Version of Record at: https://doi.org/10.1093/jrsssc/qlad063

A Supporting Information

A.1 R code to implement simulation study

```
#########################################################################
## Set initial parameters, assuming following 2x2 table of ratings
##             | Rater 1 |
##             |  0   1  |
## ------------------------
## Rater 2  0  |  a   b  | n0.
##          1  |  c   d  | n1.
## ------------------------
##             | n.0 n.1 | N
#########################################################################

## Initialize output matrices
agreeEstAvgAll=agreeEstSDAll=agreeEstCvgAll=numeric(0)

prevs_n.1=seq(0.1,0.9,by=0.1)
#prevs_n.1=0.5
for(kk in 1:length(prevs_n.1)){
  ## Prevalance for rater 1
  prev_n.1=prevs_n.1[kk]

  biases=seq(0,min(prev_n.1,0.3),by=0.1)
  for(ll in 1:length(biases)){

    guesses=rbind(c(0,0),c(0.2,0.2),c(0.5,0.5),c(0.2,0.5),c(0.8,0.8),c(1,1))
    for(mm in 1:nrow(guesses)){

      guesstypes=c("Rand","Marg")
      for(pp in 1:length(guesstypes)){

        guessWrongs=c(0,0.1,0.3,0.5)
        for(qq in 1:length(guessWrongs)){

          sampleSizes=c(25,50,100,200,500)
          for(nn in 1:length(sampleSizes)){
            ## Bias across raters (Rater 1 - Rater 2)
            bias=biases[ll]

            ## Prevalence for rater 2
            prev_n1.=prev_n.1-bias

            ## Rating probabilities
            prob_d=seq(max(prev_n1.+prev_n.1-1,0),min(prev_n1.,prev_n.1),by=0.1)
            prob_c=prev_n1.-prob_d
            prob_b=prob_c+bias
            prob_a=1-(prob_b+prob_c+prob_d)
            probs=cbind(prob_a,prob_b,prob_c,prob_d)

            ## True agreement
            trueagree=prob_a+prob_d

            ## Proportion of observations subject to guessing for each rater
            rater1guessprop=guesses[mm,1]
            rater2guessprop=guesses[mm,2]

            ## GuessType
```

Version of Record at: https://doi.org/10.1093/jrsssc/qlad063

```
guesstype=guesstypes[pp]

## Wrong guess amount
guessWrong=guessWrongs[qq]

## Sample Size
N=sampleSizes[nn]

## Number of simulations
M=10000

## Probabilities
print("Combinations of probabilities to consider")
print(probs)

#########################################################
## Sample Data
#########################################################
numAgreeStats=7
agreeEstAvg=cbind(probs,
                  matrix(NA,nrow=nrow(probs),ncol=numAgreeStats*2))
agreeEstSD=cbind(probs,
                 matrix(NA,nrow=nrow(probs),ncol=numAgreeStats))
agreeEstCvg=cbind(probs,matrix(NA,nrow=nrow(probs),ncol=1))
colnames(agreeEstSD)=c(colnames(probs),
                       "PctAgree","Cohen","Scott","Gwets","PABAK","Kripp","Ours")
colnames(agreeEstAvg)=c(colnames(probs),
                        "PctAgree","PctAgreeSE","Cohen","CohenSE",
                        "Scott","ScottSE","Gwets","GwetsSE","PABAK","PABAKSE",
                        "Kripp","KrippSE","Ours","OursSE")
colnames(agreeEstCvg)=c(colnames(probs),"NormCICov")

for(j in 1:nrow(probs)){
  agreeEsts=matrix(NA,nrow=M,ncol=numAgreeStats*2)
  agreeEstsSD=matrix(NA,nrow=M,ncol=numAgreeStats)
  agreeCIs=matrix(NA,nrow=M,ncol=1)
  for(m in 1:M){
    set.seed(m)

    ## Generation of ratings
    ratings=sample(c("a","b","c","d"),N,replace=TRUE,prob=probs[j,])
    raterdata=data.frame(cbind(id=1:N,
                               rater1=(ratings=="b" | ratings=="d"),
                               rater2=(ratings=="c" | ratings=="d")))

    ## Change some to guesses
    rater1guess=sample(c(0,1),N,replace=TRUE,
                       prob=c(1-rater1guessprop,rater1guessprop))
    rater2guess=sample(c(0,1),N,replace=TRUE,
                       prob=c(1-rater2guessprop,rater2guessprop))
    if(guesstype=="Rand"){
      rater1newrating=sample(c(0,1),N,replace=TRUE,prob=c(0.5,0.5))
      rater2newrating=sample(c(0,1),N,replace=TRUE,prob=c(0.5,0.5))
    } else if(guesstype=="Marg"){
      rater1newrating=sample(c(0,1),N,replace=TRUE,
                             prob=c(1-prev_n.1,prev_n.1))
      rater2newrating=sample(c(0,1),N,replace=TRUE,
                             prob=c(1-prev_n1.,prev_n1.))
    }
```

