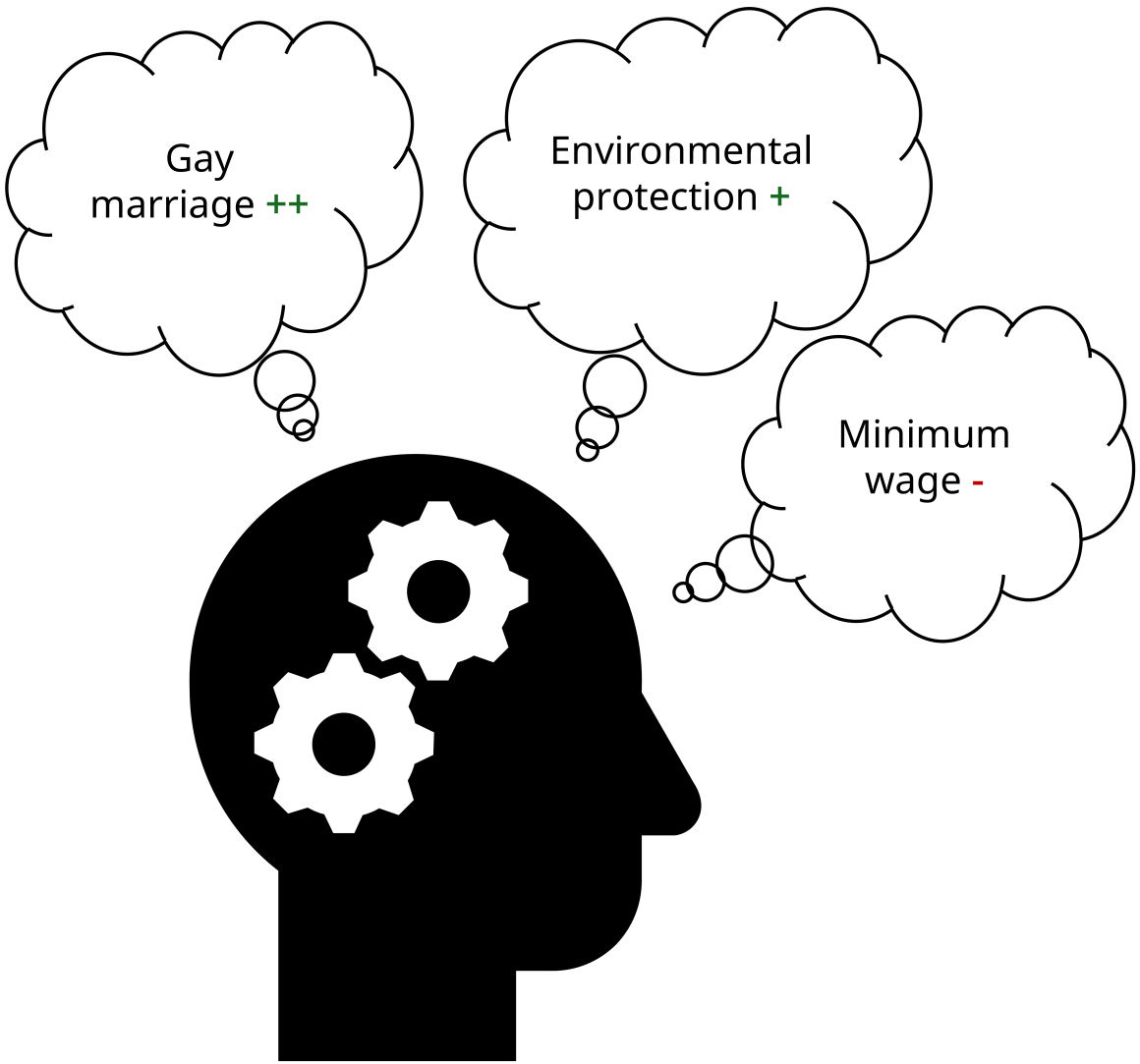
ResIN Workshop 2026

Philip Warncke, Dino Carpentras, Adrian Lüders

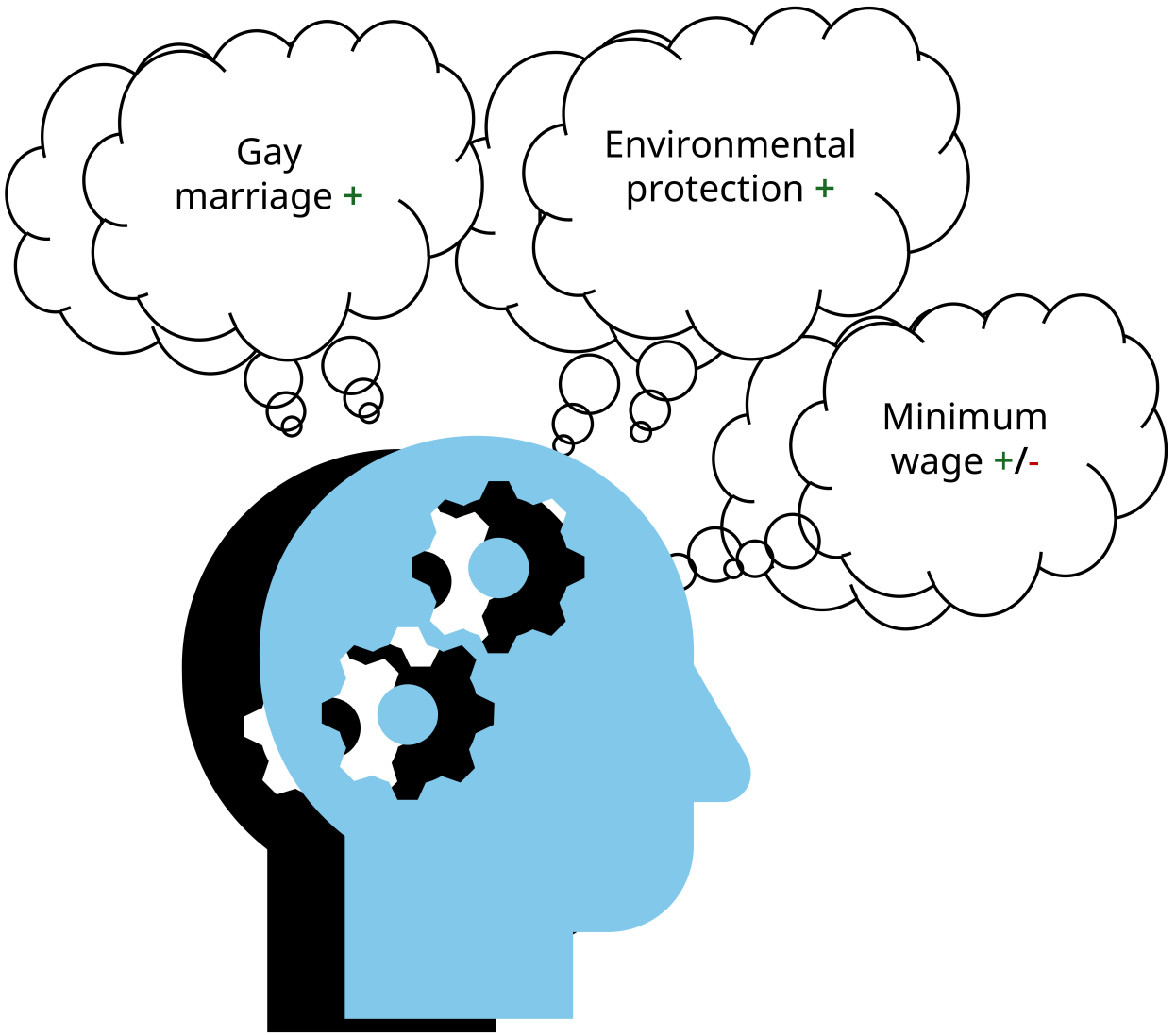
What's on the menu?

- Refresher on belief network modeling
  - Belief network analysis
  - Response Item Networks (ResIN)
- The ResIN package for R
  - Getting started
  - Simple example based on Lüders et.al. (2024)
  - Network cluster detection with the ResIN package
  - Statistical inference with the ResIN package
  - multimodal ResIN
- More applications and resources

