

Supplemental Material

A systemic perspective on the temporal interplay between adolescent depression and resilience-supporting individual and social resources

1. Correlation tables for T1, T2, T3
2. Goodness-of-Fit Statistics
3. Factor models
4. Pruned model 1: BDI Total score, physical functionality, social supports
5. Plots for stability analyses
6. R-Code

1. Correlation Tables for T1, T2, T3

Table S1

Correlation Table for T1

Variable	1	2	3	4	5
1. Overall depression					
2. Cognitive-affective symptoms	0.99**				
3. Somatic symptoms	0.92**	0.86**			
4. Physical functioning	-0.22**	-0.21**	-0.22**		
5. Perceived family support	-0.50**	-0.50**	-0.46**	0.13*	
6. Perceived friend support	-0.34**	-0.34**	-0.31**	0.12*	0.35**

* $p < .05$, ** $p < .01$. p-values adjusted for multiple testing.

Table S2

Correlation Table for T2

Variable	1	2	3	4	5
1. Overall depression					
2. Cognitive-affective symptoms	0.99**				
3. Somatic symptoms	0.91**	0.84**			
4. Physical functioning	-0.32**	-0.30**	-0.37**		
5. Perceived family support	-0.50**	-0.51**	-0.42**	0.21**	
6. Perceived friend support	-0.45**	-0.45**	-0.40**	0.18**	0.39**

* $p < .05$, ** $p < .01$. p-values adjusted for multiple testing.

Table S3

Correlation Table for T3

Variable	1	2	3	4	5
1. Overall depression					
2. Cognitive-affective symptoms	0.99**				
3. Somatic symptoms	0.90**	0.82**			
4. Physical functioning	-0.24**	-0.20**	-0.31**		
5. Perceived family support	-0.45**	-0.45**	-0.40**	0.19**	
6. Perceived friend support	-0.37**	-0.37**	-0.30**	0.19**	0.41**

* $p < .05$, ** $p < .01$. p-values adjusted for multiple testing.

2. Goodness-of-Fit Statistics

Table S4

Goodness-of-Fit Statistics

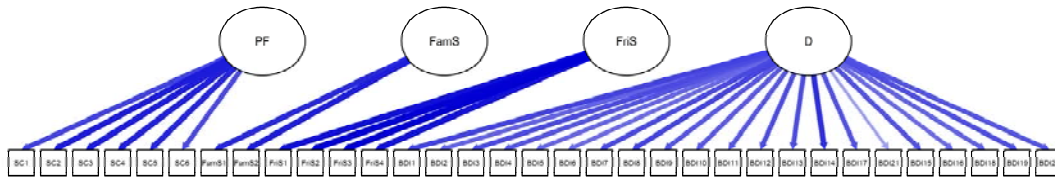
Models	χ^2	<i>df</i>	<i>AIC</i>	<i>BIC</i>	<i>RMSEA</i>	<i>TLI</i>
Model 1						
Model 1a: BDI total - saturated	8830.58	4885	66962.75	67653.94	.040 [.039, .042]	.86
Model 1b: BDI total - pruned	8855.94	4901	66956.11	67579.87	.040 [.039, .042]	.86
Model 2						
Model 2a: BDI sub-scales - saturated	8622.02	4867	66790.19	67557.25	.039 [.038, .041]	.86
Model 2b: BDI sub-scales - pruned	8663.54	4851	66775.71	67424.76	.039 [.038, .041]	.86

Note. RMSEA = root mean square error of approximation; TLI = Tucker–Lewis index; BIC = Bayes information criterion; χ^2_{diff} = statistic of the χ^2 difference test; df_{diff} = degrees of freedom of the χ^2 difference test; $p = p$ value of the χ^2 difference test.

3. Factor Models

Figure S1

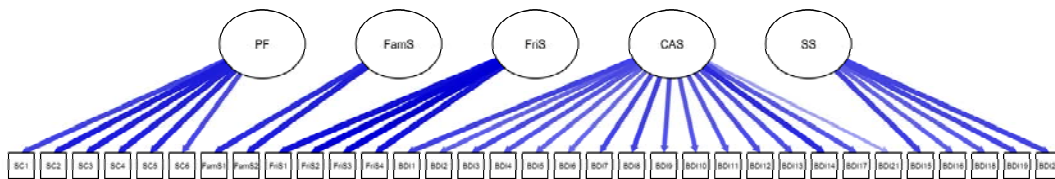
Factor Model for Model 1 (Overall Depression)



Note. D = overall depression, PF = physical functioning, FamS = family support, FriS = friend support.

Figure S2

Factor Model for Model 2 (Depression Subscales)