Version of Record at: https://doi.org/10.1093/jrsssc/qlad063

```
ratings[rater1guess==1 & ratings=="a" & rater1newrating==1]="b"
ratings[rater1guess==1 & ratings=="b" & rater1newrating==0]="a"
ratings[rater1guess==1 & ratings=="c" & rater1newrating==1]="d"
ratings[rater1guess==1 & ratings=="d" & rater1newrating==0]="c"
ratings[rater2guess==1 & ratings=="a" & rater2newrating==1]="c"
ratings[rater2guess==1 & ratings=="c" & rater2newrating==0]="a"
ratings[rater2guess==1 & ratings=="b" & rater2newrating==1]="d"
ratings[rater2guess==1 & ratings=="d" & rater2newrating==0]="b"

## Number of each pair of ratings
aa=sum(ratings=="a")
bb=sum(ratings=="b")
cc=sum(ratings=="c")
dd=sum(ratings=="d")

## Marginals
n.0=aa+cc
n.1=bb+dd
n0.=aa+bb
n1.=cc+dd

## Agreement Statistics
p=(aa+dd)/N
e_k=(n0.*n.0)/N^2+(n1.*n.1)/N^2
cohen=(p-e_k)/(1-e_k)
e_pi=((n.0+n0.)/(2*N))^2+((n.1+n1.)/(2*N))^2
scott=(p-e_pi)/(1-e_pi)
e_g=2*((n.0+n0.)/(2*N))*(1-((n.0+n0.)/(2*N)))
gwets=(p-e_g)/(1-e_g)
pabak=2*(aa+dd)/N-1
kripp=(((1-1/(2*N))*p+1/(2*N))-e_pi)/(1-e_pi)

## Mark guesses, with some error based on wrongly
## indicating uncertainty as certainty
anyguess=(rater1guess==1 | rater2guess==1)
guessErr1=guessErr2=rep(0,length(anyguess))
guessErr1=sample(0:1,length(rater1guess),
                 prob=c(1-guessWrong,guessWrong),replace=TRUE)
guessErr2=sample(0:1,length(rater2guess),
                 prob=c(1-guessWrong,guessWrong),replace=TRUE)
anyguessErr=((rater1guess*(1-guessErr1)+
                 (1-rater1guess)*guessErr1)==1 |
               (rater2guess*(1-guessErr2)+
                 (1-rater2guess)*guessErr2)==1)

## Ours empirically estimates probability of chance agreement
e_z=(sum(ratings[anyguessErr==1]=="a")+
     sum(ratings[anyguessErr==1]=="d"))/N
ours2=(p-e_z)/(1-e_z)

# SE estimates
pSE=sqrt(1/N*p*(1-p))
cohenSE=sqrt(aa/N*(1-e_k-(n0.+n.0)/N*(1-p))^2+ #Fleiss1969
               dd/N*(1-e_k-(n1.+n.1)/N*(1-p))^2+
               (1-p)^2*(bb/N*(n.1/N+n0./N)^2+
                        cc/N*(n.0/N+n.1/N)^2)-
               (p*e_k-2*e_k+p)^2)/((1-e_k)^2*sqrt(N))
scottSE=sqrt(p*(1-p)-4*(1-scott)*
```

```
                        4*(1-scott)^2*(aa/N*((n0.+n.0+n0.+n.0)/N/2/2)^2+
                                       bb/N*((n0.+n.0+n1.+n.1)/N/2/2)^2+
                                       cc/N*((n1.+n.1+n0.+n.0)/N/2/2)^2+
                                       dd/N*((n1.+n.1+n1.+n.1)/N/2/2)^2-
