0_B}\right)$ and post-training $\left(y_{ij1_A}, y_{ij1_B}\right)$, and both the average $\left(\bar{y}_{ijk_{AB}} = \frac{1}{2}\left(y_{ijk_A} + y_{ijk_B}\right)\right)$ and differences within $\left(\Delta_{ij1_{AB}} = y_{ij1_B} - y_{ij1_A}\right)$ and between $\left(\Delta_{ij.AB} = \bar{y}_{ij1_{AB}} - \bar{y}_{ij0_{AB}}\right)$ measurement times are used. we first derive some preliminary results.

$$\begin{aligned}\mathrm{Var}\left(\bar{y}_{.j0_{AB}}\right) &= \mathrm{Var}\left(\tfrac{1}{2}\left(y_{.j0_A} + y_{.j0_B}\right)\right)\\ &= \frac{1}{4}\left(\mathrm{Var}\left(y_{.j0_A}\right) + \mathrm{Var}\left(y_{.j0_B}\right) + 2\mathrm{Cov}\left(y_{.j0_A}, y_{.j0_B}\right)\right)\\ &= \frac{1}{4}\left(2\varphi^2 + 2\delta^2 + 2\varphi^2\right)\\ &= \varphi^2 + \frac{\delta^2}{2}.\end{aligned}$$

$$\begin{aligned}\mathrm{Var}\left(\bar{y}_{.01_{AB}}\right) &= \mathrm{Var}\left(\tfrac{1}{2}\left(y_{.01_A} + y_{.01_B}\right)\right)\\ &= \frac{1}{4}\left(\mathrm{Var}\left(y_{.01_A}\right) + \mathrm{Var}\left(y_{.01_B}\right) + 2\mathrm{Cov}\left(y_{.01_A}, y_{.01_B}\right)\right)\\ &= \frac{1}{4}\left(2\varphi^2 + 2\tau_{Ext}^2 + 2\delta_{.01}^2 + 2\varphi^2 + 2\tau_{Ext}^2\right)\\ &= \varphi^2 + \tau_{Ext}^2 + \frac{\delta_{.01}^2}{2}.\end{aligned}$$

$$\begin{aligned}\mathrm{Var}\left(\bar{y}_{.11_{AB}}\right) &= \mathrm{Var}\left(\tfrac{1}{2}\left(y_{.11_A} + y_{.11_B}\right)\right)\\ &= \frac{1}{4}\left(\mathrm{Var}\left(y_{.11_A}\right) + \mathrm{Var}\left(y_{.11_B}\right) + 2\mathrm{Cov}\left(y_{.11_A}, y_{.11_B}\right)\right)\\ &= \frac{1}{4}\left(2\varphi^2 + 2\tau_{Train}^2 + 2\tau_{Ext}^2 + 2\delta_{.11}^2 + 2\varphi^2 + 2\tau_{Train}^2 + 2\tau_{Ext}^2\right)\\ &= \varphi^2 + \tau_{Train}^2 + \tau_{Ext}^2 + \frac{\delta_{.11}^2}{2}.\end{aligned}$$

$$\begin{aligned}\mathrm{Var}\left(\Delta_{.0._{AB}}\right) &= \mathrm{Var}\left(\bar{y}_{.01_{AB}} - \bar{y}_{.00_{AB}}\right)\\ &= \varphi^2 + \tau_{Ext}^2 + \frac{\delta_{.01}^2}{2} + \varphi^2 + \frac{\delta^2}{2} - 2\mathrm{Cov}\left(\bar{y}_{.00_{AB}}, \bar{y}_{.01_{AB}}\right).\end{aligned}$$

$$\mathrm{Cov}(\bar{y}_{.00_{AB}}, \bar{y}_{.01_{AB}}) = \frac{1}{4}E\left(\left(y_{.00_A} + y_{.00_B}\right)\left(y_{.01_A} + y_{.01_B}\right)\right) - \frac{1}{4}\mu_{\bar{y}_{.00_{AB}}}\mu_{\bar{y}_{.01_{AB}}}$$

$$= \frac{1}{4}E\left(\left(2Y_{.00} + \epsilon_{.00_A} + \epsilon_{.00_B}\right)\left(2Y_{.00} + 2\beta_0 + 2\zeta_{Ext_{i.1}} + \epsilon_{.01_A} + \epsilon_{.01_B}\right)\right) - \frac{1}{4}\left(2\mu_0 2(\mu_0 + \beta_0)\right)$$

$$= \frac{1}{4}\left(4\left(E(Y_{.00}^2)\right) + 4\beta_0 E(Y_{.00})\right) - \mu_0^2 - \mu_0\beta_0$$

$$= \mathrm{Var}(Y_{.00}) + \mu_0^2 + \mu_0\beta_0 - \mu_0^2 - \mu_0\beta_0$$

$$= \varphi^2$$

Hence:

$$\mathrm{Var}\left(\Delta_{.0._{AB}}\right) = \tau_{Ext}^2 + \frac{1}{2}(\delta^2 + \delta_{.01}^2)$$