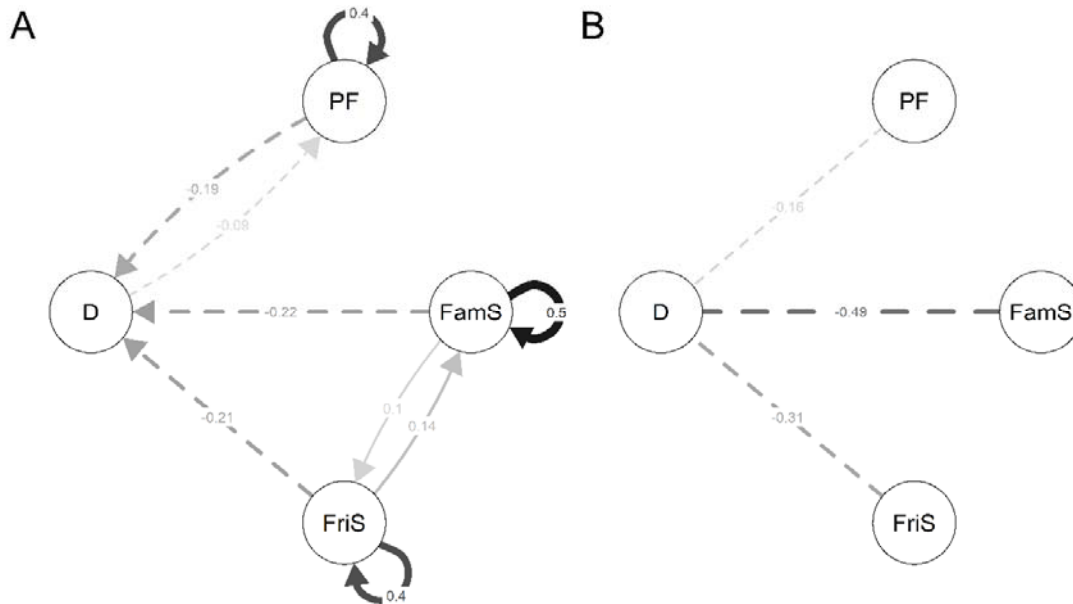


Note. CAS = cognitive-affective symptoms, SS = somatic symptoms, PF = physical functioning, FamS = family support, FriS = friend support.

4. Model 1

Figure S3

Temporal and contemporaneous network for overall depression and the resources



Note. A: temporal network, B: contemporaneous network. Straight edges = positive effects, dashed edges = negative effects. Thicker edges indicate stronger effects. D = overall depression, PF = physical functioning, FamS = family support, FriS = friend support.

Table S5

Stability Analysis for Model 1

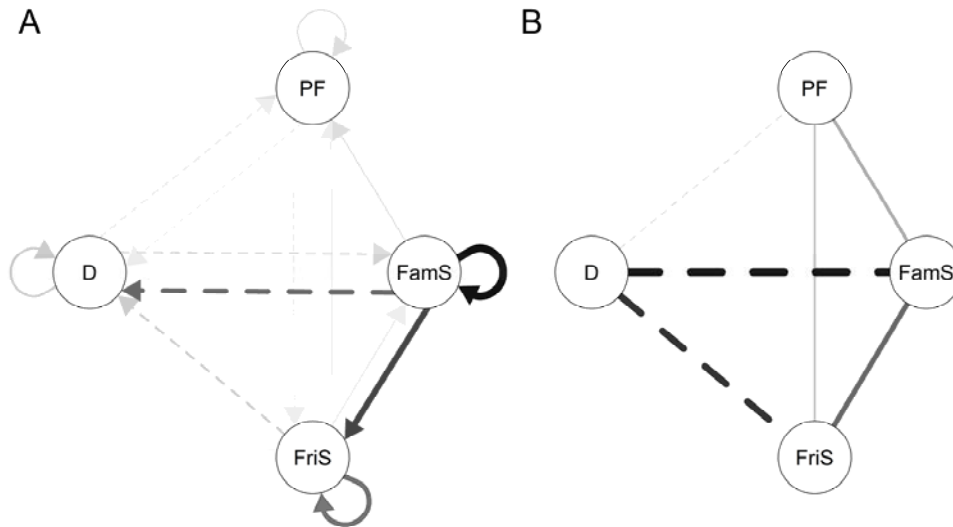
Variable	1	2	3	4
A Temporal network				
1. Physical functioning	31	0	18	17
2. Perceived family support	28	200	157	128
3. Perceived friend support	19	24	124	48
4. Overall depression	25	38	5	49
B Contemporaneous network				
1. Physical functioning	-			
2. Perceived family support	76	-		
3. Perceived friend support	51	130	-	
4. Overall depression	27	200	175	-

Note. Number of times (A) temporal and (B) contemporaneous effects were included in 200 networks that were based on random subsamples of 75% of the original sample. Bold-faced values indicate parameters that were included in the full sample.