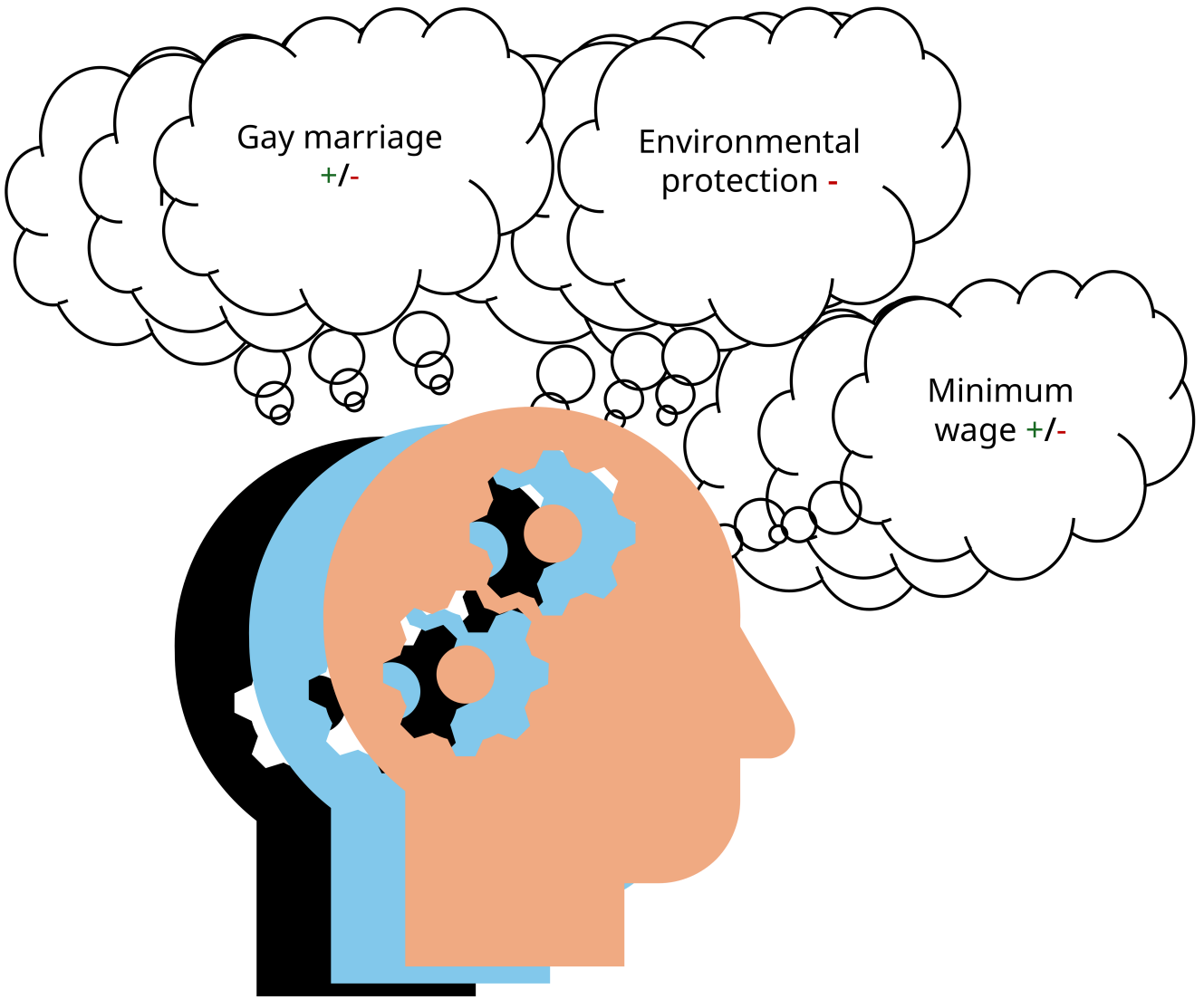
Belief network analysis



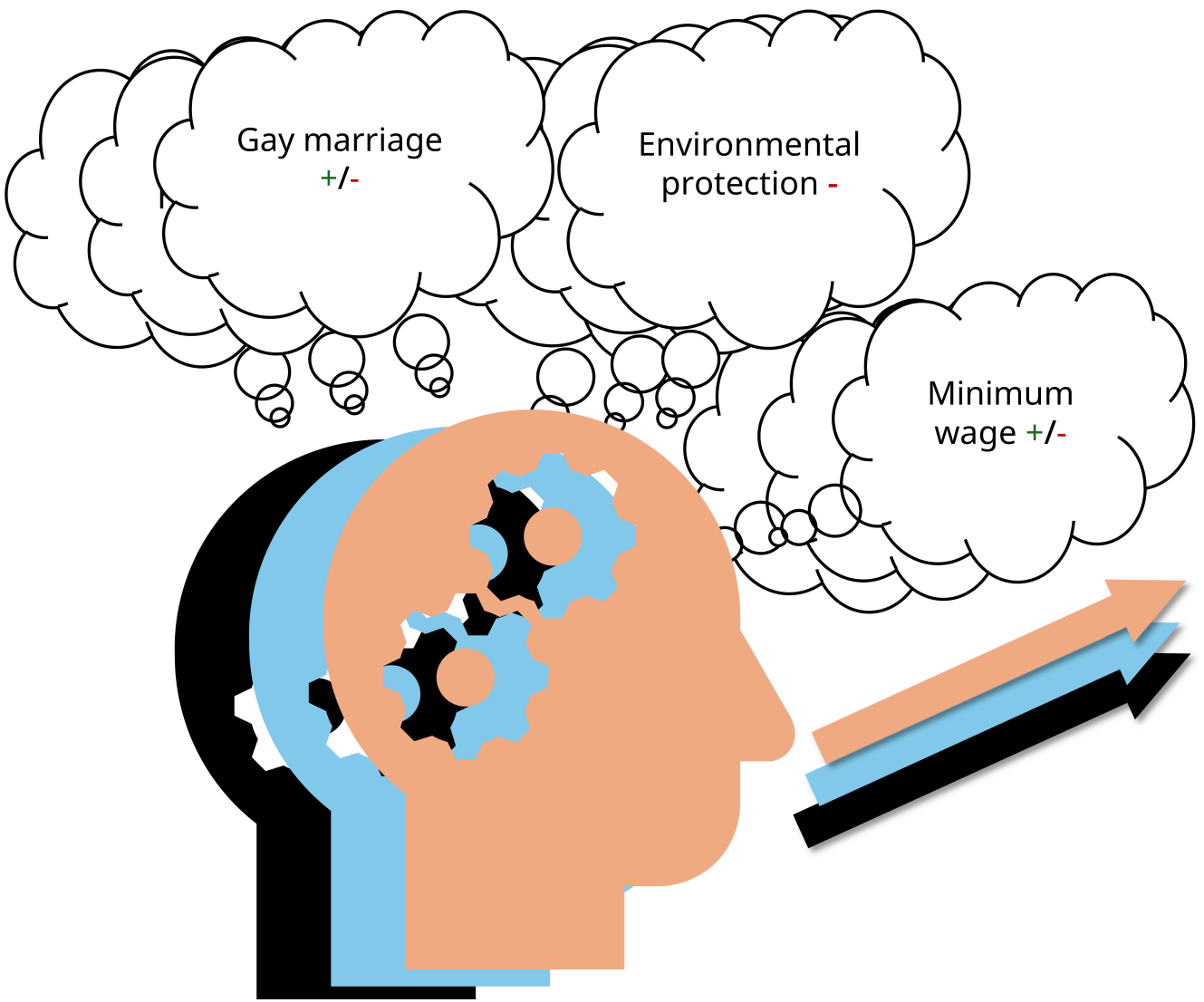
Belief network analysis

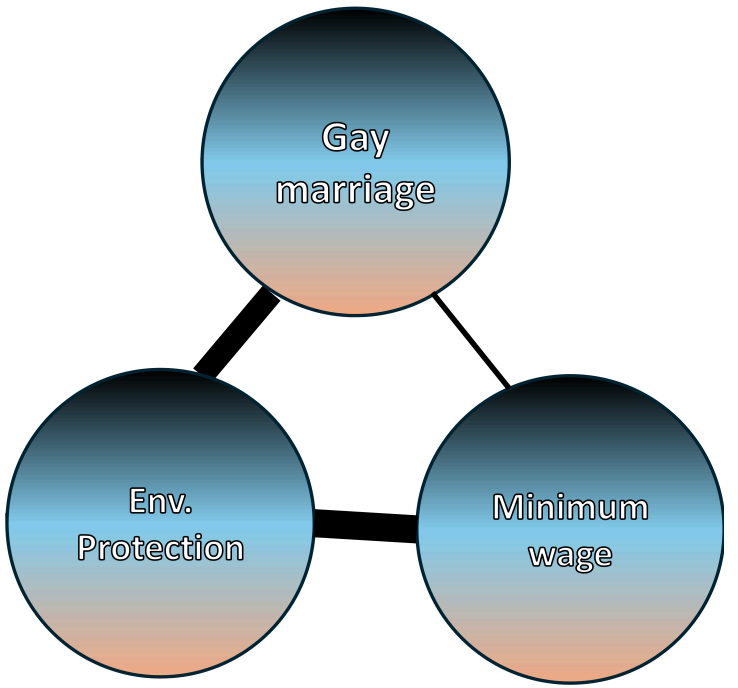


Belief network analysis

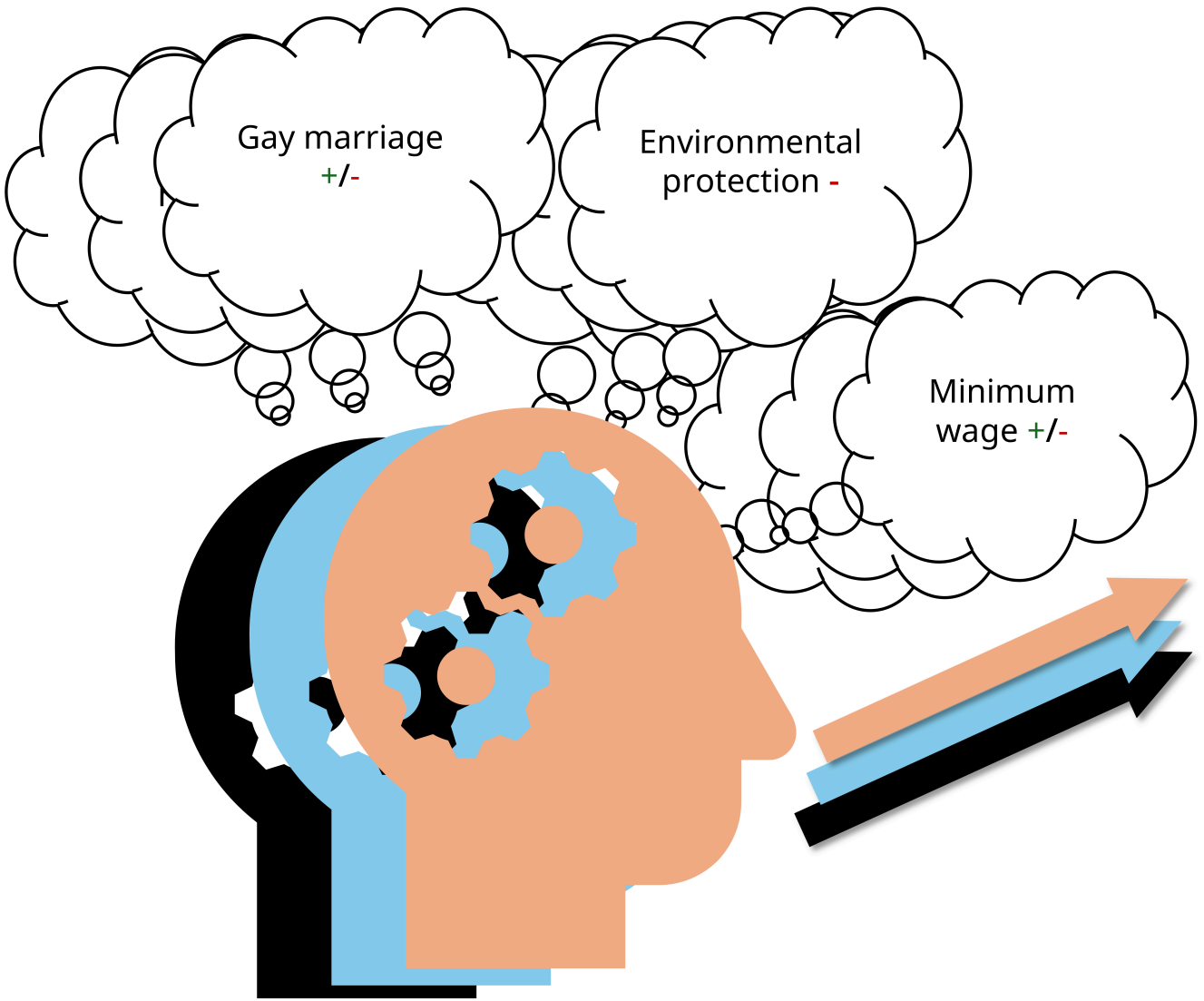


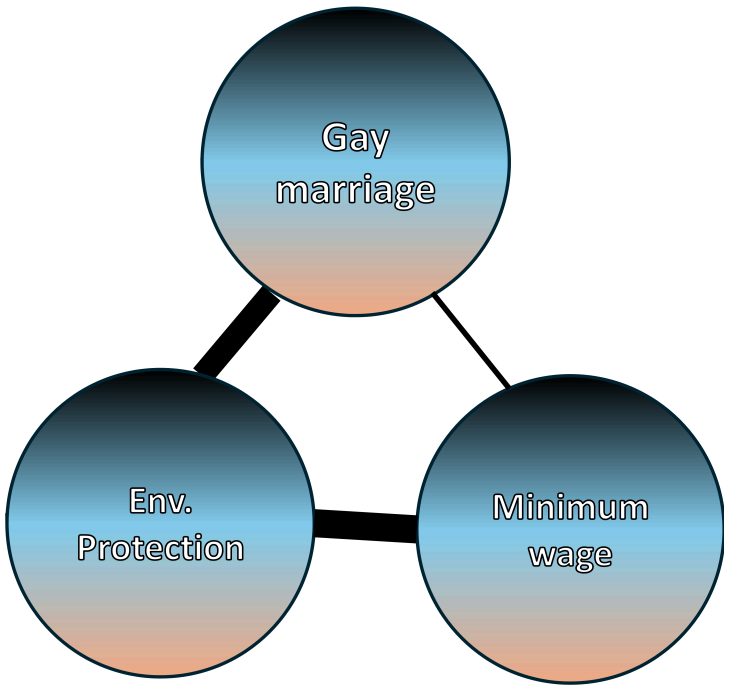
Belief network analysis





Belief network analysis

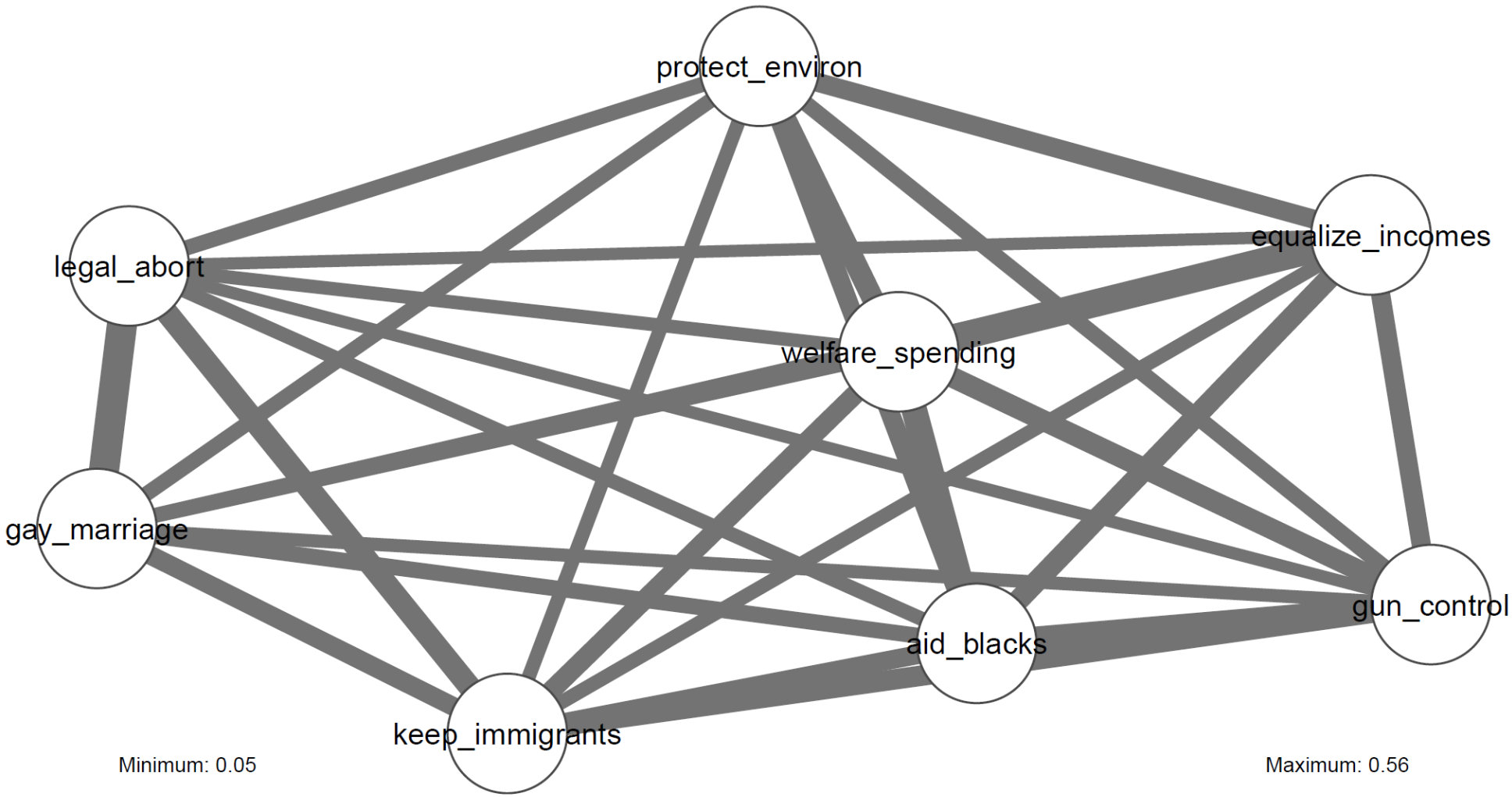




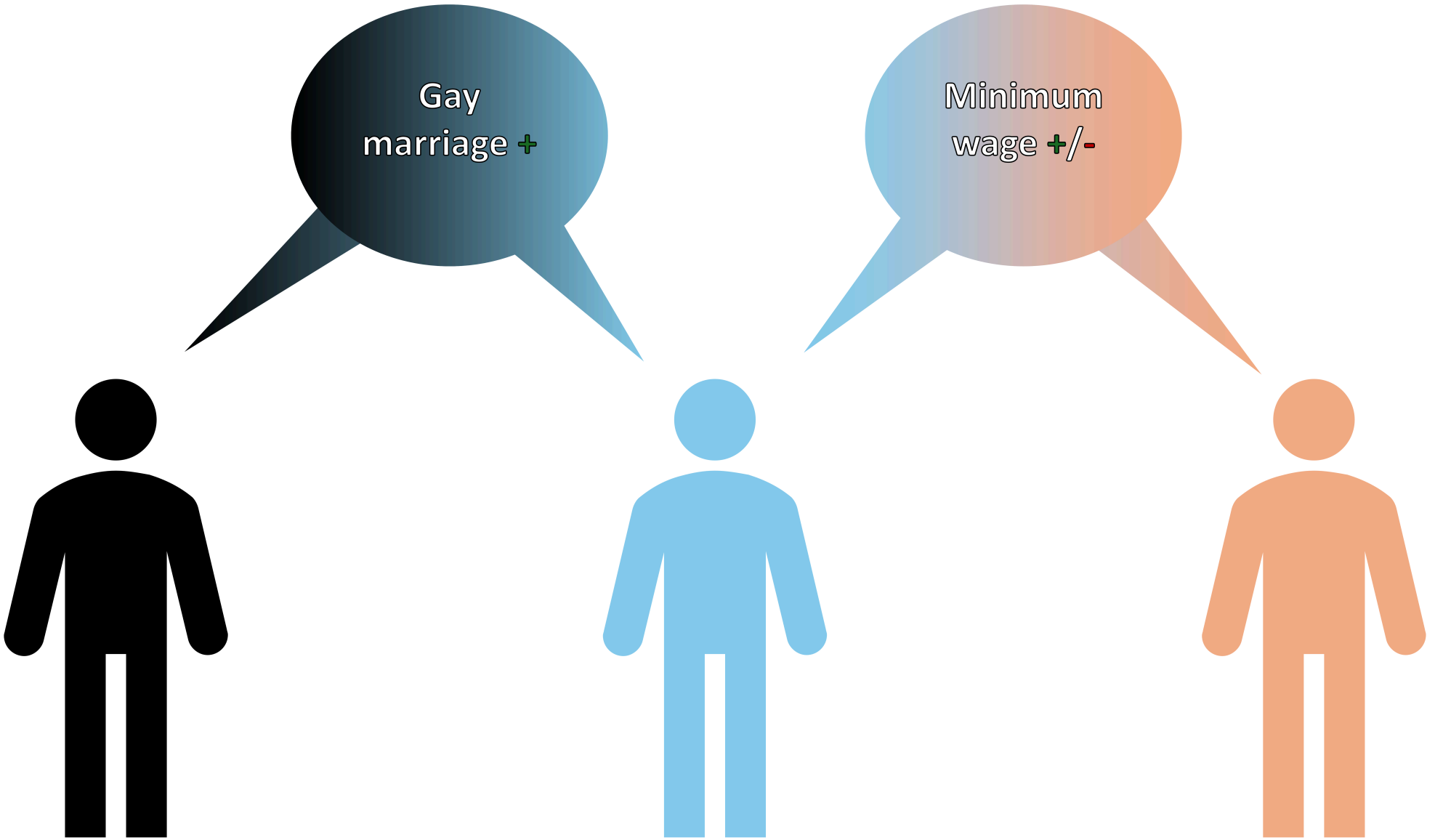
Classic Belief Network Analysis

e.g., Boutyline & Vaisey (2017)