Similarly,

$$\mathrm{Var}\left(\Delta_{.1._{AB}}\right) = \mathrm{Var}\left(\bar{y}_{.11_{AB}} - \bar{y}_{.10_{AB}}\right)$$

$$= \varphi^2 + \tau_{Train}^2 + \tau_{Ext}^2 + \frac{\delta_{.11}^2}{2} + \varphi^2 + \frac{\delta^2}{2} - 2\mathrm{Cov}\left(\bar{y}_{.10_{AB}}, \bar{y}_{.11_{AB}}\right).$$

$$= \tau_{Train}^2 + \tau_{Ext}^2 + \frac{1}{2}(\delta^2 + \delta_{.11}^2)$$

Given

$\mathrm{Var}\left(\Delta_{ij1_{AB}}\right) = 2\delta_{.j1}^2$, then under the current design with two pre- and two post-training measurements we can obtain τ_{Train} with

$$\widetilde{SD}_{IR} = \sqrt{\mathrm{Var}\left(\Delta_{.1._{AB}}\right) - \mathrm{Var}\left(\Delta_{.0._{AB}}\right) + \frac{1}{4}\left(\mathrm{Var}\left(\Delta_{i01_{AB}}\right) - \mathrm{Var}\left(\Delta_{i11_{AB}}\right)\right)}$$

$$= \sqrt{\tau_{Train}^2 + \tau_{Ext}^2 + \frac{1}{2}(\delta^2 + \delta_{.11}^2) - \tau_{Ext}^2 - \frac{1}{2}(\delta^2 + \delta_{.01}^2) + \frac{1}{4}(2\delta_{.01}^2 - 2\delta_{.11}^2)}$$

$$= \tau_{Train}\,.$$ Result 5

As was done previously, we estimate $\widetilde{SD}_{IR}$ with $\widehat{\widetilde{SD}}_{IR}$, where:

$$\widehat{\widetilde{SD}}_{IR} = \sqrt{S_{\bar{\Delta}_1}^2 - S_{\bar{\Delta}_0}^2 + \frac{1}{4}\left(S_{\Delta_{0_{AB}}}^2 - S_{\Delta_{1_{AB}}}^2\right)},$$

where $S_{\bar{\Delta}_j}^2$ is the sample variance of the difference in the average values across the two measurements pre- and post-training, and $S_{\Delta_{j_{AB}}}^2$ is the sample variance of the difference scores in the two post-training measurements used to estimate reliability in group j. Using the same processes as done previously, we estimate the standard error and calculate the $100(1-\alpha)\%$ confidence interval with:

$$\sqrt{\mathrm{Var}\left(\widehat{\widetilde{SD}}_{IR}^2\right)} = \sqrt{\mathrm{Var}\left(S_{\bar{\Delta}_1}^2 - S_{\bar{\Delta}_0}^2 + \frac{1}{4}\left(S_{\Delta_{0_{AB}}}^2 - S_{\Delta_{1_{AB}}}^2\right)\right)}$$

$$= \sqrt{\mathrm{Var}\left(S_{\bar{\Delta}_1}^2\right) + \mathrm{Var}\left(S_{\bar{\Delta}_0}^2\right) + \frac{1}{16}\left(\mathrm{Var}\left(S_{\Delta_{0_{AB}}}^2\right) + \mathrm{Var}\left(S_{\Delta_{1_{AB}}}^2\right)\right)}$$

$$\approx \sqrt{2\left(\frac{S_{\bar{\Delta}_0}^4}{n_0-1} + \frac{S_{\bar{\Delta}_1}^4}{n_1-1} + \frac{S_{\Delta_{0_{AB}}}^4}{16(n_0-1)} + \frac{S_{\Delta_{1_{AB}}}^4}{16(n_1-1)}\right)},$$

with a $100(1-\alpha)\%$ confidence interval obtained with:

$$\widehat{\widetilde{SD}}_{IR}^2 + F_z^{-1}(\alpha/2)\sqrt{2\left(\frac{S_{\bar{\Delta}_0}^4}{n_0-1} + \frac{S_{\bar{\Delta}_1}^4}{n_1-1} + \frac{S_{\Delta_{0_{AB}}}^4}{16(n_0-1)} + \frac{S_{\Delta_{1_{AB}}}^4}{16(n_1-1)}\right)} < \widetilde{SD}_{IR}^2 < \widehat{\widetilde{SD}}_{IR}^2 + F_z^{-1}(1-\alpha/2)\sqrt{2\left(\frac{S_{\bar{\Delta}_0}^4}{n_0-1} + \frac{S_{\bar{\Delta}_1}^4}{n_1-1} + \frac{S_{\Delta_{0_{AB}}}^4}{16(n_0-1)} + \frac{S_{\Delta_{1_{AB}}}^4}{16(n_1-1)}\right)},$$

where F_z^{-1} is the inverse of the standard normal cumulative distribution function.

Supplementary B5 – variance heterogeneity in external factors

To explore variance heterogeneity, we introduce the final data generating model:

$$Y_{ij1} = Y_{ij0} + \beta_0 + \beta_1 X_{1_{.j1}} + \beta_2 X_{2_{.j1}} + \zeta_{Train_{i11}} + \zeta_{Ext_{i.1}}. \qquad eq.9$$

where $X_{2_{ij1}}$ is a covariate measuring an external factor that influences the post-training value. β_2 quantifies the magnitude and direction of the influence and is constant across individuals and groups. We have $X_{2_{ij1}} \sim N\left(\psi_{2_{.j1}}, \tau_{2_{.j1}}\right)$ such that the mean and variance can differ across groups, and $X_{2_{ij1}}$ is independent of $\zeta_{Train_{i11}}$ and $\zeta_{Ext_{i.1}}$. Here we show that if the standard definition of the SD_{IR} is used we do not return τ_{Train} as desired.

$$SD_{IR} = \sqrt{\mathrm{Var}(\Delta_{.1.}) - \mathrm{Var}(\Delta_{.0.})}$$

$$= \sqrt{\beta_2^2\tau_{2_{.11}}^2 + \tau_{Train}^2 + \tau_{Ext}^2 + 2\delta^2 - \beta_2^2\tau_{2_{.01}}^2 - \tau_{Ext}^2 - 2\delta^2}$$

$$= \sqrt{\tau_{Train}^2 + \beta_2^2\left(\tau_{2_{.11}}^2 - \tau_{2_{.01}}^2\right)}. \qquad \text{Result 6}$$