5. Plots for Stability Analyses

Figure S4

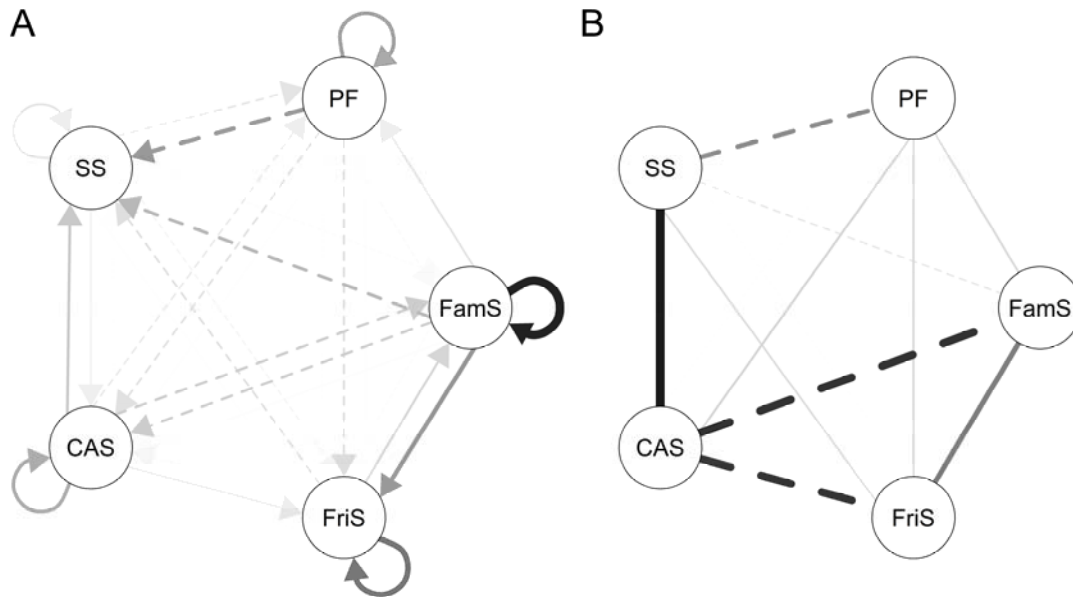
Stability Analysis for Model 1



Note. (A) temporal network, (B) contemporaneous network. Inclusion proportion of each edge based on a 25% case-drop bootstrap. Straight edges = positive effects, dashed edges = negative effects. Thicker edges indicate higher inclusion proportion.

Figure S5

Stability Analysis for Model 2



Note. (A) temporal network, (B) contemporaneous network. Inclusion proportion of each edge based on a 25% case-drop bootstrap. Straight edges = positive effects, dashed edges = negative effects. Thicker edges indicate higher inclusion proportion.

6. R-Code

```

library("qgraph")
library("dplyr")
library("devtools")
library("psych")
library("graphicalVAR")
library("psychometrics")
library("userfriendlyscience")
library("missForest")
library("semPlot")
library("ggplot2")
library("tidyr")

data <- read.csv("data.csv")

### missForest ###

data$Assessment <- factor(data$Assessment)

T1 <- subset(data, select = c(ID,
SF1512:SF1517,CYRM15,CYRM117,PeerSupp11:PeerSupp14,BDI11:BDI114,BDI117,BDI
121,BDI115,BDI116,BDI118,BDI119,BDI120))
T2 <- subset(data, Assessment == "1" | Assessment == "3", select = c(ID,
SF1522:SF1527,CYRM25,CYRM217,PeerSupp21:PeerSupp24,BDI21:BDI214,BDI217,BDI
221,BDI215,BDI216,BDI218,BDI219,BDI220))
T3 <- subset(data, Assessment == "2" | Assessment == "3", select = c(ID,
SF1532:SF1537,CYRM35,CYRM317,PeerSupp31:PeerSupp34,BDI31:BDI314,BDI317,BDI
321,BDI315,BDI316,BDI318,BDI319,BDI320))

T1 <- missForest(T1, verbose = TRUE)
T2 <- missForest(T2, verbose = TRUE)
T3 <- missForest(T3, verbose = TRUE)

T12 <- merge(T1$ximp, T2$ximp, by.x = "ID", all.x = TRUE)
dataCA <- merge(T12, T3$ximp, by.x = "ID", all.x = TRUE)
dataCA <- subset(dataCA, select =
c(SF1512:SF1517,CYRM15,CYRM117,PeerSupp11:PeerSupp14,BDI11:BDI114,BDI117,B
DI121,BDI115,BDI116,BDI118,BDI119,BDI120,
SF1522:SF1527,CYRM25,CYRM217,PeerSupp21:PeerSupp24,BDI21:BDI214,BDI217,BDI
221,BDI215,BDI216,BDI218,BDI219,BDI220,
SF1532:SF1537,CYRM35,CYRM317,PeerSupp31:PeerSupp34,BDI31:BDI314,BDI317,BDI
321,BDI315,BDI316,BDI318,BDI319,BDI320))

### Descriptives ###

t1 <- T1$ximp
t2 <- T2$ximp
t3 <- T3$ximp