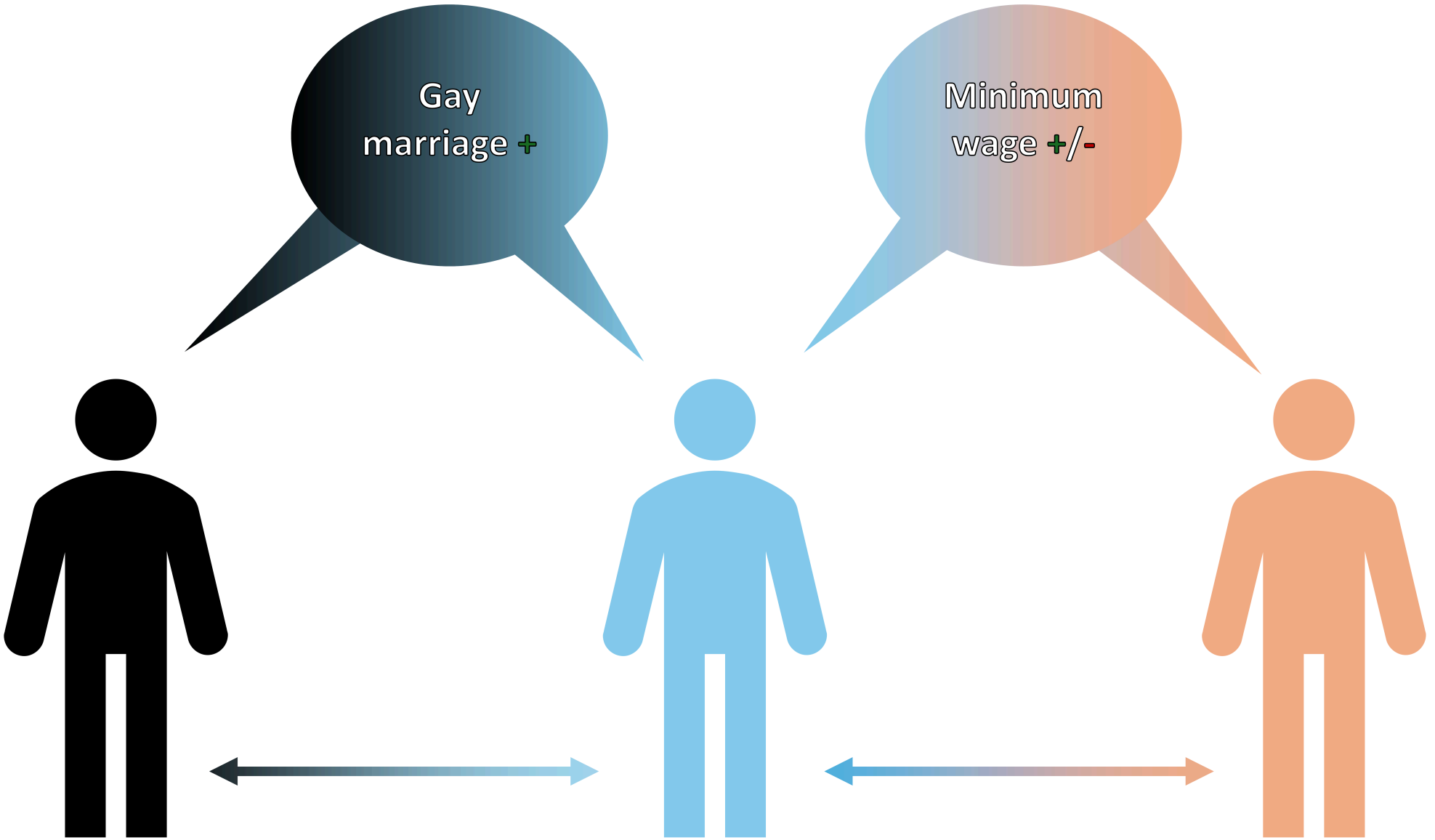
BNA example: Lüders et.al. 2023



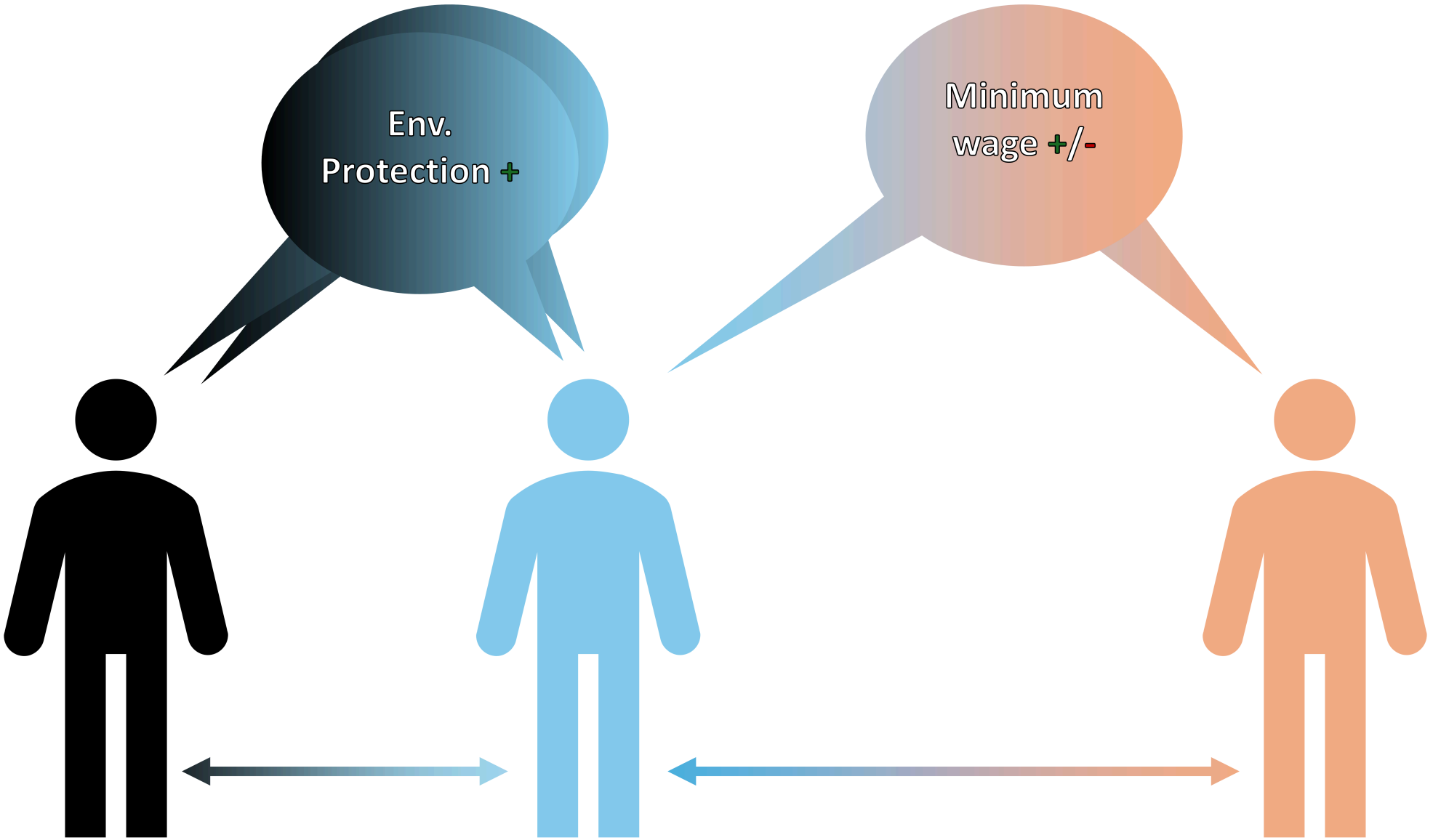
Response Item Networks (ResIN)



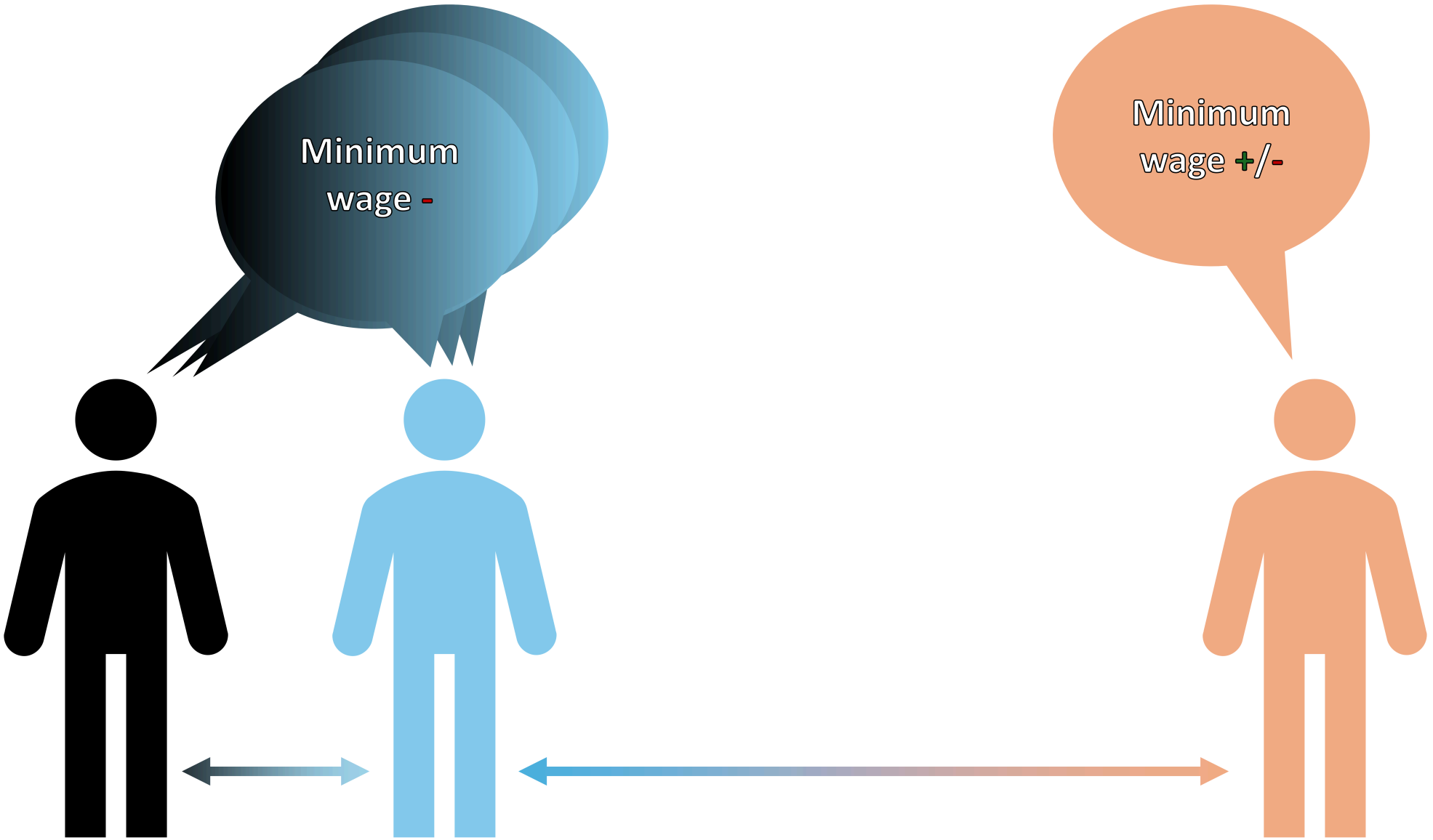
Response Item Networks (ResIN)



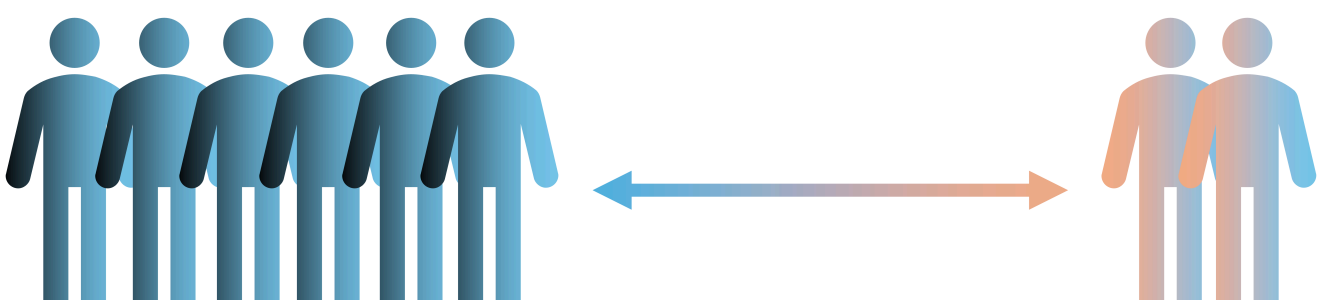
Response Item Networks (ResIN)



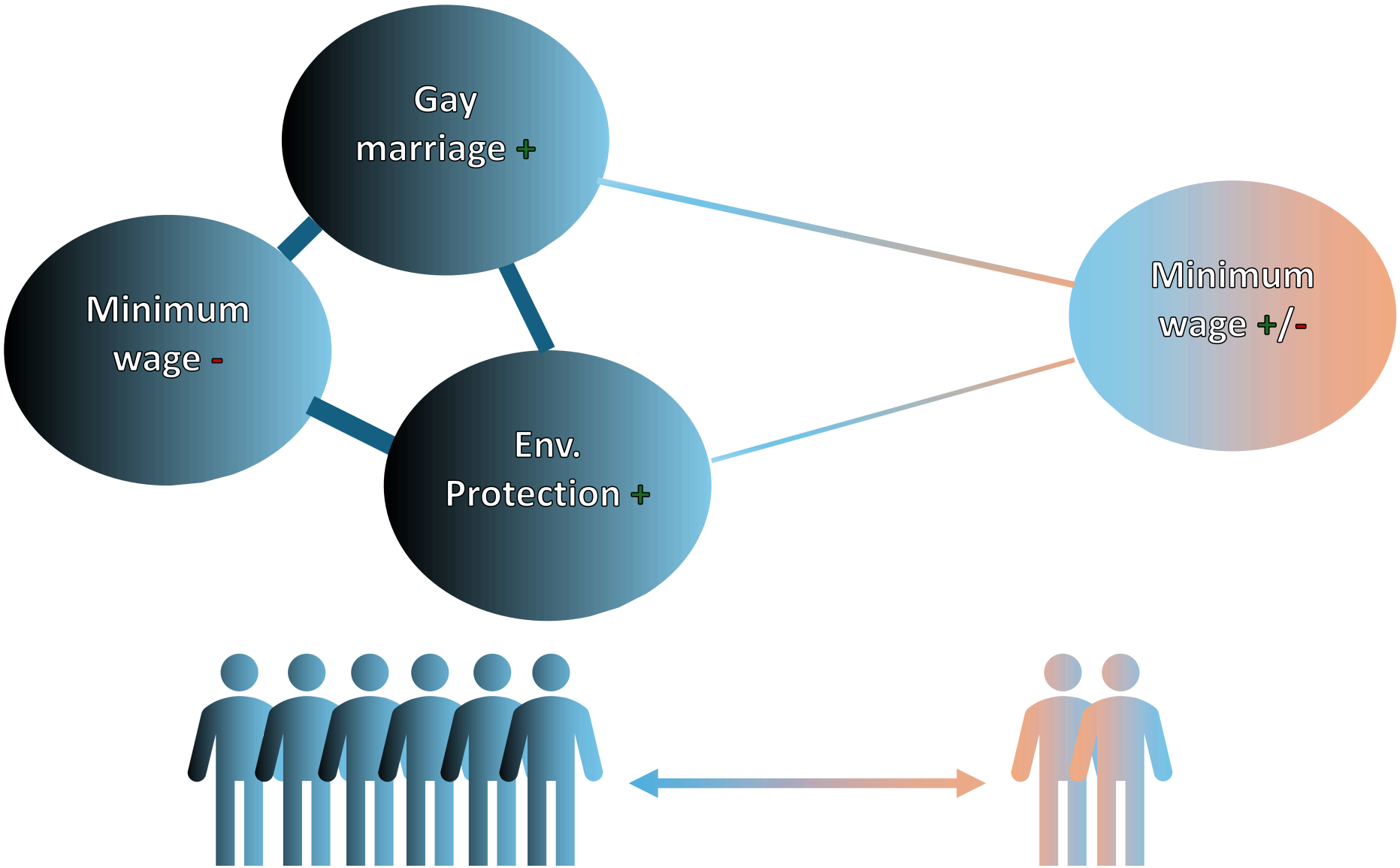
Response Item Networks (ResIN)



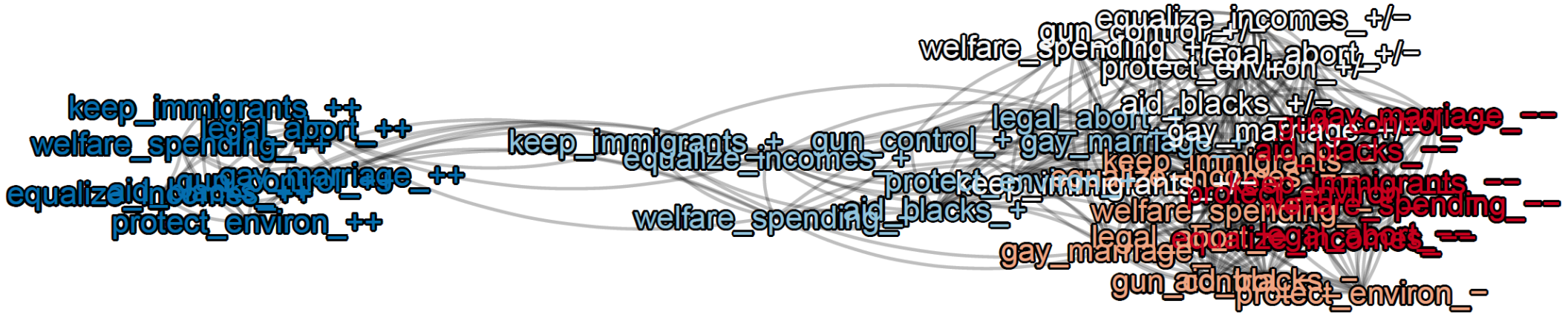
Response Item Networks (ResIN)



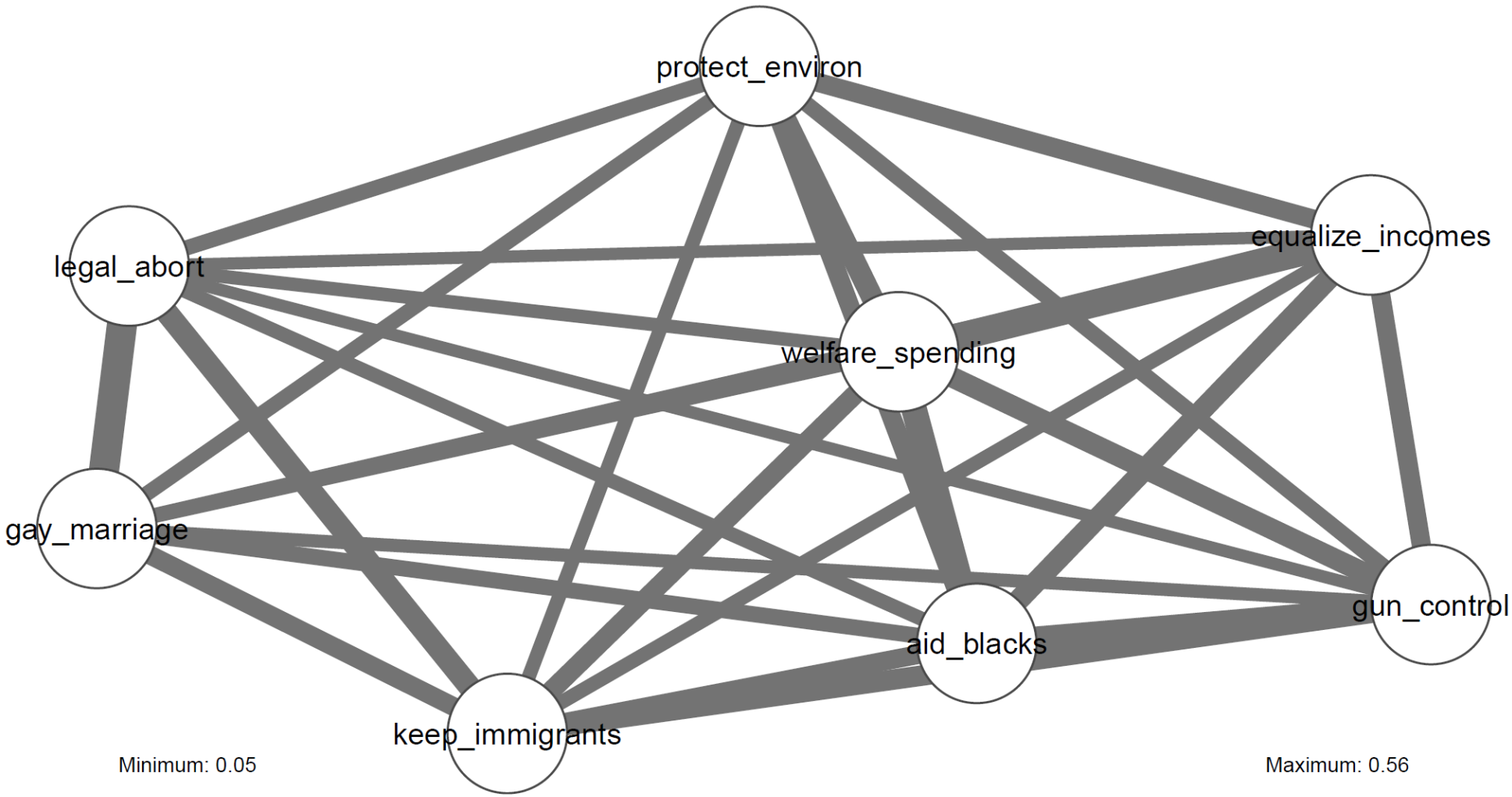
Response Item Networks (ResIN)



ResIN example: Lüders et.al. 2023



BNA example: Lüders et.al. 2023



Structural network differences

	BNA	ResIN
Nodes (verticies)	Items (e.g., ‘’CO2 taxes”,“gay marriage”,“gun control”)	Item responses (e.g.,‘’favoring CO2 taxes”,“neutral on CO2 taxes”,`”against CO2 taxes”, etc.)