Supplementary C: R code

```
# Supplementary C: R code
# In the following file we demonstrate in R the supplementary results derived
# Load packages
library(ggplot2)
library(tidybayes)

# Supplementary B1 – Variances in baseline-by-training interaction model

# Fist we create a function to generate data according to eq.1 in the main paper
# and supplementary file
DataCreateModelEq1 = function(n0,n1,Beta_2, Y_ij0_mu = 100, Y_ij0_sd = 15,Beta_0
= 5,Beta_1 = 15,
                              tau_Extsd = 6,epsilon_sd = 2){
  Y_i00 = rnorm(n0,Y_ij0_mu,Y_ij0_sd)
  Y_i10 = rnorm(n1,Y_ij0_mu,Y_ij0_sd)
  Y_i01 = Y_i00 + Beta_0 + rnorm(n0,0,tau_Extsd)
  Y_i11 = Y_i10 + Beta_0 + Beta_1 + Beta_2*Y_i10  + rnorm(n1,0,tau_Extsd)
  y_i00 = Y_i00 + rnorm(n0,0,epsilon_sd)
  y_i10 = Y_i10 + rnorm(n1,0,epsilon_sd)
  y_i01 = Y_i01 + rnorm(n0,0,epsilon_sd)
  y_i11 = Y_i11 + rnorm(n1,0,epsilon_sd)
  diff0 = y_i01-y_i00
  diff1 = y_i11-y_i10
  return(list(diff0,diff1,y_i00,y_i01,y_i10,y_i11))}

# We create three sets of data, the first where beta_2 = 0,
# the second with beta_2 = -0.7,
# the third with beta_2 = 0.7
set.seed(123)
Eq1Data1 = DataCreateModelEq1(1000000,1000000,0)
Eq1Data2 = DataCreateModelEq1(1000000,1000000,-0.7)
Eq1Data3 = DataCreateModelEq1(1000000,1000000,0.7)

# We show that when beta_2 = 0 that the change value variance is the same for
# intervention and control
round(var(Eq1Data1[[1]]),1)
# 44
round(var(Eq1Data1[[2]]),1)
# 44

# We show that when beta_2 \neq 0 that the change value variance is greater
# for the intervention and the same regardless of sign
round(var(Eq1Data2[[2]]),1)
# 154.4
round(var(Eq1Data3[[2]]),1)
# 154.4

# Check Result 1
round(var(Eq1Data1[[1]]),1)
# 44
6^2 + 2*(2^2)
# 44

# Check Result 2
round(var(Eq1Data2[[2]]),0)
# 154
round(((-0.7)^2)*(15^2) + 6^2 + 2*(2^2),0)
# 154

# Supplementary B2 – Results of the SD_IR
```

```
### Graphical overview of SDIR

SDIRPlot = data.frame(Value = c(rnorm(10000,10,8),rnorm(10000,10,5)),
                              Fill = c(rep("A",10000),rep("B",10000)))

ggplot(SDIRPlot, aes(x=Value, fill=Fill)) +
  geom_density(alpha=.25, adjust = 2) + theme_classic() + xlab("Pre- to post-
training difference") +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) +
  geom_vline(xintercept=10,
             color = "black", size=1) +
  geom_vline(xintercept=0, linetype="dashed",
             color = "red", size=0.8) +
  annotate(geom="text", x=22.5, y=0.075,
           label=expression(paste("Individual       response       =       N(Mean
change,",tau[Train]^2,")")))+
annotate(geom="text", x=32, y=0.025,
         label=expression(paste("Gross         response         =         N(Mean
change,",tau[Train]^2,"+",
                                tau[Ext]^2,"+",epsilon^2,")"))) +
  theme(legend.position = "none") + scale_x_continuous(breaks=seq(-20,40,10))


# Fist we create a function to generate data according to eq.2
DataCreateModelEq2  =  function(n0,n1,Y_ij0_mu  =  100,  Y_ij0_sd  =  15,Beta_0  =
5,Beta_1 = 15,
                           tau_Trainsd = 10,tau_Extsd = 6,epsilon_ijk = 2){
  Y_i00 = rnorm(n0,Y_ij0_mu,Y_ij0_sd)
  Y_i10 = rnorm(n1,Y_ij0_mu,Y_ij0_sd)
  Y_i01 = Y_i00 + Beta_0 + rnorm(n0,0,tau_Extsd)
  Y_i11   =   Y_i10   +   Beta_0   +   Beta_1   +   rnorm(n1,0,tau_Trainsd)   +
rnorm(n1,0,tau_Extsd)
  y_i00 = Y_i00 + rnorm(n0,0,epsilon_ijk)
  y_i10 = Y_i10 + rnorm(n1,0,epsilon_ijk)
  y_i01 = Y_i01 + rnorm(n0,0,epsilon_ijk)
  y_i11 = Y_i11 + rnorm(n1,0,epsilon_ijk)
  diff0 = y_i01-y_i00
  diff1 = y_i11-y_i10
  return(list(diff0,diff1))}

set.seed(123)
Eq2Data1 = DataCreateModelEq2(1000000,1000000)

# Check Result 3
round(sqrt(var(Eq2Data1[[2]])-var(Eq2Data1[[1]])),1)
# 10

# Supplementary B3 - SD_IR estimate and confidence intervals

# Normal distribution confidence intervals
NormalCI = function(n0,n1,niter,Trainsd){
  LB = c(NULL)
  UB = c(NULL)
  SDIR2 = c(NULL)
  # Run simulation
  for(i in 1:niter){
    # Simulate data
    Data = DataCreateModelEq2(n0,n1,tau_Trainsd=Trainsd)
    # Calculate SDIR^2 estimate
```

```
    SDIR2[i] = var(Data[[2]])-var(Data[[1]])
    # Calculate standard error for SDIR^2 estimate
    SE              =              sqrt(2*(sd(Data[[1]])^4/(length(Data[[1]])-
1)+sd(Data[[2]])^4/(length(Data[[2]])-1)))
    # Calculate 95% CI
    LB[i]=SDIR2[i]-1.96*SE
    UB[i]=SDIR2[i]+1.96*SE

  }
  # Proportion of intervals that include true vale for SDIR^2
  Prop = mean(LB<(Trainsd^2)&UB>(Trainsd^2))