                                       e_k^2))/((1-e_pi)*sqrt(N))
    gwetsSE=sqrt(p*(1-p)-4*(1-gwets)*
                 (aa/N*(1-n0./N/2-n.0/N/2)+dd/N*(1-n1./N/2-n.1/N/2)-p*e_g)+
                   4*(1-gwets)^2*(aa/N*(1-(n0.+n.0+n0.+n.0)/N/2/2)^2+
                                  bb/N*(1-(n0.+n.0+n1.+n.1)/N/2/2)^2+
                                  cc/N*(1-(n1.+n.1+n0.+n.0)/N/2/2)^2+
                                  dd/N*(1-(n1.+n.1+n1.+n.1)/N/2/2)^2-
                                  e_g^2))/((1-e_g)*sqrt(N))
    pabakSE=sqrt(p*(1-p))/((1-1/2)*sqrt(N))
    krippSE=sqrt(aa/N*(1-1/(2*N)-(1-kripp)*(n0.+n.0+n0.+n.0)/N/2)^2+
                 bb/N*(0-(1-kripp)*(n0.+n.0+n1.+n.1)/N/2)^2+
                 cc/N*(0-(1-kripp)*(n1.+n.1+n0.+n.0)/N/2)^2+
                 dd/N*(1-1/(2*N)-(1-kripp)*(n1.+n.1+n1.+n.1)/N/2)^2-
                 ((1-1/(2*N))*p-2*(1-kripp)*e_pi)^2)/
                  ((1-e_pi)*sqrt(N))

    pX=(sum(ratings[anyguessErr==0]=="a")+
        sum(ratings[anyguessErr==0]=="d"))/N
    pY=1-(sum(ratings[anyguessErr==1]=="a")+
          sum(ratings[anyguessErr==1]=="d"))/N
    sigsquared=(1-pX)/(N*pX)-(1-pY)/(N*pY)
    oursSE=sqrt((exp(sigsquared)-1)*exp(2*log(ours2)+sigsquared))

    ## 95 pct CI coverage of our estimator
    normalCI=(trueagree[j]>(ours2-qnorm(0.975)*oursSE) &
              trueagree[j]<(ours2+qnorm(0.975)*oursSE))

    agreeEsts[m,]=c(p,pSE,cohen,cohenSE,scott,scottSE,gwets,gwetsSE,
                    pabak,pabakSE,kripp,krippSE,ours2,oursSE)
    agreeEstsSD[m,]=c(p,cohen,scott,gwets,pabak,kripp,ours2)
    agreeCIs[m,]=c(normalCI)

    if (m %% 1000 == 0) print(paste("Simulation",m,"Done"))
  }
  agreeEstAvg[j,(ncol(probs)+1):ncol(agreeEstAvg)]=
    apply(agreeEsts,2,mean,na.rm=TRUE)
  agreeEstSD[j,(ncol(probs)+1):ncol(agreeEstSD)]=
    apply(agreeEstsSD,2,sd,na.rm=TRUE)
  agreeEstCvg[j,(ncol(probs)+1):ncol(agreeEstCvg)]=
    apply(agreeCIs,2,mean,na.rm=TRUE)
  print(paste("Probability combination",j,"Done"))
}

## Append simulation results
agreeEstAvgAll=rbind(agreeEstAvgAll,
    cbind(prev_n.1,bias,guesstype,rater1guessprop,
    rater2guessprop,guessWrong,N,agreeEstAvg))
agreeEstSDAll=rbind(agreeEstSDAll,
    cbind(prev_n.1,bias,guesstype,rater1guessprop,
    rater2guessprop,guessWrong,N,agreeEstSD))
agreeEstCvgAll=rbind(agreeEstCvgAll,
    cbind(prev_n.1,bias,guesstype,rater1guessprop,
    rater2guessprop,guessWrong,N,agreeEstCvg))

## Export simulation results
```