	BNA	ResIN
Edges (ties)	Absolute value of partial correlations between items	Marginal correlations between responses across different items (positive corrs only)
Spatial location (FD plotting)	Determined by attitude centrality (sort of)	Interpretable as socio-political (or ideological) distance

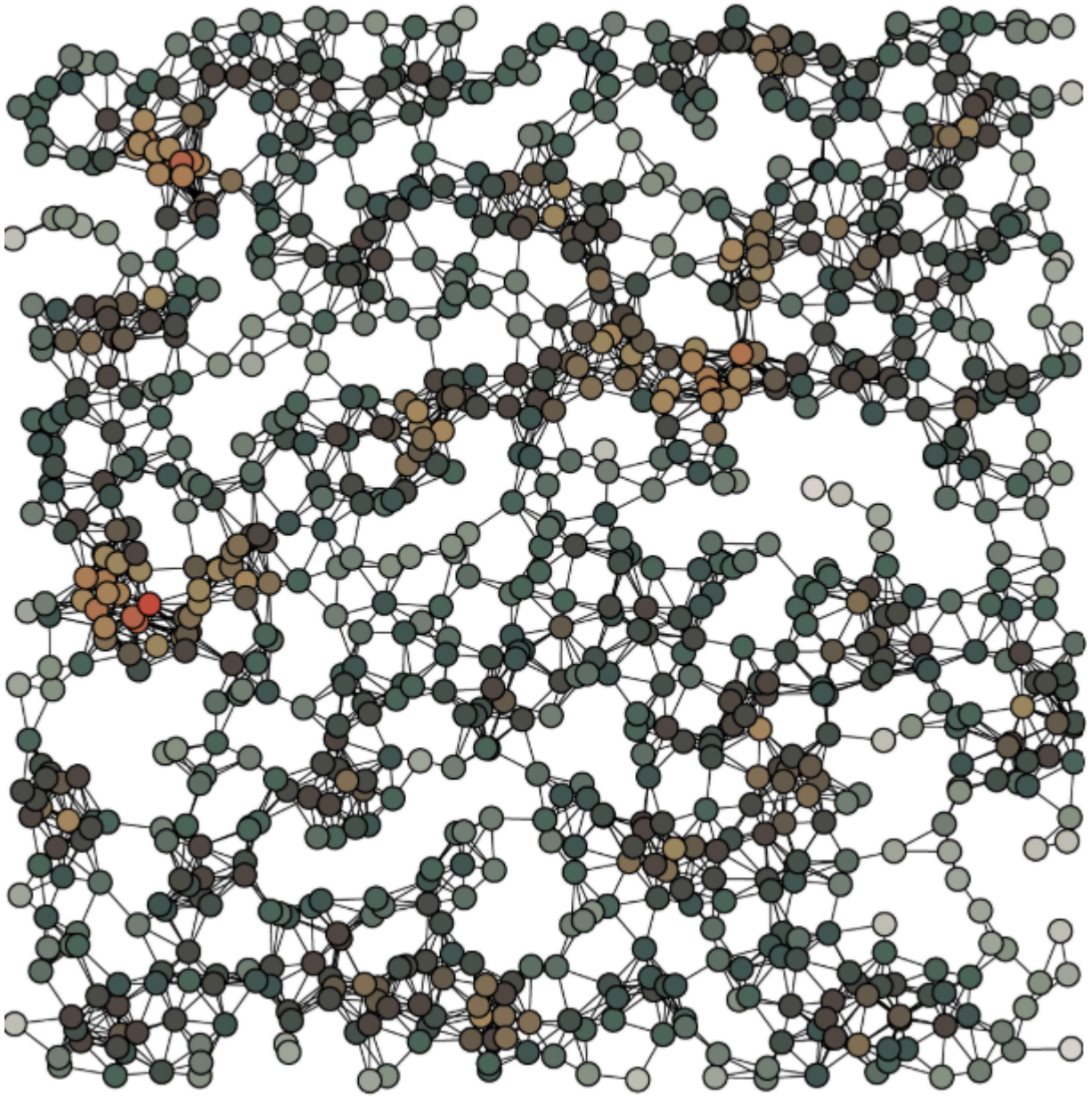
Network centrality

- Centrality is indicative of a node's function, importance, and/or redundancy in a network
- Just as mean, median, and mode give different accounts for the "center" of a variable - networks statistics provide for different "flavors" of node centrality
  - Strength, betweenness, closeness

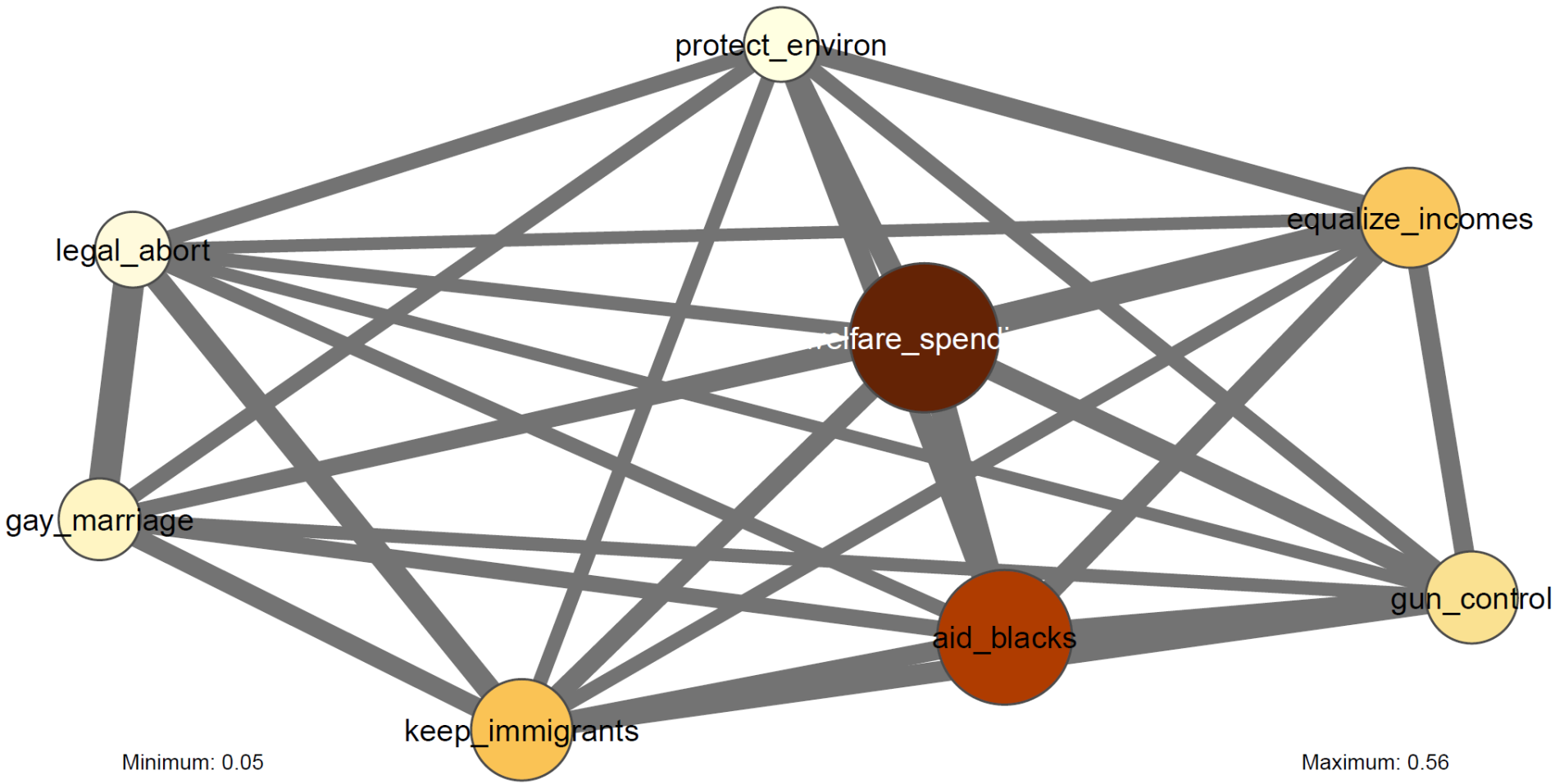
Network centrality

- Strength centrality captures the importance of immediate connectivity
  - Average of all (potential) edges connected to a node (weighted version of degree centrality)

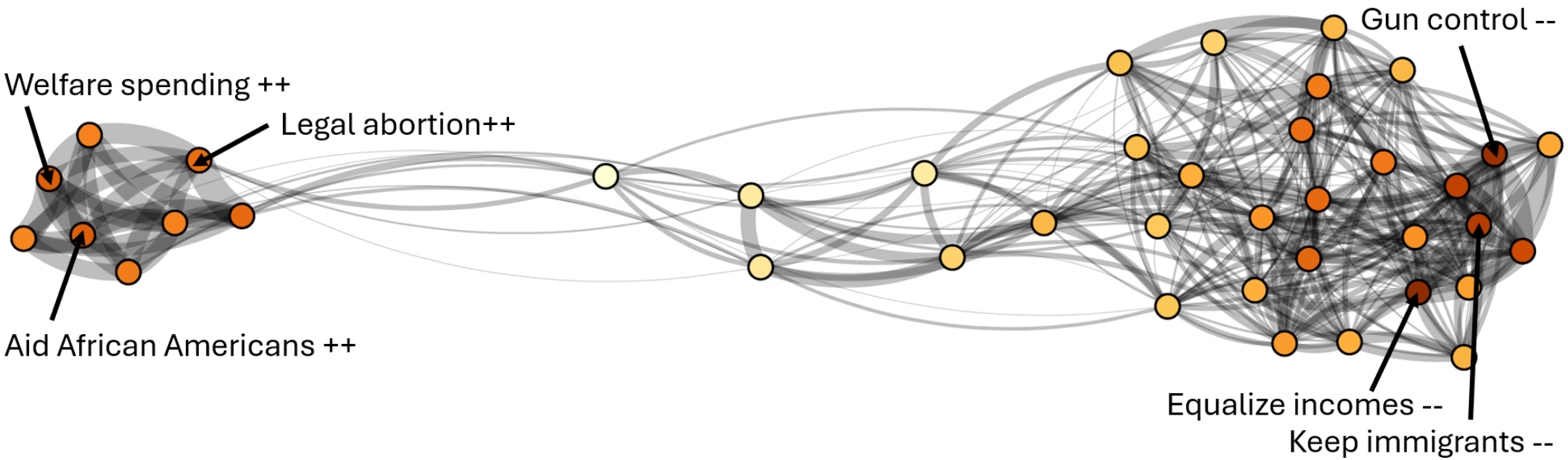
Strength centrality



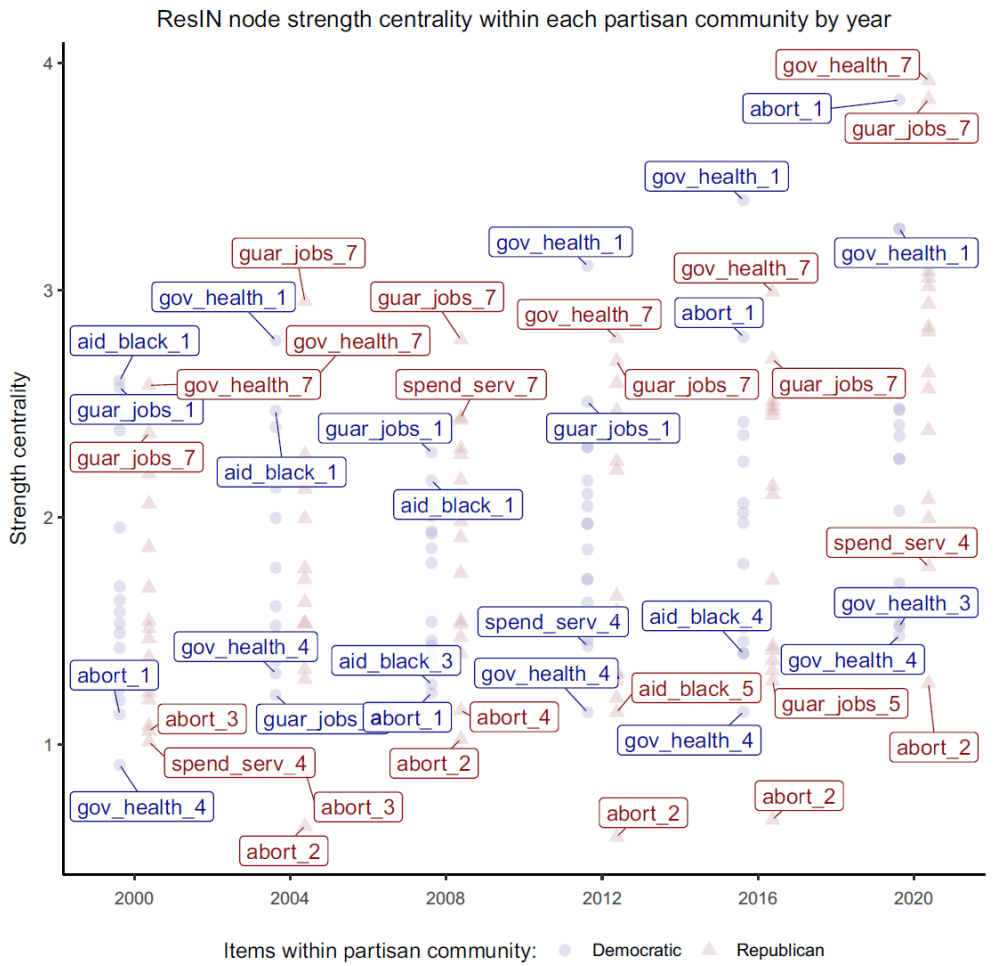
BNA: Strength centrality



ResIN: Strength centrality



ResIN: Strength centrality: Application by Chen et.al. 2025

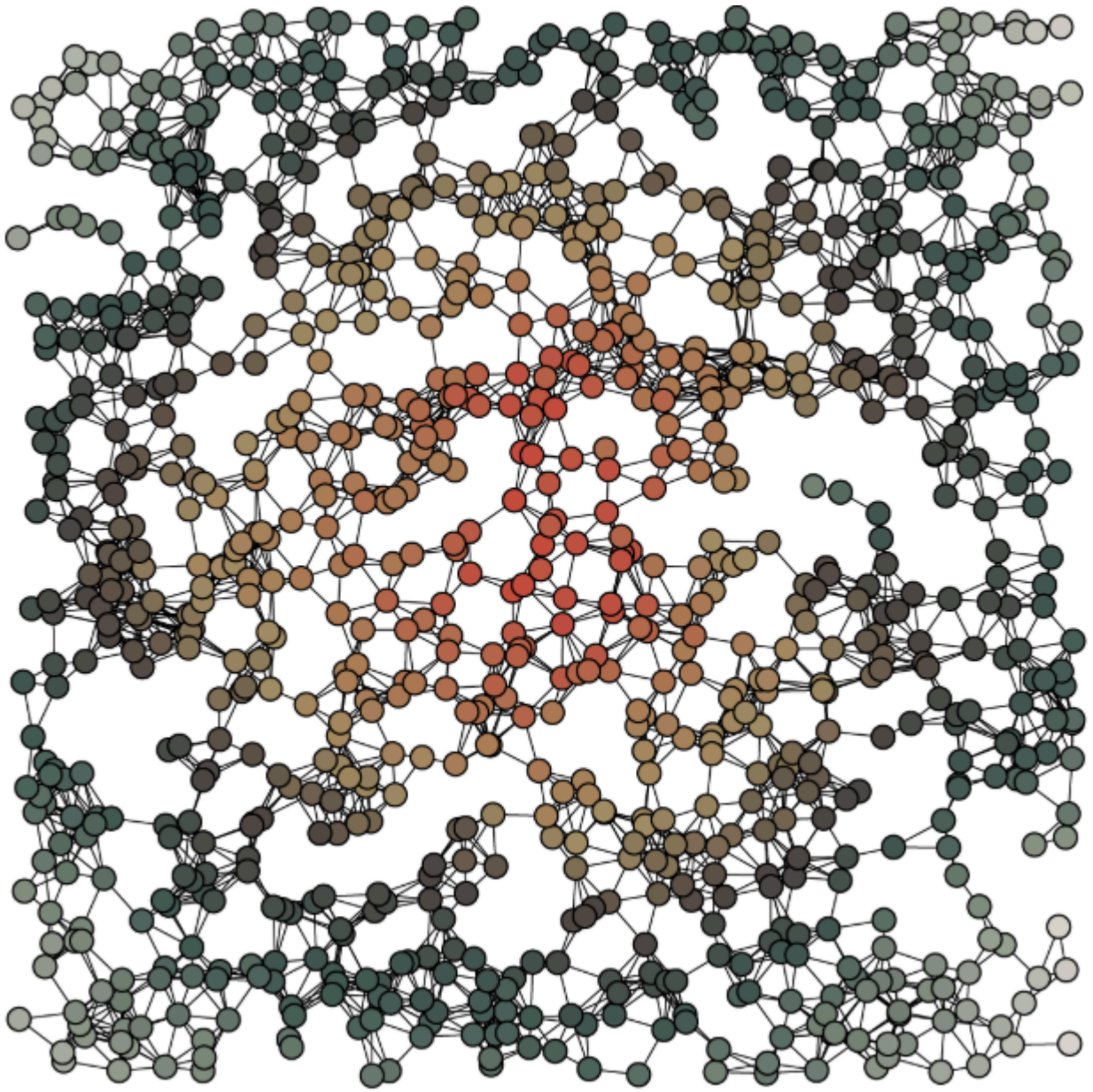


- Chen et.al. 2025

Network centrality

- Strength centrality captures the importance of immediate connectivity
- Closeness centrality captures a nodes proximity to all other nodes (how quickly can you reach the remainder of the network)
  - Average of the inverse distances from origin node to all other nodes