  Out = list(SDIR2,LB,UB,Prop)
  return(Out)}

# Test on equal sample sizes of 10/20/50/100

# Normal, N0 = 10; N1 = 10
set.seed(123)
NormalCI1010 = NormalCI(10,10,10000,10)
# Proportion of intervals that contain true value
NormalCI1010[[4]]
# 0.8961

# Normal, N0 = 20; N1 = 20
set.seed(123)
NormalCI2020 = NormalCI(20,20,10000,10)
# Proportion of intervals that contain true value
NormalCI2020[[4]]
# 0.918

# Normal, N0 = 50; N1 = 50
set.seed(123)
NormalCI5050 = NormalCI(50,50,10000,10)
# Proportion of intervals that contain true value
NormalCI5050[[4]]
# 0.9365

# Normal, N0 = 100; N1 = 100
set.seed(123)
NormalCI100100 = NormalCI(100,100,10000,10)
# Proportion of intervals that contain true value
NormalCI100100[[4]]
# 0.9472

# Unequal sample sizes
# Normal, N0 = 5; N1 = 10
set.seed(123)
NormalCI510 = NormalCI(5,10,10000,10)
# Proportion of intervals that contain true value
NormalCI510[[4]]
# 0.9359

# Normal, N0 = 10; N1 = 20
set.seed(123)
NormalCI1020 = NormalCI(10,20,10000,10)
# Proportion of intervals that contain true value
NormalCI1020[[4]]
# 0.9431

# Normal, N0 = 25; N1 = 50
set.seed(123)
```

```
NormalCI2550 = NormalCI(25,50,10000,10)
# Proportion of intervals that contain true value
NormalCI2550[[4]]
# 0.9496

# Normal, N0 = 50; N1 = 100
set.seed(123)
NormalCI50100 = NormalCI(50,100,10000,10)
# Proportion of intervals that contain true value
NormalCI50100[[4]]
# 0.9499

##### Chi-squared
ChiCI = function(n0,n1,niter,Trainsd){
  LB = c(NULL)
  UB = c(NULL)
  SDIR2 = c(NULL)
  # Run simulation
  for(i in 1:niter){
    # Simulate data
    Data = DataCreateModelEq2(n0,n1,tau_Trainsd=Trainsd)
    # Calculate SDIR^2 estimate
    # Take the absolute value to match the chi-squared being positive onlyu
    SDIR2[i] = var(Data[[2]])-var(Data[[1]])
    # Calculate CI bound
    # Calculate 95% CI
    LB[i]=(SDIR2[i]*(n1-1))/qchisq(0.975,(n1-1))
    UB[i]=(SDIR2[i]*(n1-1))/qchisq(0.025,(n1-1))

  }
  # Proportion of intervals that include true vale for SDIR^2
  Prop = mean(LB<(Trainsd^2)&UB>(Trainsd^2))
  Out = list(SDIR2,LB,UB,Prop)
  return(Out)}

# Test on equal sample sizes of 10/20/50/100

# Chi, N0 = 10; N1 = 10
set.seed(123)
ChiCI1010 = ChiCI(10,10,10000,10)
# Proportion of intervals that contain true value
ChiCI1010[[4]]
# 0.7822


# Chi, N0 = 20; N1 = 20
set.seed(123)
ChiCI2020 = ChiCI(20,20,10000,10)
# Proportion of intervals that contain true value
ChiCI2020[[4]]
# 0.7962

# Chi, N0 = 50; N1 = 50
set.seed(123)
ChiCI5050 = ChiCI(50,50,10000,10)
# Proportion of intervals that contain true value
ChiCI5050[[4]]
# 0.8

# Chi, N0 = 100; N1 = 100
set.seed(123)
```

```
ChiCI100100 = ChiCI(100,100,10000,10)
# Proportion of intervals that contain true value
ChiCI100100[[4]]
# 0.806

# Unequal sample sizes
# Chi, N0 = 5; N1 = 10
set.seed(123)
ChiCI510 = ChiCI(5,10,10000,10)
# Proportion of intervals that contain true value
ChiCI510[[4]]
# 0.7662

# Chi, N0 = 10; N1 = 20
set.seed(123)
ChiCI1020 = ChiCI(10,20,10000,10)
# Proportion of intervals that contain true value
ChiCI1020[[4]]
# 0.7756

# Chi, N0 = 25; N1 = 50
set.seed(123)
ChiCI2550 = ChiCI(25,50,10000,10)
# Proportion of intervals that contain true value
ChiCI2550[[4]]
# 0.7843

# Chi, N0 = 50; N1 = 100
set.seed(123)
ChiCI50100 = ChiCI(50,100,10000,10)
# Proportion of intervals that contain true value
ChiCI50100[[4]]
# 0.783

# Melded
MeldCI = function(n0,n1,niter,MCiter,Trainsd){
  LB = c(NULL)
  UB = c(NULL)
  SDIR2 = c(NULL)
  # Run simulation
  for(i in 1:niter){
    # Simulate data
    Data = DataCreateModelEq2(n0,n1,tau_Trainsd=Trainsd)
    # Calculate SDIR^2 estimate
    SDIR2[i] = var(Data[[2]])-var(Data[[1]])
    # Calculate CI bound
    Chisq0                =                (sd(Data[[1]])^2)*(length(Data[[1]])-
1)/qchisq(runif(MCiter),(length(Data[[1]])-1))
    Chisq1                =                (sd(Data[[2]])^2)*(length(Data[[2]])-
1)/qchisq(runif(MCiter),(length(Data[[2]])-1))
    Bound = Chisq1-Chisq0
    LB[i]=quantile(Bound,0.025)
    UB[i]=quantile(Bound,0.975)

  }
  # Proportion of intervals that include true vale for SDIR^2
  Prop = mean(LB<(Trainsd^2)&UB>(Trainsd^2))
  Out = list(SDIR2,LB,UB,Prop)
  return(Out)}

# Test on equal sample sizes of 10/20/50/100
```

```
# Meld, N0 = 10; N1 = 10
set.seed(123)
MeldCI1010 = MeldCI(10,10,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI1010[[4]]
# 0.9514


# Meld, N0 = 20; N1 = 20
set.seed(123)