```

```
t1$BDIT <- rowSums(t1[c(14:34)])
t1$BDICA <- rowSums(t1[c(14:29)])
t1$BDIS <- rowSums(t1[c(30:34)])
t1$self <- rowSums(t1[c(2:7)])
t1$family <- rowSums(t1[c(8:9)])
t1$friends <- rowSums(t1[c(10:13)])
```

```
t2$BDIT <- rowSums(t2[c(14:34)])
t2$BDICA <- rowSums(t2[c(14:29)])
t2$BDIS <- rowSums(t2[c(30:34)])
t2$self <- rowSums(t2[c(2:7)])
t2$family <- rowSums(t2[c(8:9)])
t2$friends <- rowSums(t2[c(10:13)])
```

```
t3$BDIT <- rowSums(t3[c(14:34)])
t3$BDICA <- rowSums(t3[c(14:29)])
t3$BDIS <- rowSums(t3[c(30:34)])
t3$self <- rowSums(t3[c(2:7)])
t3$family <- rowSums(t3[c(8:9)])
t3$friends <- rowSums(t3[c(10:13)])
```

```
describe(t1[,c(35:40)])
describe(t2[,c(35:40)])
describe(t3[,c(35:40)])
```

```
scaleReliability(dat = t1, items = c(14:34))
scaleReliability(dat = t2, items = c(14:34))
scaleReliability(dat = t3, items = c(14:34))
```

```
scaleReliability(dat = t1, items = c(14:29))
scaleReliability(dat = t2, items = c(14:29))
scaleReliability(dat = t3, items = c(14:29))
```

```
scaleReliability(dat = t1, items = c(30:34))
scaleReliability(dat = t2, items = c(30:34))
scaleReliability(dat = t3, items = c(30:34))
```

```
scaleReliability(dat = t1, items = c(2:7))
scaleReliability(dat = t2, items = c(2:7))
scaleReliability(dat = t3, items = c(2:7))
```

```
scaleReliability(dat = t1, items = c(8:9))
scaleReliability(dat = t2, items = c(8:9))
scaleReliability(dat = t3, items = c(8:9))
```

```
scaleReliability(dat = t1, items = c(10:13))
scaleReliability(dat = t2, items = c(10:13))
scaleReliability(dat = t3, items = c(10:13))
```



```

c("F1", "F2", "F3","F4","F5"))

factors2 <- c("Self","Family", "Friends", "BDI-CA","BDI-S")

# Estimation algorithm:
alpha <- 0.05
adjust <- "none"
searchstrategy <- "modelsearch"

## Form model: BDI total ##
model_BDI_Ts <- dlvm1(
  data = dataCA, # data frame
  vars = varMat, # The design matrix, with a row indicating a variable and a column a wave of
measurements. Note that NA indicates missing variables
  lambda = lambda1, # Measurement model
  within_latent = "ggm", # Model within-subject contemporaneous as GGM
  within_residual = "chol", # Model within-subject residuals as Cholesky
  between_latent = "ggm", # Model between-subject latent as GGM
  between_residual = "chol", # Model between-subject residuals as Cholesky
  estimator = "FIML", # Full-information max likelihood
  latents = factors1 # Names of the latent variables to use
)

model_BDI_Ts <- model_BDI_Ts %>% runmodel

# Fix the problematic parameters to zero:
diaglep <- diag(getmatrix(model_BDI_Ts, "lowertri_epsilon_between"))
if (any(abs(diaglep) < 1e-6)){
  for (i in which(abs(diaglep) < 1e-6)){
    model_BDI_Ts <- model_BDI_Ts %>%
      fixpar("lowertri_epsilon_between",i,i,value = 0)
  }
}

model_BDI_Ts <- model_BDI_Ts %>% runmodel

# Estimation algorithm (prune step):
model_BDI_Tp <- model_BDI_Ts %>%
  runmodel %>%
  prune(alpha = alpha, adjust = adjust, recursive = FALSE)

if (searchstrategy == "stepup"){
  model_BDI_Tp <- model_BDI_Tp %>% stepup(alpha = alpha, criterion = "bic")
} else if (searchstrategy == "modelsearch"){
  model_BDI_Tp <- model_BDI_Tp %>% modelsearch(prunealpha = alpha, addalpha =
alpha)
}

```

```

model_BDI_Ts %>% fit
model_BDI_Tp %>% fit

# significant edges #
id_BDI_Tp = c("Self", "Family", "Friends", "BDI")
p_BDI_Tp <- model_BDI_Tp@parameters[model_BDI_Tp@parameters$var1 %in%
id_BDI_Tp,c("var1","op","var2","est","p","matrix")]
p_BDI_Tp %>% mutate_if(is.numeric, ~round(., 3))
write.csv(p_BDI_Tp, file = "p_BDI_Tp.csv")

### FACTOR MODEL ###

# Standardize at the within-subject level

# Unstandardized factor loadings:
lambda_BDI_Tp <- getmatrix(model_BDI_Tp, "lambda")

# Residual variances:
theta_BDI_Tp <- getmatrix(model_BDI_Tp, "sigma_epsilon_within")

# Latent variance-covariance:
psi_BDI_Tp <- getmatrix(model_BDI_Tp, "sigma_zeta_within")

# Use semPlot to standardize:
library("semPlot")
semPlot_mod_BDI_Tp <- lisrelModel(LY = lambda_BDI_Tp, TE = theta_BDI_Tp, PS =
psi_BDI_Tp)
modMats_BDI_Tp <- modelMatrices(semPlot_mod_BDI_Tp, "Mplus")
lambdastd_BDI_Tp <- modMats_BDI_Tp$Lambda[[1]]$std

# Number of latents:
nLat_BDI_Tp <- ncol(lambdastd_BDI_Tp)

# Number of observed:
nObs_BDI_Tp <- nrow(lambdastd_BDI_Tp)

# Edgelist for graph:
Edgelist_BDI_Tp <- cbind(

c(col(lambdastd_BDI_Tp)),c(row(lambdastd_BDI_Tp))+ncol(lambdastd_BDI_Tp),c(lambdas
td_BDI_Tp)
)

# shape:
shape_BDI_Tp <- c(rep("ellipse",nLat_BDI_Tp),rep("rectangle",nObs_BDI_Tp))

# Size1:
size1_BDI_Tp <- c(rep(10,nLat_BDI_Tp),rep(3,nObs_BDI_Tp))

```

```

# Size2:
size2_BDI_Tp <- c(rep(8, nLat_BDI_Tp), rep(3, nObs_BDI_Tp))

# Edge connect points:
ECP_BDI_Tp <- Edgelist_BDI_Tp
ECP_BDI_Tp[,1] <- NA # 0.5*pi
ECP_BDI_Tp[,2] <- 0

# Manifest labels:
obs1 <- c(
  "PF1", " PF 2", " PF 3", " PF 4", " PF 5", " PF 6",
  "FamS1", "FamS2",
  "FriS1", "FriS2", "FriS3", "FriS4",
  "BDI1", "BDI2", "BDI3", "BDI4",
  "BDI5", "BDI6", "BDI7", "BDI8", "BDI9", "BDI10", "BDI11",
  "BDI12", "BDI13", "BDI14", "BDI17", "BDI21",
  "BDI15", "BDI16", "BDI18", "BDI19", "BDI20"
)

# Size of labels:
labelCex_BDI_Tp <- c(
  rep(1.5, nLat_BDI_Tp),
  rep(1, nObs_BDI_Tp)
)

# Starting layout:
Layout_FM_BDI_Tp <- rbind(
  cbind(
    seq(-1, 1, length=nLat_BDI_Tp),
    1
  ),
  cbind(
    seq(-2, 2, length=nObs_BDI_Tp+2)[-c(1, nObs_BDI_Tp+2)],
    0
  )
)

# Plot:
png("FM_BDI_Tp.png", width=15000, height=2500, res=600)
qgraph(Edgelist_BDI_Tp,
  shape = shape_BDI_Tp,
  vsize = size1_BDI_Tp,
  vsize2 = size2_BDI_Tp,
  layout = Layout_FM_BDI_Tp,
  mar = c(4, 1, 7, 1),
  edgeConnectPoints = ECP_BDI_Tp,
  labels = c(factors1, obs1),
  label.scale = FALSE,
  label.cex = labelCex_BDI_Tp,