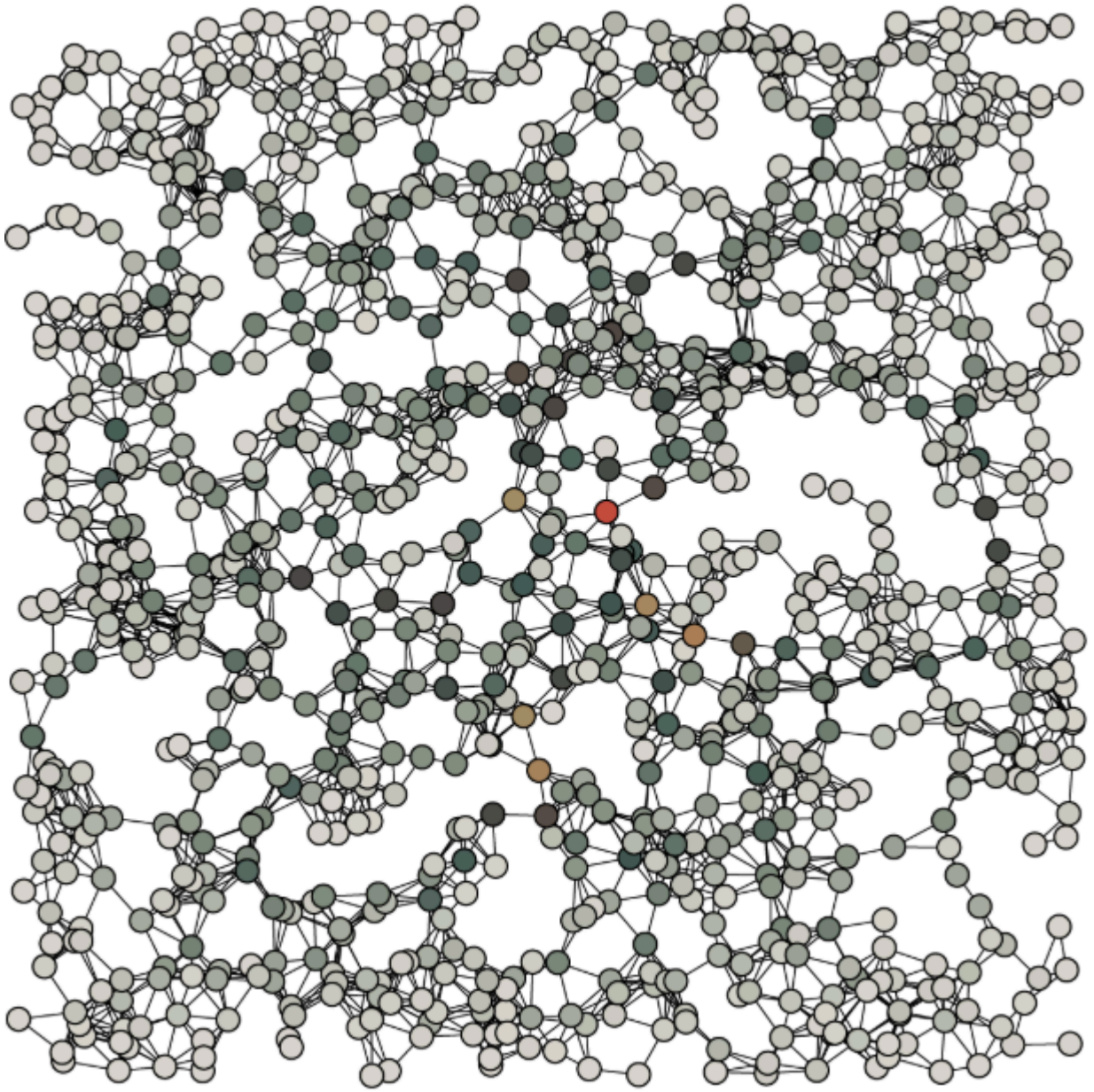
Closeness centrality



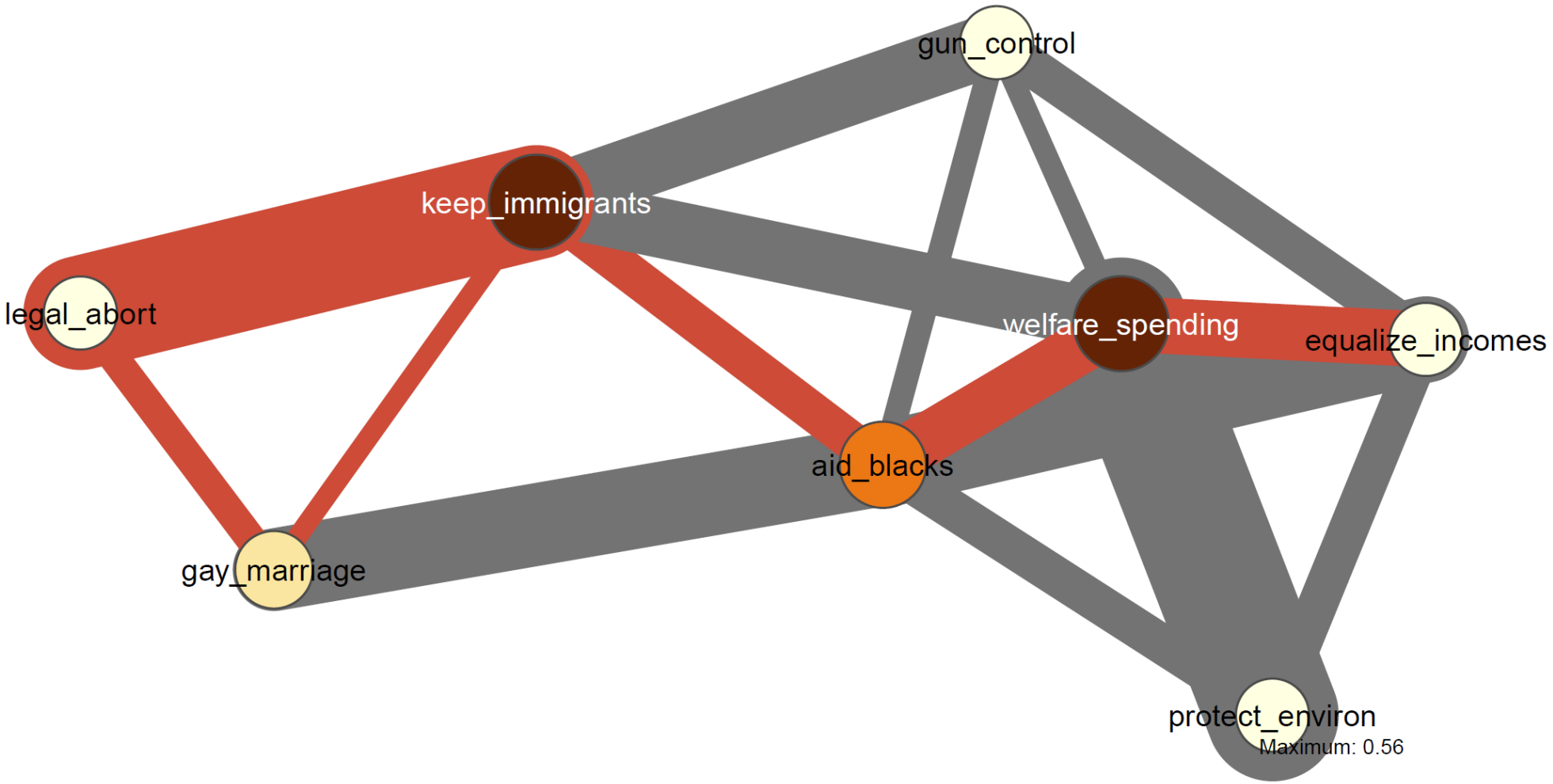
Network centrality

- Strength centrality captures the importance of immediate connectivity
- Closeness centrality captures a nodes proximity to all other nodes (how quickly can you reach the remainder of the network)
- Betweenness centrality captures a node's bridging potential between different network segments
  - Tally of the shortest paths among all node pairs that passes through a particular node

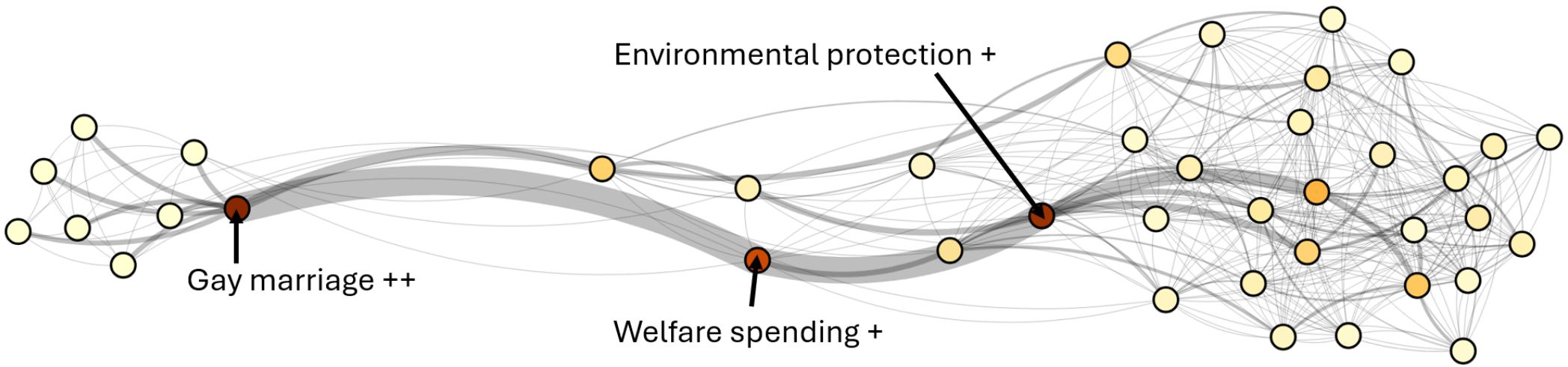
Betweenness centrality



BNA: Betweenness centrality

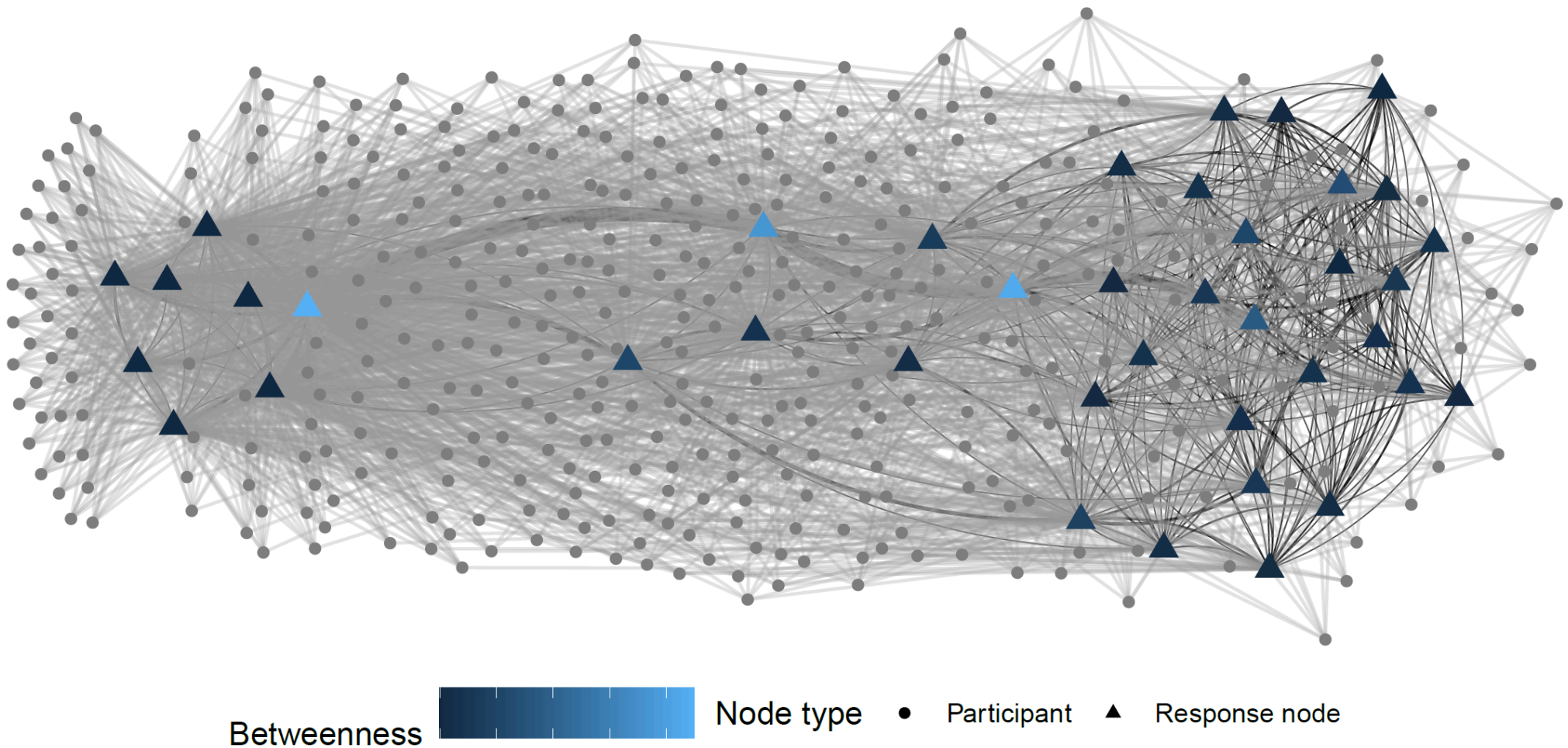


ResIN: Betweenness centrality



ResIN: Respondent centrality (multi-modal, parasocial network)

Multimodal ResIN graph



Centrality differences

	BNA	ResIN
Strength	Immediate item importance (belief influence)	Immediate importance within sub-graph (communities)
Betweenness	Global item importance / dynamic constraint	Attitude bridging potential - cross community information flow
Respondent	(not supported)	Centrality can be modeled in

BNA	ResIN
	multimodal ResIN, yielding strength and betweenness centrality estimates for individual respondents

ResIN package for R

ResIN package: getting started

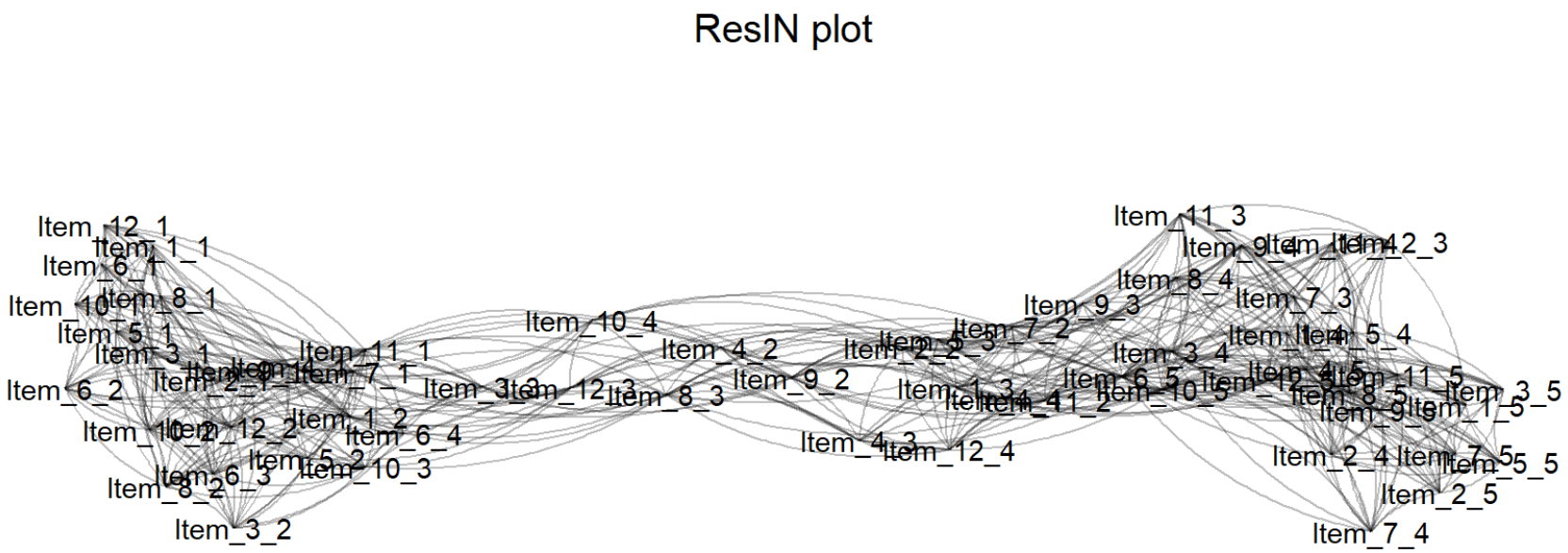
```r
## Install from CRAN (latest stable version)
options(repos = c(CRAN = "https://cran.rstudio.com/"))
if (!requireNamespace("ResIN", quietly = TRUE)) {
  install.packages("ResIN")}

## Install from Github (bleeding edge)
if (!requireNamespace("devtools", quietly = TRUE)) {
  install.packages("devtools")}

if (!requireNamespace("ResIN", quietly = TRUE)) {
  devtools::install_github("pwarncke77/ResIN")}

library("ResIN")
```