MeldCI2020 = MeldCI(20,20,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI2020[[4]]
# 0.9496

# Meld, N0 = 50; N1 = 50
set.seed(123)
MeldCI5050 = MeldCI(50,50,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI5050[[4]]
# 0.9514

# Meld, N0 = 100; N1 = 100
set.seed(123)
MeldCI100100 = MeldCI(100,100,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI100100[[4]]
# 0.9512

# Unequal sample sizes
# Meld, N0 = 5; N1 = 10
set.seed(123)
MeldCI510 = MeldCI(5,10,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI510[[4]]
# 0.9551

# Meld, N0 = 10; N1 = 20
set.seed(123)
MeldCI1020 = MeldCI(10,20,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI1020[[4]]
# 0.9515

# Meld, N0 = 25; N1 = 50
set.seed(123)
MeldCI2550 = MeldCI(25,50,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI2550[[4]]
# 0.948

# Meld, N0 = 50; N1 = 100
set.seed(123)
MeldCI50100 = MeldCI(50,100,10000,1000,10)
# Proportion of intervals that contain true value
MeldCI50100[[4]]
# 0.9488

# Plots
```

```
# SDIR2
SDIR2DF = data.frame(SDIR2 = c(NormalCI1010[[1]],NormalCI2020[[1]],
                               NormalCI5050[[1]],NormalCI100100[[1]]),
                     Group = c(rep("N1=10",length(NormalCI1010[[1]])),
                               rep("N1=20",length(NormalCI2020[[1]])),
                               rep("N1=50",length(NormalCI5050[[1]])),
                               rep("N1=100",length(NormalCI100100[[1]]))))

SDIR2DF$Group = factor(SDIR2DF$Group, levels=c("N1=10","N1=20",
                                               "N1=50","N1=100"))

SDIR2DFOR = SDIR2DF[SDIR2DF$SDIR2<300,]
ggplot(SDIR2DFOR,aes(x = Group, y = SDIR2, fill=Group)) +
  stat_halfeye() +  theme_classic() + theme(legend.position="none")  +
  labs(x="", y=expression(widehat(SD)[IR]^2)) +
  geom_hline(yintercept = 100, color = "red", linetype=2) +
  scale_x_discrete(labels=c("N1=10" = bquote("n" [1]~"=10"), "N1=20" = bquote("n"
[1]~"=20"),
                            "N1=50" = bquote("n" [1]~"=50"),"N1=100" = bquote("n"
[1]~"=100")))

# CI Plot
CIDFPlot = data.frame(Bound = c(NormalCI1010[[2]],NormalCI2020[[2]],
                                NormalCI5050[[2]],NormalCI100100[[2]],
                                NormalCI1010[[3]],NormalCI2020[[3]],
                                NormalCI5050[[3]],NormalCI100100[[3]],

                                ChiCI1010[[2]],ChiCI2020[[2]],
                                ChiCI5050[[2]],ChiCI100100[[2]],
                                ChiCI1010[[3]],ChiCI2020[[3]],
                                ChiCI5050[[3]],ChiCI100100[[3]],

                                MeldCI1010[[2]],MeldCI2020[[2]],
                                MeldCI5050[[2]],MeldCI100100[[2]],
                                MeldCI1010[[3]],MeldCI2020[[3]],
                                MeldCI5050[[3]],MeldCI100100[[3]]),

                          Group                                                 =
c(rep(c(rep("N1=10",length(NormalCI1010[[1]])),
                                  rep("N1=20",length(NormalCI2020[[1]])),
                                  rep("N1=50",length(NormalCI5050[[1]])),

rep("N1=100",length(NormalCI100100[[1]]))),2),

                                rep(c(rep("N1=10",length(ChiCI1010[[1]])),
                                      rep("N1=20",length(ChiCI2020[[1]])),
                                      rep("N1=50",length(ChiCI5050[[1]])),

rep("N1=100",length(ChiCI100100[[1]]))),2),

                                rep(c(rep("N1=10",length(MeldCI1010[[1]])),
                                      rep("N1=20",length(MeldCI2020[[1]])),
                                      rep("N1=50",length(MeldCI5050[[1]])),

rep("N1=100",length(MeldCI100100[[1]]))),2)),

                                BoundType                                       =
c(rep("LB",(length(NormalCI1010[[1]])+
                                           length(NormalCI1010[[1]])+
                                           length(NormalCI1010[[1]])+
                                           length(NormalCI1010[[1]]))),
```

```
rep("UB",(length(NormalCI1010[[1]])+
                                                  length(NormalCI1010[[1]])+
                                                  length(NormalCI1010[[1]])+
                                                  length(NormalCI1010[[1]]))),


rep("LB",(length(ChiCI1010[[1]])+

length(ChiCI1010[[1]])+

length(ChiCI1010[[1]])+

length(ChiCI1010[[1]]))),

rep("UB",(length(ChiCI1010[[1]])+