```



```

    asize = 2,
    theme = "colorblind",
    width = 15,
    height = 3,
    cut = 0)
dev.off()

##### Form model: BDI subscales #####
model_BDI_SSs <- dlvm1(
  data = dataCA, # data frame
  vars = varMat, # The design matrix, with a row indicating a variable and a column a wave of
measurements. Note that NA indicates missing variables
  lambda = lambda2, # Measurement model
  within_latent = "ggm", # Model within-subject contemporaneous as GGM
  within_residual = "chol", # Model within-subject residuals as Cholesky
  between_latent = "ggm", # Model between-subject latent as GGM
  between_residual = "chol", # Model between-subject residuals as Cholesky
  estimator = "FIML", # Full-information max likelihood
  latents = factors2 # Names of the latent variables to use
)

model_BDI_SSs <- model_BDI_SSs %>% runmodel

# Fix the problematic parameters to zero:
diaglep <- diag(getmatrix(model_BDI_SSs, "lowertri_epsilon_between"))
if (any(abs(diaglep) < 1e-6)){
  for (i in which(abs(diaglep) < 1e-6)){
    model_BDI_SSs <- model_BDI_SSs %>%
      fixpar("lowertri_epsilon_between",i,i,value = 0)
  }
}

model_BDI_SSs <- model_BDI_SSs %>% runmodel

# Estimation algorithm (prune step):
model_BDI_SSp <- model_BDI_SSs %>%
  runmodel %>%
  prune(alpha = alpha, adjust = adjust, recursive = FALSE)

if (searchstrategy == "stepup"){
  model_BDI_SSp <- model_BDI_SSp %>% stepup(alpha = alpha, criterion = "bic")
} else if (searchstrategy == "modelsearch"){
  model_BDI_SSp <- model_BDI_SSp %>% modelsearch(prunealpha = alpha, addalpha =
alpha)
}

model_BDI_SSs %>% fit
model_BDI_SSp %>% fit

```

```

compare(model_BDI_SSs,model_BDI_SSp)

# significant edges #
id_BDI_SSp = c("Self", "Family", "Friends", "BDI-CA", "BDI-S")
p_BDI_SSp <- model_BDI_SSp@parameters[model_BDI_SSp@parameters$var1 %in%
id_BDI_SSp,c("var1", "op", "var2", "est", "p", "matrix")]
p_BDI_SSp %>% mutate_if(is.numeric, ~round(., 3))
write.csv(p_BDI_SSp, file = "p_BDI_SSp.csv")

#### FACTOR MODEL ####

# Standardize at the within-subject level

# Unstandardized factor loadings:
lambda_BDI_SSp <- getmatrix(model_BDI_SSp, "lambda")

# Residual variances:
theta_BDI_SSp <- getmatrix(model_BDI_SSp, "sigma_epsilon_within")

# Latent variance-covariance:
psi_BDI_SSp <- getmatrix(model_BDI_SSp, "sigma_zeta_within")

# Use semPlot to standardize:
library("semPlot")
semPlot_mod_BDI_SSp <- lisrelModel(LY = lambda_BDI_SSp, TE = theta_BDI_SSp, PS =
psi_BDI_SSp)
modMats_BDI_SSp <- modelMatrices(semPlot_mod_BDI_SSp, "Mplus")
lambdastd_BDI_SSp <- modMats_BDI_SSp$Lambda[[1]]$std

# Number of latents:
nLat_BDI_SSp <- ncol(lambdastd_BDI_SSp)

# Number of observed:
nObs_BDI_SSp <- nrow(lambdastd_BDI_SSp)

# Edgelist for graph:
Edgelist_BDI_SSp <- cbind(

c(col(lambdastd_BDI_SSp)),c(row(lambdastd_BDI_SSp))+ncol(lambdastd_BDI_SSp),c(lam
bdastd_BDI_SSp)
)

# shape:
shape_BDI_SSp <- c(rep("ellipse",nLat_BDI_SSp),rep("rectangle",nObs_BDI_SSp))

# Size1:
size1_BDI_SSp <- c(rep(10,nLat_BDI_SSp),rep(3,nObs_BDI_SSp))

# Size2:
size2_BDI_SSp <- c(rep(8,nLat_BDI_SSp),rep(3,nObs_BDI_SSp))