ResIN package: getting started

```r
data(lik_data)
head(lik_data)[1:5]
```

```
  Item_1 Item_2 Item_3 Item_4 Item_5
1      4      4      4      5      4
2      2      2      3      1      3
3      3      2      4      4      3
4      3      2      4      5      3
5      3      2      4      3      3
6      2      2      3      1      3
```

```r
first_ResIN <- ResIN(lik_data)
```




```
1  summary(first_ResIN)
```

```
<Summary of ResIN>
  Nodes : 60
  Edges : 570
  Scores available : yes
  Plot available   : yes
```

```
  Aux objects: adj_matrix, adj_matrix_neg, same_items, df_dummies,
cluster_probabilities, max_clusterprob, ResIN_arglist, meta_information
```

```
1 print(first_ResIN)
```

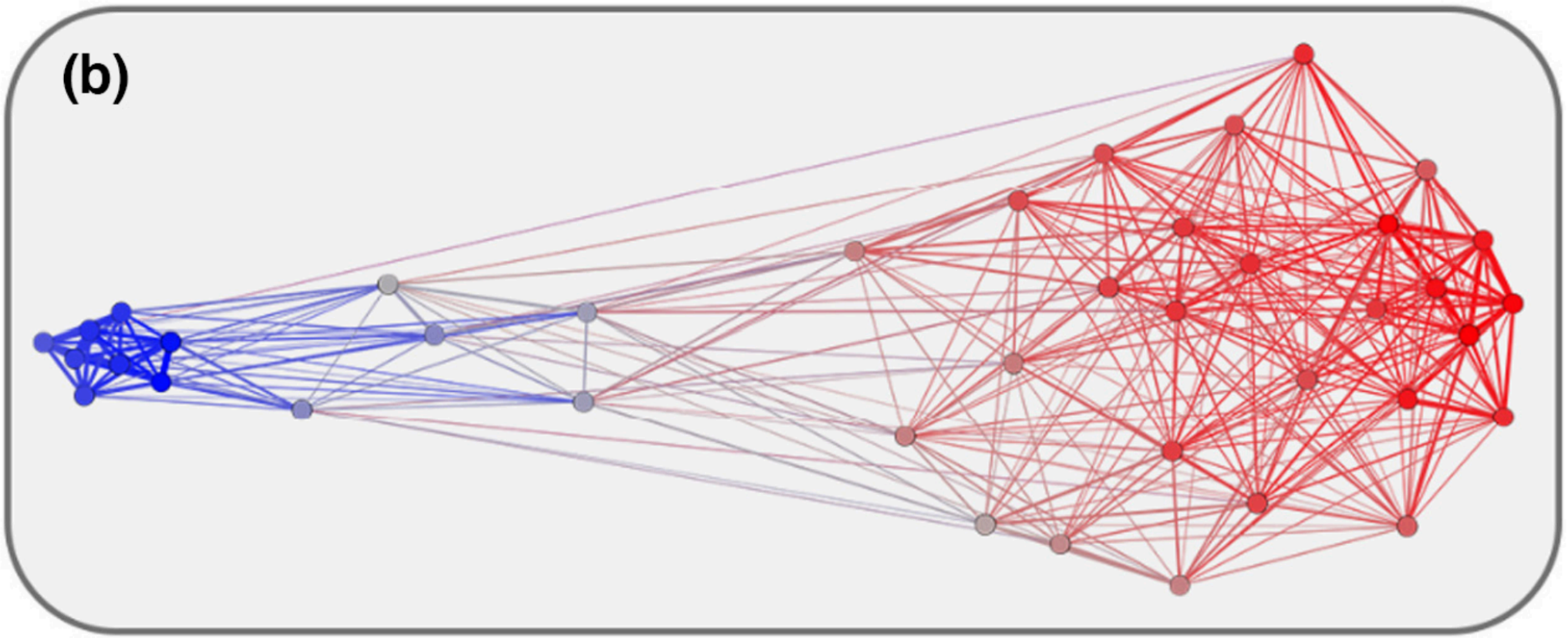
```
<ResIN object>
  ResIN_edgelist : 570 x 9
  ResIN_nodeframe: 60 x 8
  ResIN_scores   : 1000 x 6
  ResIN_ggplot   : ggplot object
  graph_stats    : list(2 elements)
  aux_objects    : list(adj_matrix, adj_matrix_neg, same_items, df_dummies,
cluster_probabilities, max_clusterprob, ResIN_arglist, meta_information)
  multimodal_output: not generated
  df_id          : ecf0d7e90a87279631d1879b31aca58e
  ResIN_version  : 2.3.1
```

```
1 plot(first_ResIN)
```

Applied example by Lüders et.al. 2023

- **Lüders, Carpentras, Quayle (2024)** "Attitude networks as intergroup realities: Using network-modelling to research attitude-identity relationships in polarized political contexts."

Applied example by Lüders et.al. 2023



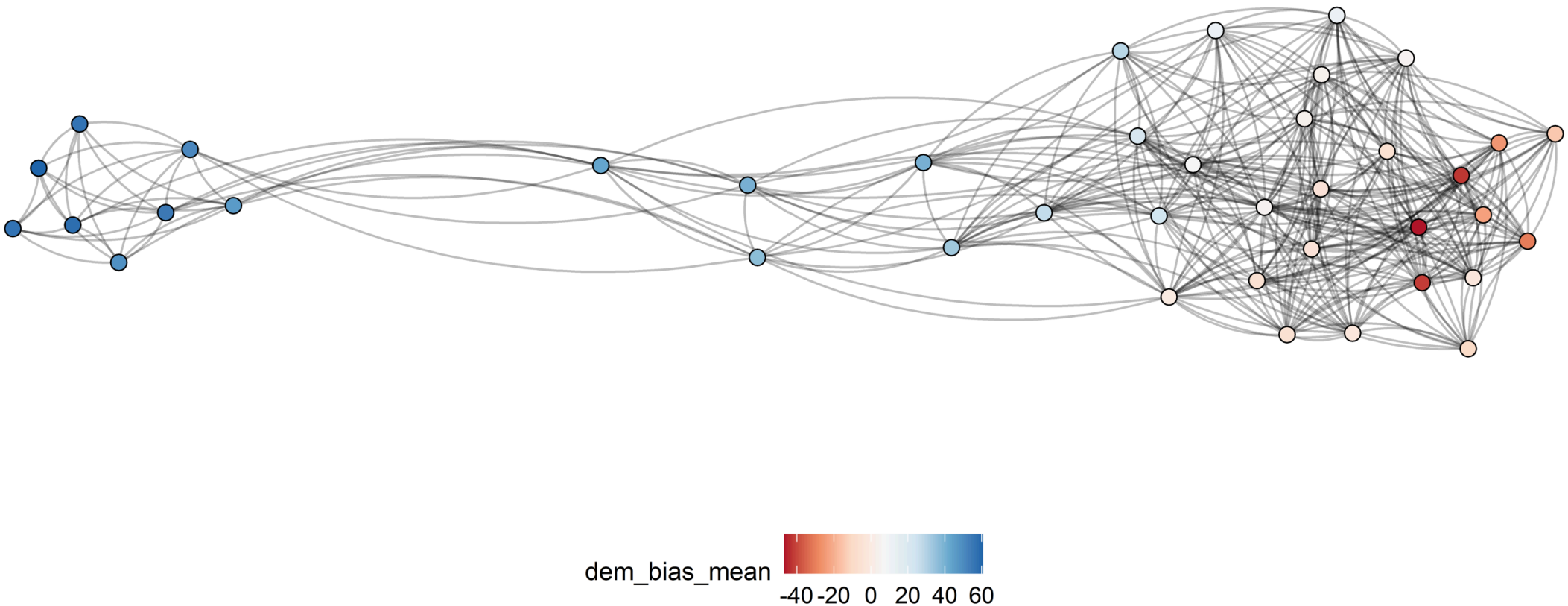
Lüders et.al. 2023: prep-work

```r
if(!require("tidyverse")) install.packages('tidyverse')
library(tidyverse, warn.conflicts = FALSE)

## Loading the data
BrJSocPsychol_2024 <- ResIN::BrJSocPsychol_2024

## Sub-setting and re-coding items in a liberal-conservative direction
Core_Items <- BrJSocPsychol_2024 %>% dplyr::select(Q9_1, Q9_2, Q9_3, Q9_4,
                                                   Q9_5, Q9_6, Q9_7, Q9_8)

## Relabeling the attitudes
node_vars = c("legal_abort", "equalize_incomes", "keep_immigrants",
              "welfare_spending", "gay_marriage", "protect_environ",
              "gun_control", "aid_blacks")

colnames(Core_Items) <- node_vars

# Assigning response symbols for easier interpretation
```

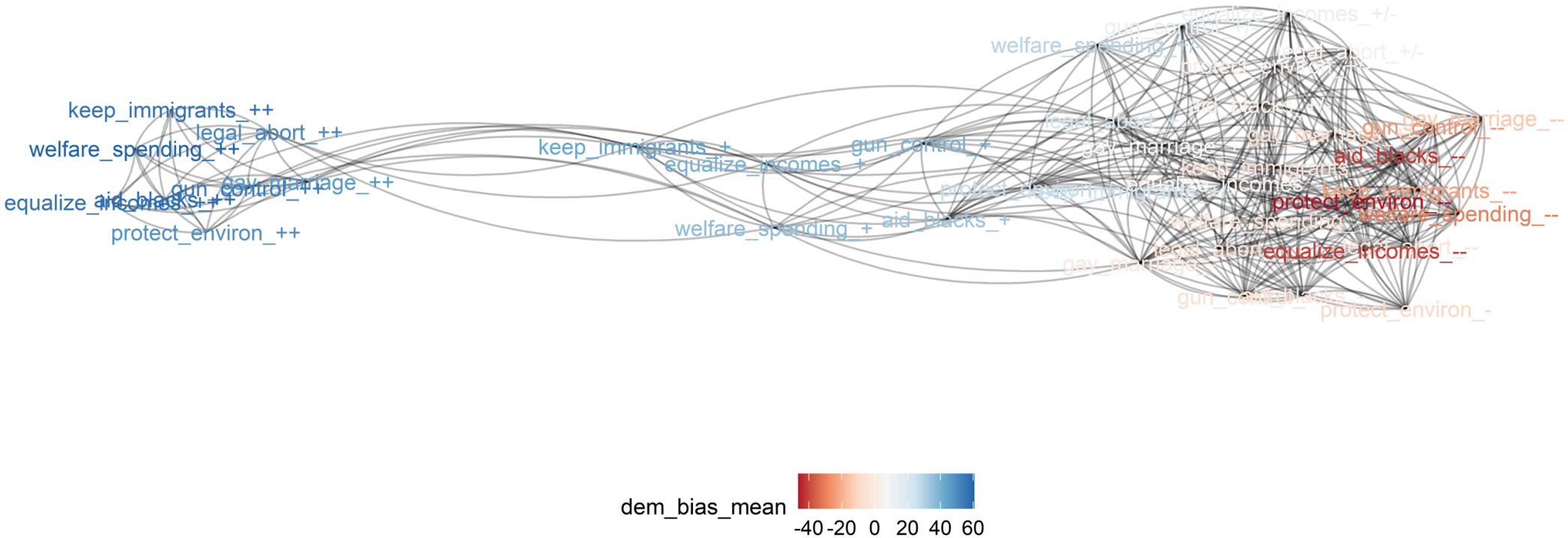
Lüders et.al. 2023: first graph

```
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   node_covars = c("dem_bias"),
                   node_costats = c("mean"),
                   plot_whichstat = "dem_bias_mean",
                   plot_responselabels = F, plot_ggplot = F,
                   left_anchor = "legal_abort_++",
                   color_palette = "RdBu", seed = 22)
```

Lüders et.al. 2023: first graph



Same graph with plot_responselabels = T



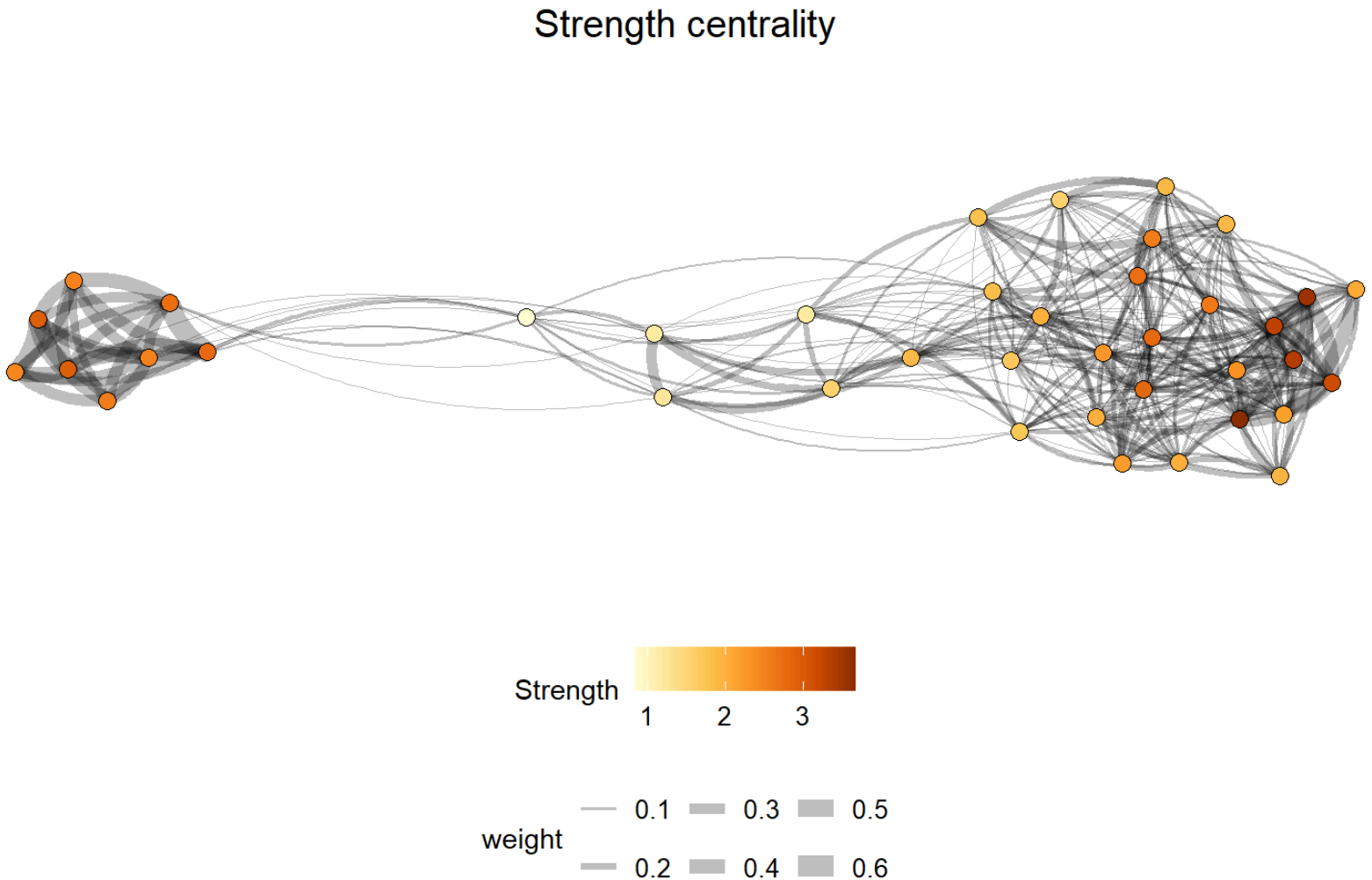
Exercise 1) Recreating strength centrality plot

```r
## Relying on the help documentation, can you find the term you need to supply to
##  supply to the "plot_whichstat" argument in to have a nodes' color intensity
##  correspond to its relative strength centralty wihin the network?

## What argument do you need to feed to "plot_edgestat" to have the thickness
##  of edges correspond to correation weights?

ResIN_out <- ResIN(Core_Items, node_vars = node_vars,
  plot_whichstat = NULL, ## Which argument do you need to supply to plot_whichstat
  plot_edgestat = NULL, ## Which argument do you need to supply to plot_edgestat?
      plot_responselabels = F,
      left_anchor = "legal_abort_++", plot_ggplot = FALSE,
      plot_title = "Strength centrality",
      color_palette = "YlOrBr", seed = 22)

## You can call ?ResIN() to retreive the help documentation
```

Exercise 1) Recreating strength centrality plot: Solution

```r
ResIN_out <- ResIN(Core_Items, node_vars = node_vars,
  plot_whichstat = "Strength",
  plot_edgestat = "weight",
  plot_responselabels = F,
  left_anchor = "legal_abort_++",
  plot_title = "Strength centrality",
  color_palette = "YlOrBr", seed = 22,
  plot_ggplot = F)
```