length(ChiCI1010[[1]])+

length(ChiCI1010[[1]])+

length(ChiCI1010[[1]]))),


rep("LB",(length(MeldCI1010[[1]])+

length(MeldCI1010[[1]])+

length(MeldCI1010[[1]])+

length(MeldCI1010[[1]]))),

rep("UB",(length(MeldCI1010[[1]])+

length(MeldCI1010[[1]])+

length(MeldCI1010[[1]])+

length(MeldCI1010[[1]])))),

                  Distribution = c(rep(rep("Normal",(length(NormalCI1010[[1]])+
                                                     length(NormalCI1010[[1]])+
                                                     length(NormalCI1010[[1]])+
                                                     length(NormalCI1010[[1]]))),2),

                  rep(rep("Chi-squared",(length(ChiCI1010[[1]])+
                                         length(ChiCI1010[[1]])+
                                         length(ChiCI1010[[1]])+
                                         length(ChiCI1010[[1]]))),2),

                  rep(rep("Melded",(length(MeldCI1010[[1]])+
                                    length(MeldCI1010[[1]])+
                                    length(MeldCI1010[[1]])+
                                    length(MeldCI1010[[1]]))),2)))


CIDFPlot$Group = factor(CIDFPlot$Group, levels=c("N1=10","N1=20",
                                                 "N1=50","N1=100"))

ggplot(CIDFPlot[CIDFPlot$Bound<700&
```

```
                  CIDFPlot$Bound>-200,],aes(x = Distribution, y = Bound, fill =
BoundType)) +
  stat_halfeye(position = "dodge",width=0.6) +
  theme_classic() + theme(legend.position="none")  +
  labs(x="", y=expression(widehat(SD)[IR]^2)) +
  scale_y_continuous(breaks = seq(-200,600,200))+
  geom_hline(yintercept = 100, color = "red", linetype=2) +
  facet_wrap(~Group)


# Supplementary B4 – Model misspecification, post training error magnitude

# Function to create data
DataCreatePostR = function(n0,n1,Y_ij0_mu = 100, Y_ij0_sd = 15,Beta_0 = 5,Beta_1
= 15,
                           tau_Trainsd = 10,tau_Extsd = 6,epsilon_ij0 = 4,
                           epsilon_i01 = 4, epsilon_i11 = 2){
  Y_i00 = rnorm(n0,Y_ij0_mu,Y_ij0_sd)
  Y_i10 = rnorm(n1,Y_ij0_mu,Y_ij0_sd)
  Y_i01 = Y_i00 + Beta_0 + rnorm(n0,0,tau_Extsd)
  Y_i11   =   Y_i10   +   Beta_0   +   Beta_1   +   rnorm(n1,0,tau_Trainsd)   +
rnorm(n1,0,tau_Extsd)
  y_i00 = Y_i00 + rnorm(n0,0,epsilon_ij0)
  y_i10 = Y_i10 + rnorm(n1,0,epsilon_ij0)
  y_i01 = Y_i01 + rnorm(n0,0,epsilon_i01)
  y_i01a = Y_i01 + rnorm(n0,0,epsilon_i01)
  y_i01b = Y_i01 + rnorm(n0,0,epsilon_i01)
  y_i11 = Y_i11 + rnorm(n1,0,epsilon_i11)
  y_i11a = Y_i11 + rnorm(n1,0,epsilon_i11)
  y_i11b = Y_i11 + rnorm(n1,0,epsilon_i11)
  diff0 =  y_i01 - y_i00
  diff1 =  y_i11 - y_i10
  diff0AB = y_i01b - y_i01a
  diff1AB = y_i11b - y_i11a
  return(list(diff0,diff1,diff0AB,diff1AB))}

# Check Result 4
ModelMisAB = DataCreatePostR(10000,10000)
round(sqrt(var(ModelMisAB[[2]])-var(ModelMisAB[[1]])+
       0.5*var(ModelMisAB[[3]])+0.5*var(ModelMisAB[[4]])),0)
# 10


# Function to create CIs
NormalCICompare = function(n0,n1,niter,Trainsd){
  LBOriginal = c(NULL)
  UBOriginal = c(NULL)
  LBUpdate = c(NULL)
  UBUpdate = c(NULL)
  SDIR2Original = c(NULL)
  SDIR2Update = c(NULL)
  # Run simulation
  for(i in 1:niter){
    # Simulate data
    Data = DataCreatePostR(n0,n1,tau_Trainsd=Trainsd)
    # Calculate SDIR^2 estimates
    SDIR2Original[i] = var(Data[[2]])-var(Data[[1]])
    SDIR2Update[i]   =   var(Data[[2]])-var(Data[[1]])   +   0.5*var(Data[[3]])   -
0.5*var(Data[[4]])
    # Calculate standard error for SDIR^2 estimate
```

```
    SEOriginal           =           sqrt(2*(sd(Data[[1]])^4/(length(Data[[1]])-
1)+sd(Data[[2]])^4/(length(Data[[2]])-1)))
    SEUpdate             =           sqrt(2*(sd(Data[[1]])^4/(length(Data[[1]])-
1)+sd(Data[[2]])^4/(length(Data[[2]])-1)+
                          sd(Data[[3]])^4/(4*(length(Data[[1]])-
1))+sd(Data[[4]])^4/(4*(length(Data[[1]])-1))))
    # Calculate 95% CI
    LBOriginal[i]=SDIR2Original[i]-1.96*SEOriginal
    UBOriginal[i]=SDIR2Original[i]+1.96*SEOriginal
    LBUpdate[i]=SDIR2Update[i]-1.96*SEUpdate
    UBUpdate[i]=SDIR2Update[i]+1.96*SEUpdate

  }
  # Proportion of intervals that include true vale for SDIR^2
  PropOriginal = mean(LBOriginal<(Trainsd^2)&UBOriginal>(Trainsd^2))
  PropUpdate = mean(LBUpdate<(Trainsd^2)&UBUpdate>(Trainsd^2))
  Out = list(SDIR2Original,SDIR2Update,LBOriginal,UBOriginal,
             LBUpdate,UBUpdate,PropOriginal,PropUpdate )
  return(Out)}

# Normal, N0 = 10; N1 = 10
set.seed(123)