```

```

# Edge connect points:
ECP_BDI_SSp <- Edgelist_BDI_SSp
ECP_BDI_SSp[,1] <- NA # 0.5*pi
ECP_BDI_SSp[,2] <- 0

# Size of labels:
labelCex_BDI_SSp <- c(
  rep(1.5,nLat_BDI_SSp),
  rep(1,nObs_BDI_SSp)
)

# Starting layout:
Layout_FM_BDI_SSp <- rbind(
  cbind(
    seq(-1,1,length=nLat_BDI_SSp),
    1
  ),
  cbind(
    seq(-2,2,length=nObs_BDI_SSp+2)[-c(1,nObs_BDI_SSp+2)],
    0
  )
)

# Plot:
png("FM_BDI_SSp.png", width=15000, height=2500, res=600)
qgraph(Edgelist_BDI_SSp,
  shape = shape_BDI_SSp,
  vsize = size1_BDI_SSp,
  vsize2 = size2_BDI_SSp,
  layout = Layout_FM_BDI_SSp,
  mar = c(4,1,7,1),
  edgeConnectPoints = ECP_BDI_SSp,
  labels = c(factors2, obs1),
  label.scale = FALSE,
  label.cex = labelCex_BDI_SSp,
  asize = 2,
  theme = "colorblind",
  width = 15,
  height = 3,
  cut = 0)
dev.off()

```

```

### NETWORK MODELS ###
temporal_BDI_Tp <- getmatrix(model_BDI_Tp, "PDC")
contemporaneous_BDI_Tp <- getmatrix(model_BDI_Tp, "omega_zeta_within")
temporal_BDI_SSp <- getmatrix(model_BDI_SSp, "PDC")
contemporaneous_BDI_SSp <- getmatrix(model_BDI_SSp, "omega_zeta_within")

loopRotation <- rep(NA, 6)
loopRotation[3] <- 0.8 * pi

max_A <- max(c(abs(temporal_BDI_Tp),abs(temporal_BDI_SSp)))
max_B <- max(c(abs(contemporaneous_BDI_Tp),abs(contemporaneous_BDI_SSp)))

Layout1 <- structure(c(1, 2, 1, -1, 1.5, 0, -1.5, 0), .Dim = c(4L, 2L))
Layout2 <- structure(c(1, 2, 1, -1,-1, 1.5, 0, -1.5, -1, 1), .Dim = c(5L, 2L))

png("BDI_Tp.png", width=9000, height=5000, res=600)
par(mfrow=c(1,2))
qgraph(temporal_BDI_Tp, theme = "gray", vsize = 13, mar = rep(5,4), asize = 8,
        directed = TRUE, labels = factors1, esize = 10, edge.label.margin = 0.02,
        vTrans = 254, loopRotation = loopRotation, layout = Layout1, maximum = max_A,
        label.scale.equal = TRUE, title = "A", title.cex = 3, edge.labels = TRUE)
qgraph(contemporaneous_BDI_Tp, layout = Layout1, theme = "gray", vsize = 13, mar =
rep(5,4),
        labels = factors1, maximum = max_B, esize = 10, edge.labels = TRUE, edge.label.margin
= 0.02,
        vTrans = 254, loopRotation = loopRotation,
        label.scale.equal = TRUE, title = "B", title.cex = 3)
dev.off()

png("BDI_SSp.png", width=9000, height=5000, res=600)
par(mfrow=c(1,2))
qgraph(temporal_BDI_SSp, theme = "gray", vsize = 13, mar = rep(5,4), asize = 8,
        directed = TRUE, labels = factors2, esize = 10, edge.labels = TRUE, edge.label.margin =
0.02,
        vTrans = 254, loopRotation = loopRotation, layout = Layout2, maximum = max_A,
        label.scale.equal = TRUE, title = "A", title.cex = 3)
qgraph(contemporaneous_BDI_SSp, layout = Layout2, theme = "gray", vsize = 13, mar =
rep(5,4),
        labels = factors2, maximum = max_B, esize = 10, edge.labels = TRUE, edge.label.margin
= 0.02,
        vTrans = 254, loopRotation = loopRotation,
        label.scale.equal = TRUE, title = "B", title.cex = 3)
dev.off()

```

```

### BOOTSTRAP BDI_T ###
subDataCA <- na.omit(dataCA[,na.omit(c(varMat))])
nBoot <- 200

boot_BDI_Tp <- lapply(1:nBoot, function(x){
  message("Simulation: ",x)
  repeat{
    tryres <- try({
      repeat{
        library("psychometrics")
        library("dplyr")

        # Sample from the data:
        bootData <- subDataCA[sample(1:nrow(subDataCA), round(0.75 * nrow(subDataCA))),
]

        # Form model:
        mod_boot <- dlvm1(
          bootData,
          vars = varMat, # The design matrix, with a row indicating a variable and a column a
wave of measurements. Note that NA indicates missing variables
          lambda = lambda1, # Measurement model
          within_latent = "ggm", # Model within-subject contemporaneous as GGM
          within_residual = "chol", # Model within-subject residuals as Cholesky
          between_latent = "ggm", # Model between-subject latent as GGM
          between_residual = "chol", # Model between-subject residuals as Cholesky
          latents = factors1 # Names of the latent variables to use
        )

        # Run first time:
        mod_boot <- mod_boot %>% runmodel

        # Fix the problematic parameters to zero:
        diaglep <- diag(getmatrix(mod_boot, "lowertri_epsilon_between"))
        if (any(abs(diaglep) < 1e-6)){
          for (i in which(abs(diaglep) < 1e-6)){
            mod_boot <- mod_boot %>%
              fixpar("lowertri_epsilon_between",i,i,value = 0)
          }
        }

        # Run again:
        mod_boot <- mod_boot %>% runmodel

        mod_boot <- mod_boot %>%
          runmodel %>%
          prune(alpha = alpha, adjust = adjust, recursive = FALSE, matrices =
c("beta","omega_zeta_within","omega_zeta_between"))

        # Search strategy