Exercise 1) Recreating strength centrality plot: Solution



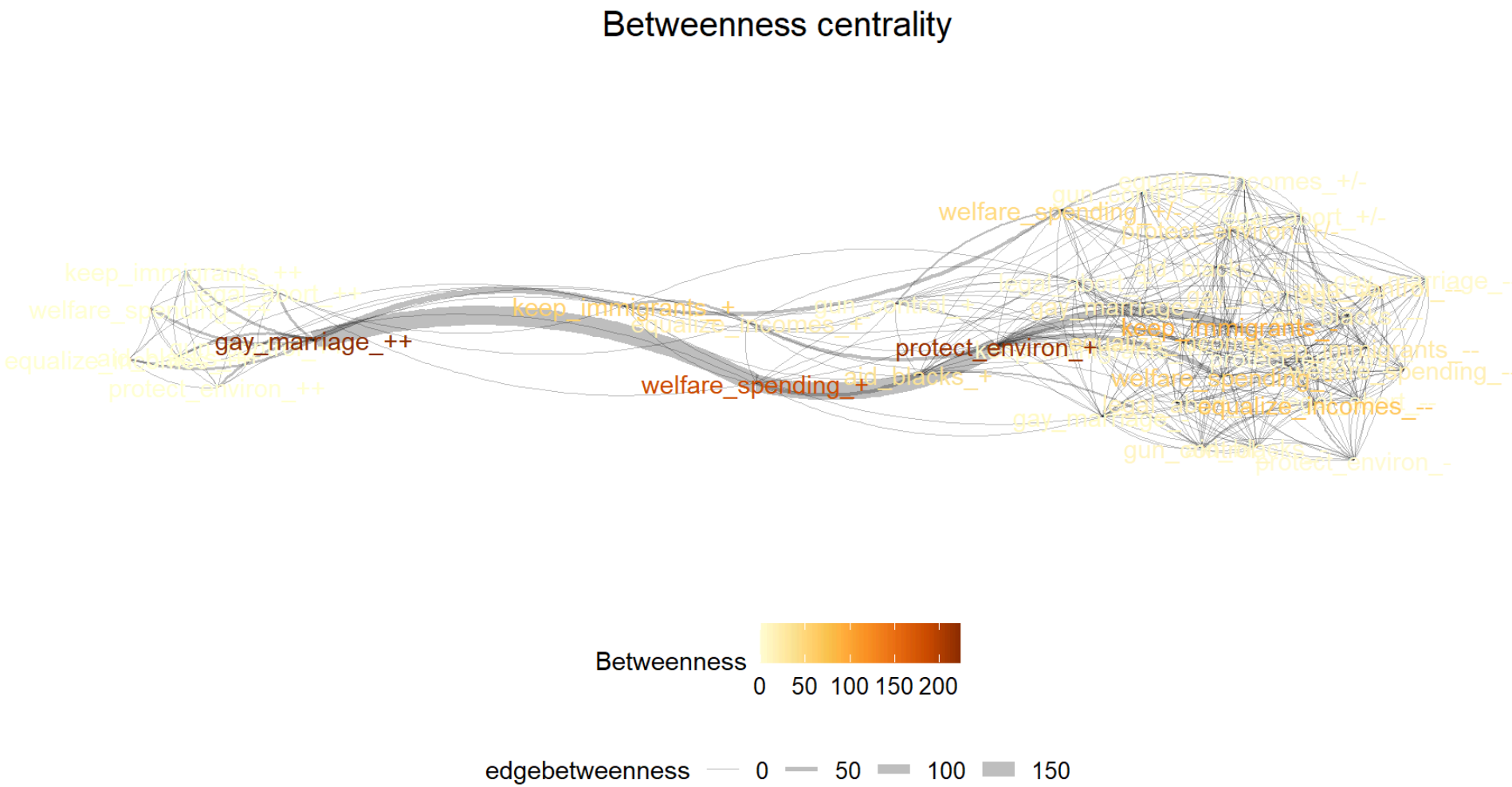
Exercise 2) Recreating betweenness centrality plot

```
## Relying on the help documentation, can you find the term you need to supply to
##  supply to the "plot_whichstat" argument in to have a nodes' color intensity
##  correspond to its relative betweenness centralty wihin the network?

## What argument do you need to feed to "plot_edgestat" to have the thickness
##  correspond to edge-betweenness centrality to correation weights?

ResIN_out <- ResIN(Core_Items, node_vars = node_vars,
   plot_whichstat = NULL, ## find the correct argument here
   plot_edgestat = NULL , ## find the correct argument here
   plot_responselabels = T,
   left_anchor = "legal_abort_++", plot_ggplot = F,
   plot_title = "Betweenness centrality", color_palette = "YlOrBr", seed = 22)
```

Exercise 2) Recreating betweenness centrality plot: solution

```
ResIN_out <- ResIN(Core_Items, node_vars = node_vars,
    plot_whichstat = "Betweenness",
    plot_edgestat = "edgebetweenness",
    plot_responselabels = T,
    left_anchor = "legal_abort_++", plot_ggplot = F,
    plot_title = "Betweenness centrality", color_palette = "YlOrBr", seed = 22)
```

Exercise 2) Recreating betweenness centrality plot: solution

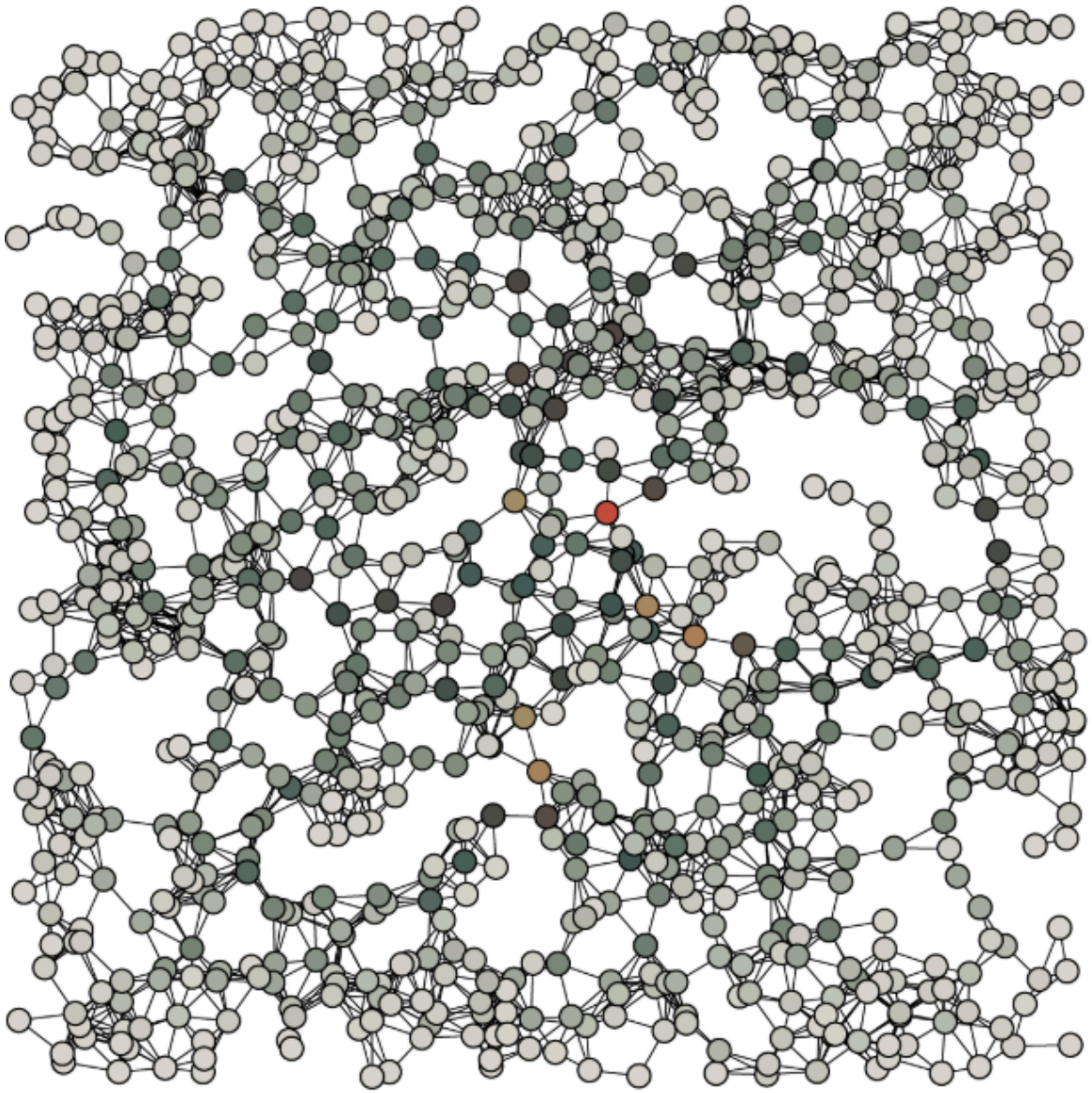


Day 2: Cluster detection (& bootstrapping)

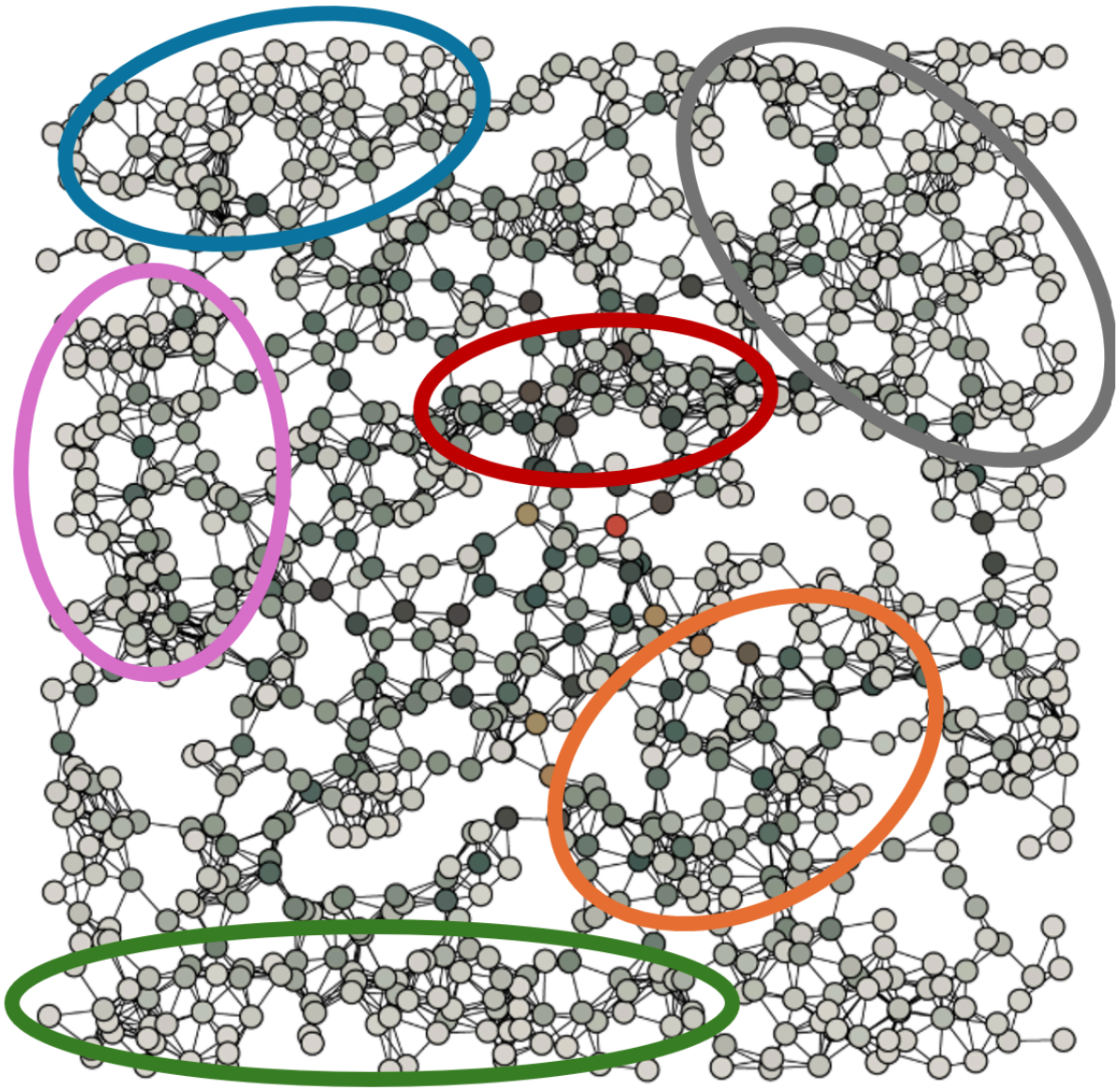
Cluster detection

- How do we make out sub-structures (communities) in within networks?
- Can we leverage statistics to get reasonable (defensible) answers?
- Goal: Cluster detection aims identify groupings of nodes that are more tightly connected to each other than to the rest of the network