NormalCI1010Compare = NormalCICompare(10,10,10000,10)
# Proportion of intervals that contain true value
NormalCI1010Compare[[7]]
# 0.8907
NormalCI1010Compare[[8]]
# 0.9199

# Normal, N0 = 20; N1 = 20
set.seed(123)
NormalCI2020Compare = NormalCICompare(20,20,10000,10)
# Proportion of intervals that contain true value
NormalCI2020Compare[[7]]
# 0.9055
NormalCI2020Compare[[8]]
# 0.9389

# Normal, N0 = 50; N1 = 50
set.seed(123)
NormalCI5050Compare = NormalCICompare(50,50,10000,10)
# Proportion of intervals that contain true value
NormalCI5050Compare[[7]]
# 0.9018
NormalCI5050Compare[[8]]
# 0.9455

# Normal, N0 = 100; N1 = 100
set.seed(123)
NormalCI100100Compare = NormalCICompare(100,100,10000,10)
# Proportion of intervals that contain true value
NormalCI100100Compare[[7]]
# 0.8923
NormalCI100100Compare[[8]]
# 0.9498


#### Two points pre and post
DataCreatePostR2 = function(n0,n1,Y_ij0_mu = 100, Y_ij0_sd = 15,Beta_0 = 5,Beta_1
= 15,
                            tau_Trainsd = 10,tau_Extsd = 6,epsilon_ij0 = 4,
```

```
                          epsilon_i01 = 4, epsilon_i11 = 2){
  Y_i00 = rnorm(n0,Y_ij0_mu,Y_ij0_sd)
  Y_i10 = rnorm(n1,Y_ij0_mu,Y_ij0_sd)
  Y_i01 = Y_i00 + Beta_0 + rnorm(n0,0,tau_Extsd)
  Y_i11   =   Y_i10   +   Beta_0   +   Beta_1   +   rnorm(n1,0,tau_Trainsd)   +
rnorm(n1,0,tau_Extsd)
  y_i00a = Y_i00 + rnorm(n0,0,epsilon_ij0)
  y_i00b = Y_i00 + rnorm(n0,0,epsilon_ij0)
  y_i10a = Y_i10 + rnorm(n1,0,epsilon_ij0)
  y_i10b = Y_i10 + rnorm(n1,0,epsilon_ij0)
  y_i01a = Y_i01 + rnorm(n0,0,epsilon_i01)
  y_i01b = Y_i01 + rnorm(n0,0,epsilon_i01)
  y_i11a = Y_i11 + rnorm(n1,0,epsilon_i11)
  y_i11b = Y_i11 + rnorm(n1,0,epsilon_i11)
  bary_i00 = 0.5*(y_i00a+y_i00b)
  bary_i10 = 0.5*(y_i10a+y_i10b)
  bary_i01 = 0.5*(y_i01a+y_i01b)
  bary_i11 = 0.5*(y_i11a+y_i11b)
  diff0 =  bary_i01 - bary_i00
  diff1 =  bary_i11 - bary_i10
  diff0AB = y_i01b - y_i01a
  diff1AB = y_i11b - y_i11a
  return(list(diff0,diff1,diff0AB,diff1AB))}

# Check Result 5
ModelMisAB2 = DataCreatePostR2(10000,10000)
round(sqrt(var(ModelMisAB2[[2]])-var(ModelMisAB2[[1]])+
             0.25*var(ModelMisAB2[[3]])+0.25*var(ModelMisAB2[[4]])),0)
# 10

# Function to create CIs
NormalCICompare2 = function(n0,n1,niter,Trainsd){
  LB = c(NULL)
  UB = c(NULL)
  SDIR2 = c(NULL)

  # Run simulation
  for(i in 1:niter){
    # Simulate data
    Data = DataCreatePostR2(n0,n1,tau_Trainsd=Trainsd)
    # Calculate SDIR^2 estimates
    SDIR2[i]    =    var(Data[[2]])-var(Data[[1]])    +    0.25*var(Data[[3]])    -
0.25*var(Data[[4]])
    # Calculate standard error for SDIR^2 estimate
    SE                =                sqrt(2*(sd(Data[[1]])^4/(length(Data[[1]])-
1)+sd(Data[[2]])^4/(length(Data[[2]])-1)+
                          sd(Data[[3]])^4/(16*(length(Data[[1]])-
1))+sd(Data[[4]])^4/(16*(length(Data[[1]])-1))))
    # Calculate 95% CI
    LB[i]=SDIR2[i]-1.96*SE
    UB[i]=SDIR2[i]+1.96*SE

  }
  # Proportion of intervals that include true vale for SDIR^2
  Prop = mean(LB<(Trainsd^2)&UB>(Trainsd^2))
  Out = list(SDIR2,LB,UB,Prop)
  return(Out)}

# Normal, N0 = 10; N1 = 10
set.seed(123)
NormalCI1010Compare2 = NormalCICompare2(10,10,10000,10)
```

```
# Proportion of intervals that contain true value
NormalCI1010Compare2[[4]]
# 0.905

# Normal, N0 = 20; N1 = 20
set.seed(123)
NormalCI2020Compare2 = NormalCICompare2(20,20,10000,10)
# Proportion of intervals that contain true value
NormalCI2020Compare2[[4]]
# 0.9253


# Normal, N0 = 50; N1 = 50
set.seed(123)
NormalCI5050Compare2 = NormalCICompare2(50,50,10000,10)
# Proportion of intervals that contain true value
NormalCI5050Compare2[[4]]

# Normal, N0 = 100; N1 = 100
set.seed(123)
NormalCI100100Compare2 = NormalCICompare2(100,100,10000,10)
# Proportion of intervals that contain true value
NormalCI100100Compare2[[4]]

# Compare
mean(NormalCI1010Compare[[1]])
mean(NormalCI1010Compare[[2]])
mean(NormalCI1010Compare2[[1]])

median(NormalCI1010Compare[[3]])
median(NormalCI1010Compare[[5]])
median(NormalCI1010Compare2[[2]])

median(NormalCI1010Compare[[4]])
median(NormalCI1010Compare[[6]])
median(NormalCI1010Compare2[[3]])


# CI Plot
CIDFPlot2                  =                  data.frame(Bound                  =
c(NormalCI1010Compare[[3]],NormalCI2020Compare[[3]],

NormalCI5050Compare[[3]],NormalCI100100Compare[[3]],

NormalCI1010Compare[[4]],NormalCI2020Compare[[4]],

NormalCI5050Compare[[4]],NormalCI100100Compare[[4]],


NormalCI1010Compare[[5]],NormalCI2020Compare[[5]],