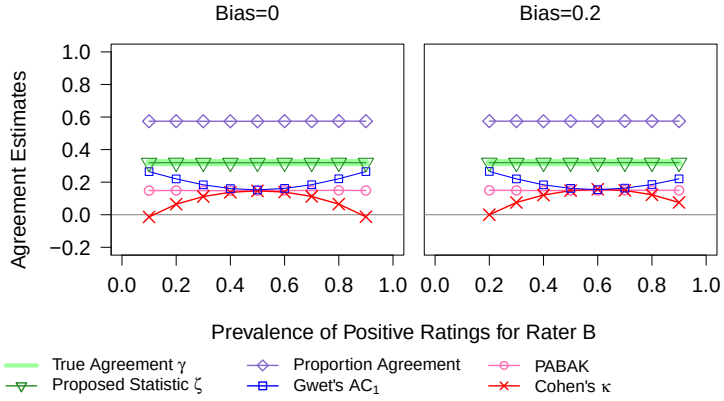
Version of Record at: https://doi.org/10.1093/jrsssc/qlad063

```
            write.csv(agreeEstAvgAll,"SimResults_agreeEstAvg.csv")
            write.csv(agreeEstSDAll,"SimResults_agreeEstSD.csv")
            write.csv(agreeEstCvgAll,"SimResults_agreeEstCvg.csv")

            print(paste("Sample Size",N,"Done"))
          }
          print(paste("Wrong Guess",guessWrong,"Done"))
        }
        print(paste("Guess Type",guesstype,"Done"))
      }
      print(paste("Guesses",rater1guessprop,rater2guessprop,"Done"))
    }
    print(paste("Bias",bias,"Done"))
  }
  print(paste("Prev",prev_n.1,"Done"))
}
```