```

```

if (searchstrategy == "stepup"){
  mod_boot <- mod_boot %>% stepup(alpha = alpha, criterion = "bic")
} else if (searchstrategy == "modelsearch"){
  mod_boot <- mod_boot %>% modelsearch(prunealpha = alpha, addalpha = alpha)
}

# Sometimes the analysis results in a non positive-definite result. In this case, try again:
if (all(eigen(mod_boot@information)$values > -sqrt(.Machine$double.eps))){
  break
}
}
})
# Sometimes the analysis results in an error, in that case try again
if (!is(tryres, "try-error")){
  break
}
}
return(mod_boot)
})

# Bootstrap inclusions probabilities - NETWORK MODELS #

tempInclude_BDIp <- signInclude(boot_BDIp, "beta", transpose = TRUE)

contInclude_BDIp <- signInclude(boot_BDIp, "omega_zeta_within", lowertri = TRUE)

png("BDI_Tp_b.png", width=9000, height=5000, res=600)
par(mfrow=c(1,2))
qgraph(tempInclude_BDIp[,1:3], layout = Layout1, gray = TRUE, vsize = 13, mar =
rep(5,4), asize = 8,
  directed = TRUE, labels = factors1, maximum = 1, esize = 10, edge.labels = FALSE,
  lty = ifelse(tempInclude_BDIp$type=="pos",1,2),
  vTrans = 254, loopRotation = loopRotation, diag = TRUE,
  label.scale.equal = TRUE, parallelEdge = TRUE, title = "A", title.cex = 3)
qgraph(contInclude_BDIp[,1:3], layout = Layout1, gray = TRUE, vsize = 13, mar =
rep(5,4),
  labels = factors1, maximum = 1, esize = 10, edge.labels = FALSE,
  lty = ifelse(contInclude_BDIp$type=="pos",1,2),
  vTrans = 254, loopRotation = loopRotation, directed = FALSE,
  label.scale.equal = TRUE, parallelEdge = TRUE, title = "B", title.cex = 3)
dev.off()

### Make tables ###
### Parameter estimates ###
# Sanitize function:

# Temporal:
temporal_table_BDI_Tp <- temporal_BDI_Tp
temporal_table_BDI_Tp[temporal_table_BDI_Tp==0] <- NA
rownames(temporal_table_BDI_Tp) <- colnames(temporal_table_BDI_Tp) <- factors1
write.csv(temporal_table_BDI_Tp, file = "tt_BDI_Tp.csv")

```

```

# Contemporaneous:
contemporaneous_table_BDI_Tp <- contemporaneous_BDI_Tp

# Add marginal correlations:
contemporaneous_cors_BDI_Tp <- cov2cor(getmatrix(model_BDI_Tp,
"sigma_zeta_within"))

# Add to upper tri:
contemporaneous_table_BDI_Tp[upper.tri(contemporaneous_table_BDI_Tp)] <-
contemporaneous_cors_BDI_Tp[upper.tri(contemporaneous_cors_BDI_Tp)]

# Make zeroes NA:
contemporaneous_table_BDI_Tp[contemporaneous_table_BDI_Tp==0] <- NA
rownames(contemporaneous_table_BDI_Tp) <- colnames(contemporaneous_table_BDI_Tp)
<- factors1
write.csv(contemporaneous_table_BDI_Tp, file = "ct_BDI_Tp.csv")

### Bootstrap results estimates ###
# Boldify function:
boldify <- function(x,ind){
  x[ind] <- ifelse(is.na(x[ind]),NA,paste0("\\textbf{'",x[ind],''}"))
  x
}

# Inclusion over edges (disregard sign):
tempInclude_noSign_BDI_Tp <- edgeInclude(boot_BDI_Tp,"beta",TRUE)
# Temporal:
rownames(tempInclude_noSign_BDI_Tp) <- colnames(tempInclude_noSign_BDI_Tp) <-
factors1
write.csv(tempInclude_noSign_BDI_Tp, file = "btt_BDI_Tp.csv")

# Contemporaneous (lowertri)
# Inclusion over edges (disregard sign):
contInclude_noSign_BDI_Tp <- edgeInclude(boot_BDI_Tp,"omega_zeta_within")
rownames(contInclude_noSign_BDI_Tp) <- colnames(contInclude_noSign_BDI_Tp) <-
factors1
contInclude_noSign_BDI_Tp[upper.tri(contInclude_noSign_BDI_Tp,diag=TRUE)] <- NA
write.csv(contInclude_noSign_BDI_Tp, file = "bct_BDI_Tp.csv")

### BOOTSTRAP BDI_SS###
boot_BDISSp <- lapply(1:nBoot, function(x){
  message("Simulation: ",x)
  repeat{
    tryres <- try({
      repeat{