NormalCI5050Compare[[5]],NormalCI100100Compare[[5]],

NormalCI1010Compare[[6]],NormalCI2020Compare[[6]],

NormalCI5050Compare[[6]],NormalCI100100Compare[[6]],


NormalCI1010Compare2[[2]],NormalCI2020Compare2[[2]],

NormalCI5050Compare2[[2]],NormalCI100100Compare2[[2]],
```

```
NormalCI1010Compare2[[3]],NormalCI2020Compare2[[3]],

NormalCI5050Compare2[[3]],NormalCI100100Compare2[[3]]),

                      Group                                                    =
c(rep(c(rep("N1=10",length(NormalCI1010Compare[[1]])),

rep("N1=20",length(NormalCI2020Compare[[1]])),

rep("N1=50",length(NormalCI5050Compare[[1]])),

rep("N1=100",length(NormalCI100100Compare[[1]]))),2),


rep(c(rep("N1=10",length(NormalCI1010Compare[[1]])),

rep("N1=20",length(NormalCI2020Compare[[1]])),

rep("N1=50",length(NormalCI5050Compare[[1]])),

rep("N1=100",length(NormalCI100100Compare[[1]]))),2),


rep(c(rep("N1=10",length(NormalCI1010Compare2[[1]])),

rep("N1=20",length(NormalCI2020Compare2[[1]])),

rep("N1=50",length(NormalCI5050Compare2[[1]])),

rep("N1=100",length(NormalCI100100Compare2[[1]]))),2)),

                      BoundType = c(rep("LB",(length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]]))),
                                    rep("UB",(length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]]))),

                                    rep("LB",(length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]]))),
                                    rep("UB",(length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]]))),
```

```
                                    rep("LB",(length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]]))),
                                    rep("UB",(length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])))),

                      Distribution                                                =
c(rep(rep("Normal0",(length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]]))),2),


rep(rep("Normal1",(length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]])+

length(NormalCI1010Compare[[1]]))),2),


rep(rep("Normal2",(length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]])+

length(NormalCI1010Compare2[[1]]))),2)))


CIDFPlot2$Group = factor(CIDFPlot2$Group, levels=c("N1=10","N1=20",
                                                   "N1=50","N1=100"))

ggplot(CIDFPlot2[CIDFPlot2$Bound<700&
                   CIDFPlot2$Bound>-200,],aes(x = Distribution, y = Bound, fill =
BoundType)) +
  stat_halfeye(position = "dodge",width=0.6) +
  theme_classic() + theme(legend.position="none")  +
  labs(x="", y=expression(widehat(SD)[IR]^2)) +
  scale_y_continuous(breaks = seq(-200,600,200))+
  geom_hline(yintercept = 100, color = "red", linetype=2) +
  facet_wrap(~Group)

# Supplementary B5 – variance heterogeneity in external factors

# Fist we create a function to generate data according to eq.9 in the main paper
# and supplementary file
```

```
DataCreateModelEq9 = function(n0,n1,Beta_2, Y_ij0_mu = 100, Y_ij0_sd = 15,Beta_0
= 5,Beta_1 = 15,
                              tau_Trainsd = 10, tau_Extsd = 6,epsilon_sd = 2,
                              X2mu = 5, X2sd0 = 4, X2sd1 = 2){
  Y_i00 = rnorm(n0,Y_ij0_mu,Y_ij0_sd)
  Y_i10 = rnorm(n1,Y_ij0_mu,Y_ij0_sd)
  X2_i01 = rnorm(n0,X2mu,X2sd0)
  X2_i11 = rnorm(n0,X2mu,X2sd1)
  Y_i01 = Y_i00 + Beta_0 + Beta_2*X2_i01 + rnorm(n0,0,tau_Extsd)
  Y_i11 = Y_i10 + Beta_0 + Beta_1 + Beta_2*X2_i11  + rnorm(n1,0,tau_Trainsd) +
rnorm(n1,0,tau_Extsd)
  y_i00 = Y_i00 + rnorm(n0,0,epsilon_sd)
  y_i10 = Y_i10 + rnorm(n1,0,epsilon_sd)
  y_i01 = Y_i01 + rnorm(n0,0,epsilon_sd)
  y_i11 = Y_i11 + rnorm(n1,0,epsilon_sd)
  diff0 = y_i01-y_i00
  diff1 = y_i11-y_i10
  return(list(diff0,diff1,y_i00,y_i01,y_i10,y_i11))}

# We create three sets of data, the first where beta_2 = 0,
# the second with beta_2 = -2,
# the third with beta_2 = 2
set.seed(123)
Eq9Data1 = DataCreateModelEq9(1000000,1000000,0)
Eq9Data2 = DataCreateModelEq9(1000000,1000000,-2)
Eq9Data3 = DataCreateModelEq9(1000000,1000000,2)

# We show that when beta_2 = 0 that the SD_IR returns the correct value
round(sqrt(var(Eq9Data1[[2]])-var(Eq9Data1[[1]])),1)

# Check Result 6
# We show that when beta_2 \neq 0 that the SD_IR returns Result 6 in the
supplementary
round(sqrt(var(Eq9Data2[[2]])-var(Eq9Data2[[1]])),1)
# 7.2
round(sqrt(var(Eq9Data3[[2]])-var(Eq9Data3[[1]])),1)
# 7.2
round(sqrt(10^2 + ((2^2)*((2^2)-(4^2)))),1)
# 7.2
```