A.2 Additional simulation results

Figure S1: Simulation results showing agreement estimates comparing bias of 0 and 0.2. Parameters used true agreement under certainty of 0.8, 50% rater uncertainty, unbiased coin flips for uncertain ratings, and $N = 200$



A.3 Dental study data example using summarized uncertainty data

We describe an example in which uncertainty indications for each observation were not collected when ratings were made. Instead, we asked raters to provide an estimated percent of observations that were previously rated with uncertainty, i.e., a summary of the presumed observation-level uncertainty data. We then made assumptions about the distribution of uncertain cases to generate a distribution of values for our proposed statistic. Ceramic crown restorations were independently evaluated by two teams of dental clinicians according to modified U.S. Public Health Service (USPHS) criteria, as described in Fasbinder et al. (2010). In the current study, the response variable of interest was margin adaptation–specifically, whether dentists agreed on whether there was evidence of a crevice between the crown and tooth.

Team A clinicians reported that they were certain in their ratings for 98% of cases on average, while the Team B clinicians reported certainty for 90% of cases. Overall, the amount of certainty was high, which likely reflects well-developed study protocols: all clinicians had at least eight (and up to 21) years of training and experience in implementing the same criteria, which have been consistent and well-recognized in the literature for many years. Team A clinicians further explained that their higher proportion of certainty

Version of Record at: https://doi.org/10.1093/jrsssc/qlad063

compared to Team B clinicians was likely due to the fact that the Team A clinicians were the ones who wrote the criteria for the study. Team B clinicians cited two reasons for uncertainty: first, the fact that some cases were especially difficult to rate and second, awareness of previous inter-rater agreement studies in which their ratings disagreed with consensus ratings.

Since Team A reported uncertainty for 2% of cases and Team B reported uncertainty for 10% of cases, we considered two "extreme" scenarios: 1) that the cases with uncertainty were completely overlapping between Team A and Team B and 2) that the cases with uncertainty had no overlap. By assuming a uniform distribution of uncertainty across the $N = 92$ cases under each of these scenarios, we randomly sampled 10% of observations to have any uncertainty under scenario 1 and 12% of observations to have any uncertainty under scenario 2. We then calculated our proposed agreement statistic ζ with repeated sampling $M = 200$ times for each scenario. Existing statistics were also calculated, although values do not change with repeated sampling. For each scenario, Figure S2 shows a histogram distribution of ζ estimates across the $m = 200$ samples, with reference lines showing the agreement estimates from existing statistics.

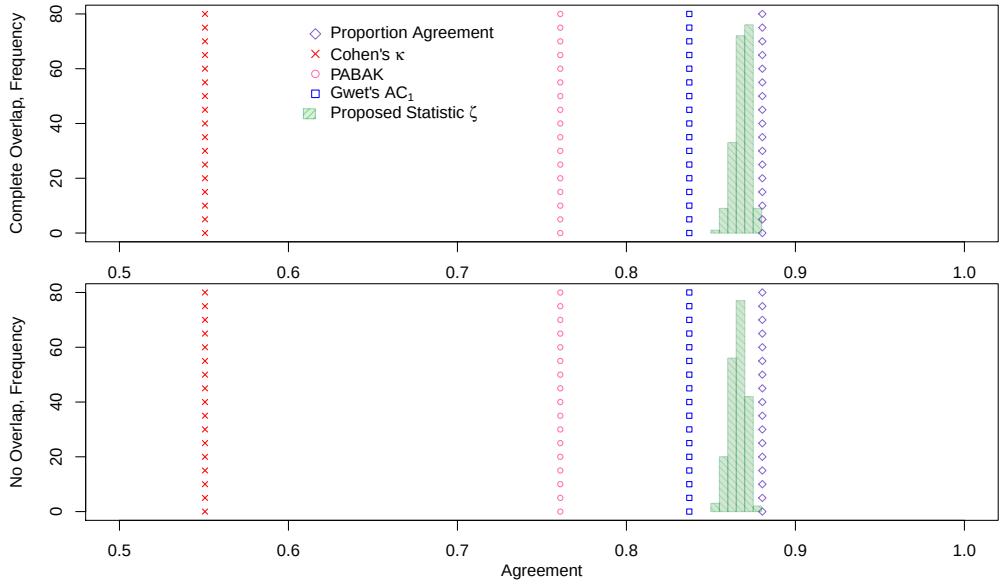


Figure S2: Agreement estimates for margin adaptation ratings of dental ceramic crown restorations, assuming no overlap (top panel) or complete overlap (bottom panel) of cases rated with uncertainty across raters.

The two extreme scenarios resulted in similar, narrow distributions of values for ζ, since the amount of uncertainty was low for all raters and extremely low for Team A in particular. The median agreement estimate for ζ was 0.87 under either scenario, indicating that inter-rater agreement was about 0.87 after chance agreement was removed from consideration. This value is higher than those of any agreement estimates using existing chance-corrected agreement statistics.

References

Fasbinder, D. J., Dennison, J. B., Heys, D. and Neiva, G. (2010) A clinical evaluation of chairside lithium disilicate cad/cam crowns. *The Journal of the American Dental Association*, **141**, 10S–14S.