```

```

library("psychometrics")
library("dplyr")

# Sample from the data:
bootData <- subDataCA[sample(1:nrow(subDataCA), round(0.75 * nrow(subDataCA))),
]

# Form model:
mod_boot <- dlvm1(
  bootData,
  vars = varMat, # The design matrix, with a row indicating a variable and a column a
wave of measurements. Note that NA indicates missing variables
  lambda = lambda2, # Measurement model
  within_latent = "ggm", # Model within-subject contemporaneous as GGM
  within_residual = "chol", # Model within-subject residuals as Cholesky
  between_latent = "ggm", # Model between-subject latent as GGM
  between_residual = "chol", # Model between-subject residuals as Cholesky
  latents = factors2 # Names of the latent variables to use
)

# Run first time:
mod_boot <- mod_boot %>% runmodel

# Fix the problematic parameters to zero:
diaglep <- diag(getmatrix(mod_boot, "lowertri_epsilon_between"))
if (any(abs(diaglep) < 1e-6)){
  for (i in which(abs(diaglep) < 1e-6)){
    mod_boot <- mod_boot %>%
      fixpar("lowertri_epsilon_between",i,i,value = 0)
  }
}

# Run again:
mod_boot <- mod_boot %>% runmodel

mod_boot <- mod_boot %>%
  runmodel %>%
  prune(alpha = alpha, adjust = adjust, recursive = FALSE, matrices =
c("beta","omega_zeta_within","omega_zeta_between"))

# Search strategy
if (searchstrategy == "stepup"){
  mod_boot <- mod_boot %>% stepup(alpha = alpha, criterion = "bic")
} else if (searchstrategy == "modelsearch"){
  mod_boot <- mod_boot %>% modelsearch(prunealpha = alpha, addalpha = alpha)
}

# Sometimes the analysis results in a non positive-definite result. In this case, try again:
if (all(eigen(mod_boot@information)$values > -sqrt(.Machine$double.eps))){
  break
}

```



```

    }
  })
  # Sometimes the analysis results in an error, in that case try again
  if (!is(tryres, "try-error")){
    break
  }
}
return(mod_boot)
})

# Bootstrap inclusions probabilities - NETWORK MODELS #
tempInclude_BDISSp <- signInclude(boot_BDISSp, "beta", transpose = TRUE)

contInclude_BDISSp <- signInclude(boot_BDISSp, "omega_zeta_within", lowertri = TRUE)

png("BDI_SSp_b.png", width=9000, height=5000, res=600)
par(mfrow=c(1,2))
qgraph(tempInclude_BDISSp[,1:3], layout = Layout2, gray = TRUE, vsize = 13, mar =
rep(5,4), asize = 8,
  directed = TRUE, labels = factors2, maximum = 1, esize = 10, edge.labels = FALSE,
  lty = ifelse(tempInclude_BDISSp$type=="pos",1,2),
  vTrans = 254, loopRotation = loopRotation, diag = TRUE,
  label.scale.equal = TRUE, parallelEdge = TRUE, title = "A", title.cex = 3)
qgraph(contInclude_BDISSp[,1:3], layout = Layout2, gray = TRUE, vsize = 13, mar =
rep(5,4),
  labels = factors2, maximum = 1, esize = 10, edge.labels = FALSE,
  lty = ifelse(contInclude_BDISSp$type=="pos",1,2),
  vTrans = 254, loopRotation = loopRotation, directed = FALSE,
  label.scale.equal = TRUE, parallelEdge = TRUE, title = "B", title.cex = 3)
dev.off()

qgraph(contInclude_BDItp[,1:3], layout = Layout1, gray = TRUE, vsize = 13, mar =
rep(5,4),
  labels = factors1, maximum = 1, esize = 10, edge.labels = FALSE,
  lty = ifelse(contInclude_BDISSp$type=="pos",1,2),
  vTrans = 254, loopRotation = loopRotation, directed = FALSE,
  label.scale.equal = TRUE, parallelEdge = TRUE, title = "B", title.cex = 3)

### Make tables ###
### Parameter estimates ###
# Sanitize function:

# Temporal:
temporal_table_BDI_SSp <- temporal_BDI_SSp
temporal_table_BDI_SSp[temporal_table_BDI_SSp==0] <- NA
rownames(temporal_table_BDI_SSp) <- colnames(temporal_table_BDI_SSp) <- factors2
write.csv(temporal_table_BDI_SSp, file = "tt_BDI_SSp.csv")

# Contemporaneous:

```

```

contemporaneous_table_BDI_SSp <- contemporaneous_BDI_SSp

# Add marginal correlations:
contemporaneous_cors_BDI_SSp <- cov2cor(getmatrix(model_BDI_SSp,
"sigma_zeta_within"))

# Add to upper tri:
contemporaneous_table_BDI_SSp[upper.tri(contemporaneous_table_BDI_SSp)] <-
contemporaneous_cors_BDI_SSp[upper.tri(contemporaneous_cors_BDI_SSp)]

# Make zeroes NA:
contemporaneous_table_BDI_SSp[contemporaneous_table_BDI_SSp==0] <- NA
rownames(contemporaneous_table_BDI_SSp) <-
colnames(contemporaneous_table_BDI_SSp) <- factors2
write.csv(contemporaneous_table_BDI_SSp, file = "ct_BDI_SSp.csv")

### Bootstrap results estimates ###

# Inclusion over edges (disregard sign):
tempInclude_noSign_BDI_SSp <- edgeInclude(boot_BDI_SSp,"beta",TRUE)
# Temporal:
rownames(tempInclude_noSign_BDI_SSp) <- colnames(tempInclude_noSign_BDI_SSp) <-
factors2
write.csv(tempInclude_noSign_BDI_SSp, file = "bt_BDI_SSp.csv")

# Contemporaneous (lowertri)
# Inclusion over edges (disregard sign):
contInclude_noSign_BDI_SSp <- edgeInclude(boot_BDI_SSp,"omega_zeta_within")
rownames(contInclude_noSign_BDI_SSp) <- colnames(contInclude_noSign_BDI_SSp) <-
factors2
contInclude_noSign_BDI_SSp[upper.tri(contInclude_noSign_BDI_SSp,diag=TRUE)] <- NA
write.csv(contInclude_noSign_BDI_SSp, file = "bct_BDI_SSp.csv")

save.image("D:\Arbeitslaptop HP\Arbeit\RRC Time\RYSE\master
dataset\Analysis\LNA\CA\workspace.RData")

```