Cluster detection



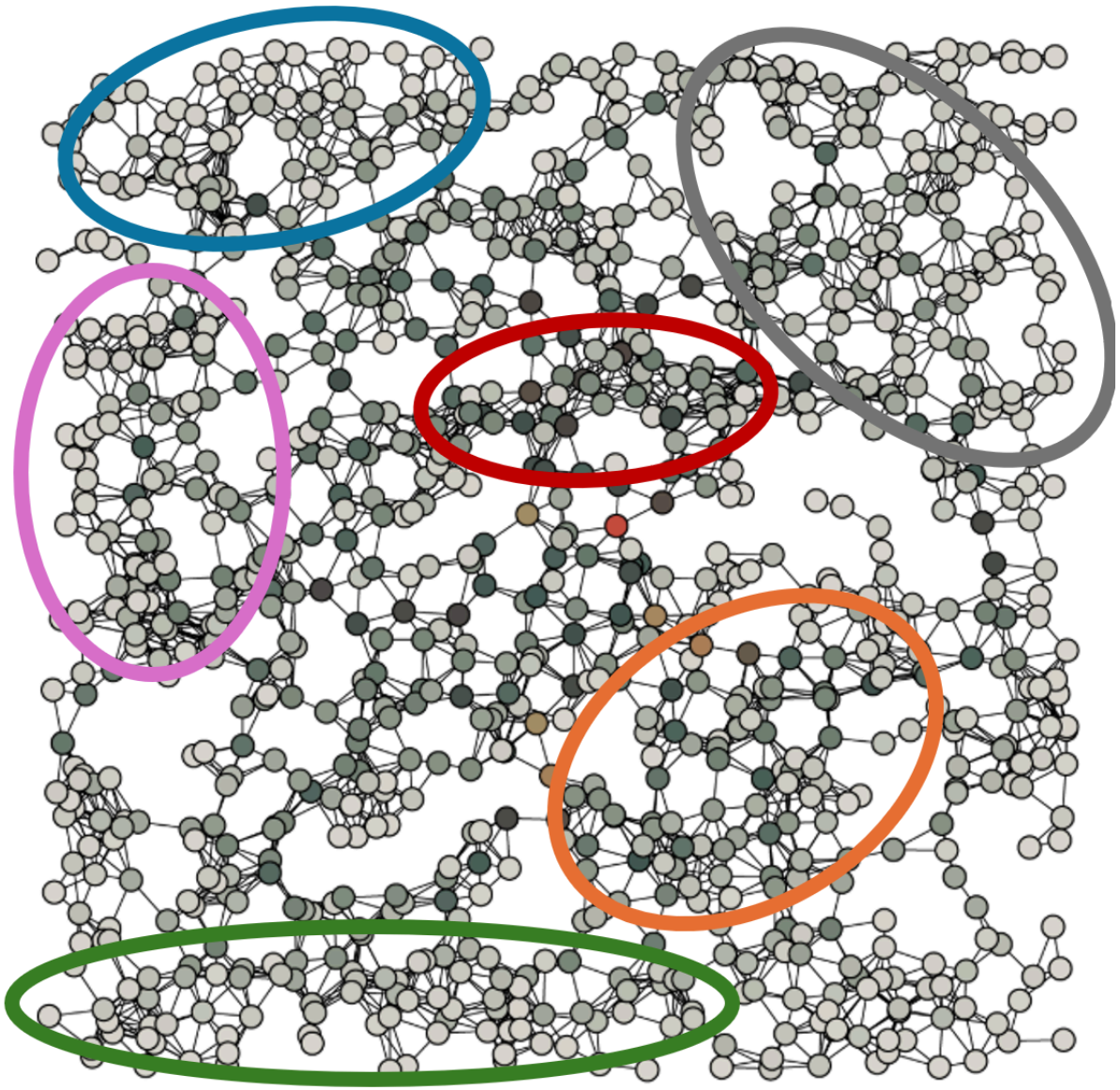
Cluster detection



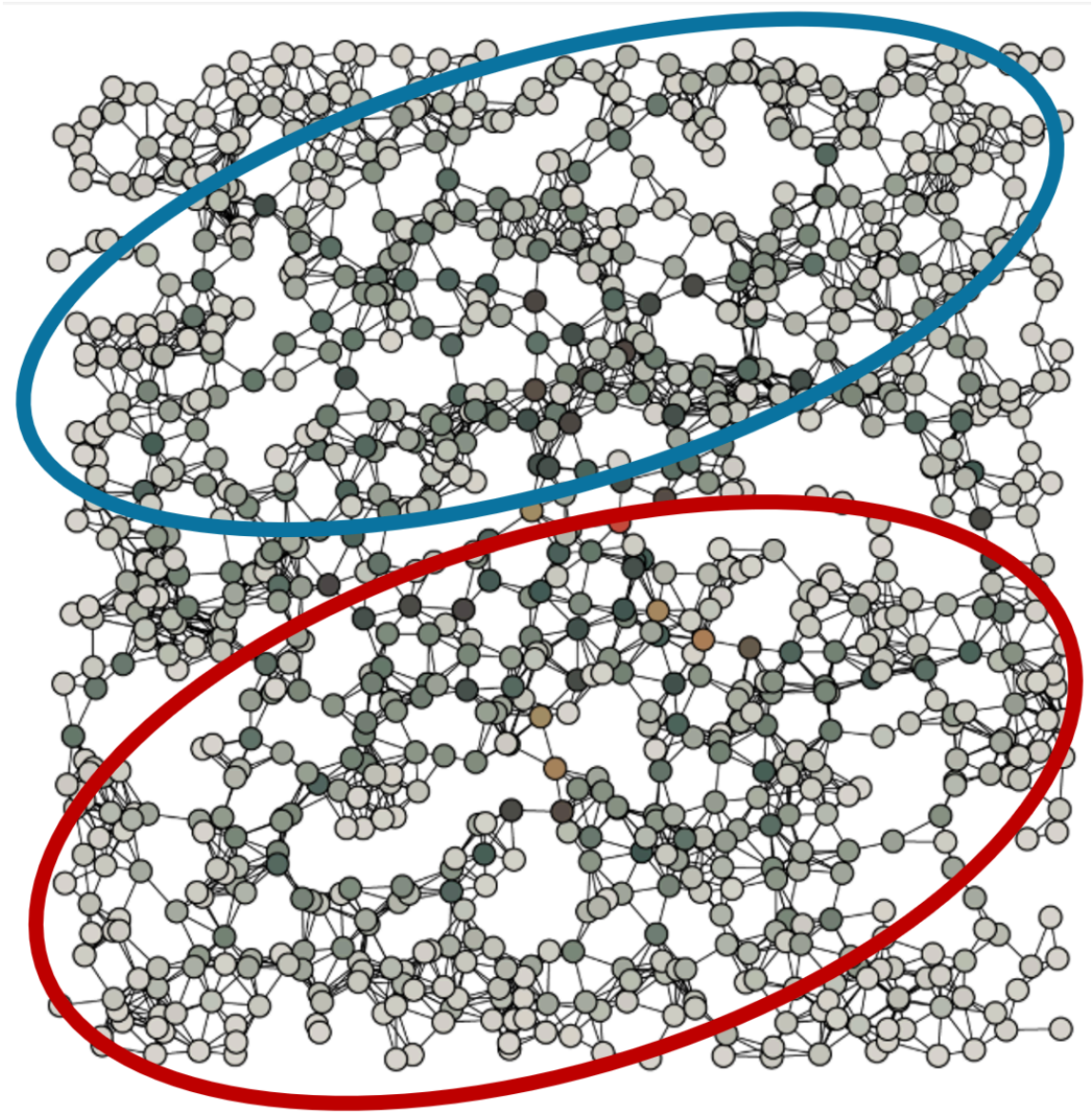
Cluster detection

- SPOILER ALLERT: An area of math/statistics where art meets science
  - some choices (methods, algorithms, solutions) more reasonable than others … BUT:
  - there are no ground truths and lots of trade-offs to be aware of
  - trade-off in parsimony: are you a lumper or a chopper?

Cluster detection



Cluster detection



Modularity detection

- Measure of the relative density of edges within clusters versus within communities
- Optimizing it provides the theoretically optimal grouping of the nodes in a network

$$Q = \frac{1}{2m} \sum_{i=1}^{N} \sum_{j=1}^{N} [a_{i,j} - \frac{k_i k_j}{2m}] \delta(c_i, c_j)$$

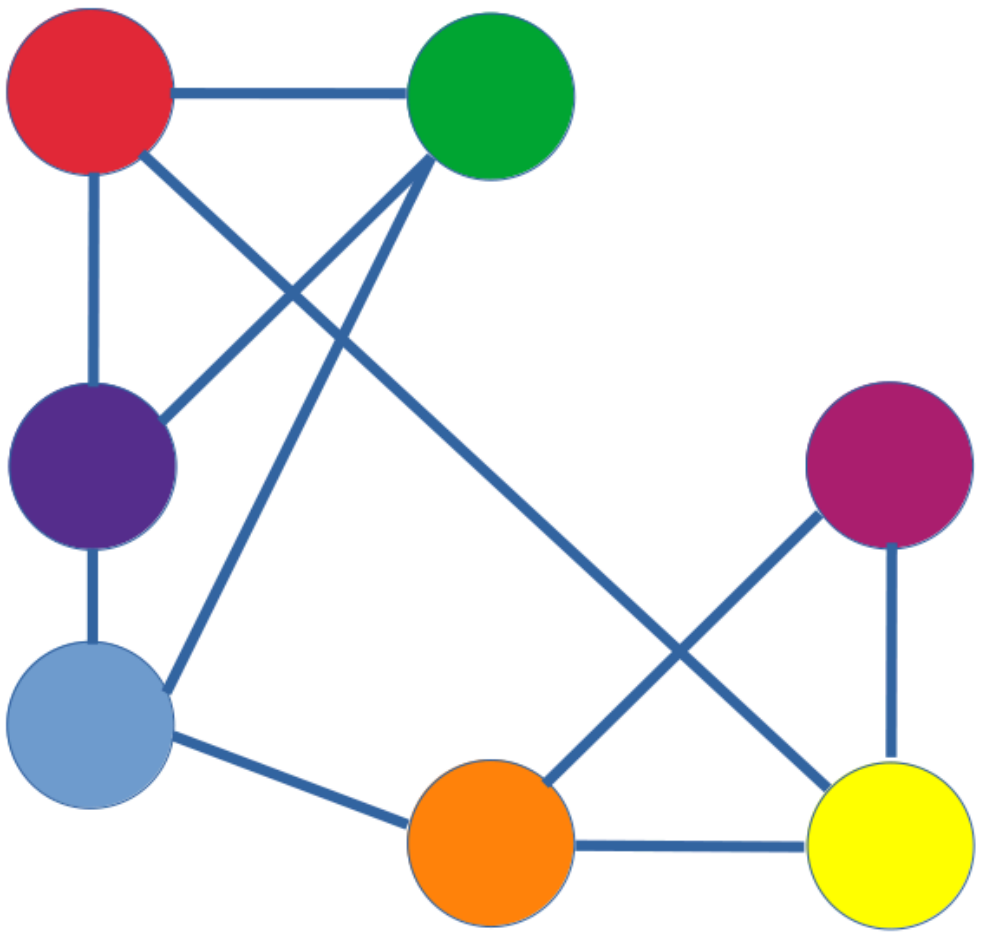
- Need heuristic algorithms to solve (approximate) modularity in practice

Louvain algorithm phase 1

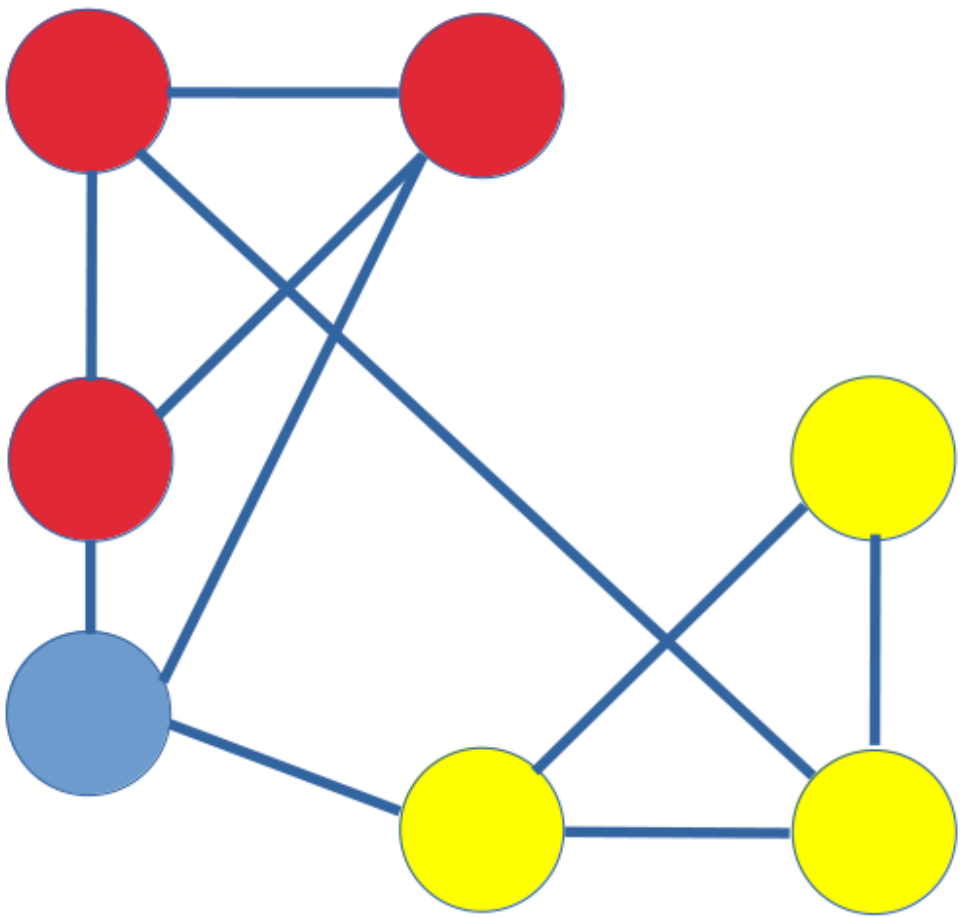
Start by considering each node as its separate community

1. Move any give node into any adjacent community
2. Observe how the local modularity (of the new community) changes; if it improves move the node into the new community, if not move it into the next community and so on
3. Rinse and repeat for all nodes until global modularity plateaus.

Louvain algorithm phase 1



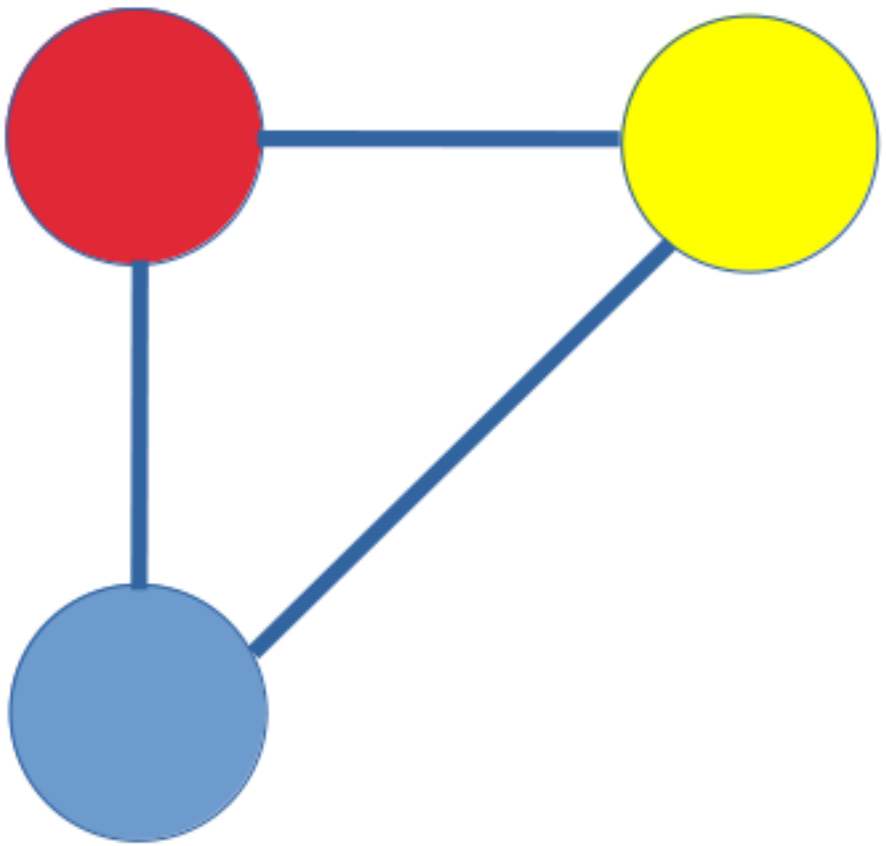
Louvain algorithm phase 1



Louvain algorithm phase 2

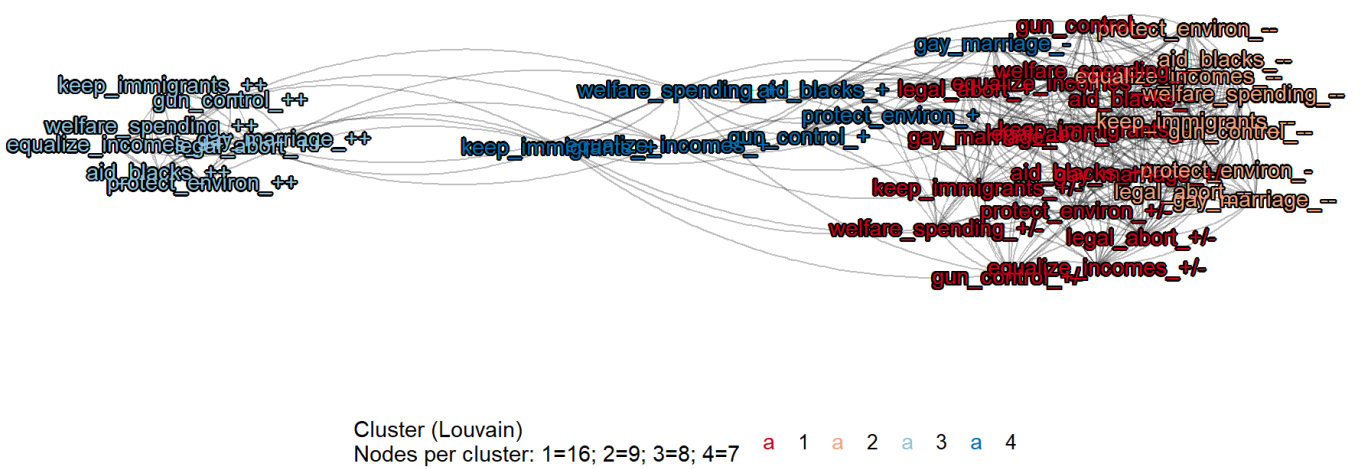
- We next move one level up and treat each community as a node in the simplified, higher-order network
- If modularity increases by separating/reallocating clusters we combine them until modularity doesn't increase any further

Louvain algorithm phase 2



Cluster detection with ResIN: Louvain example

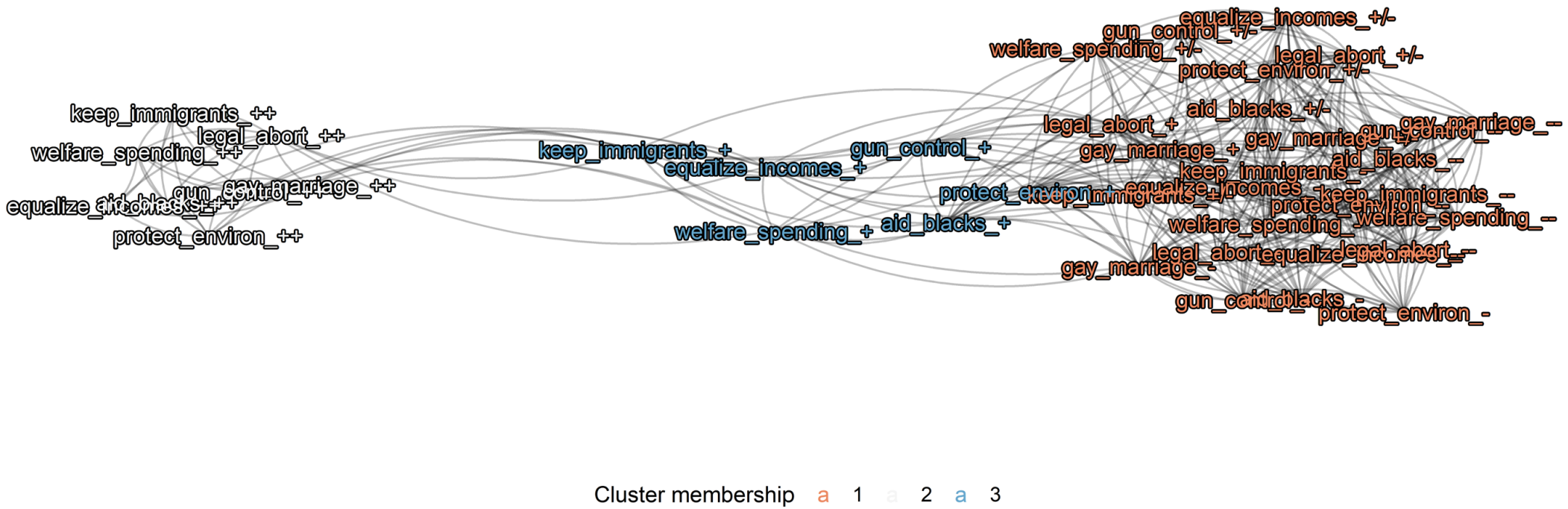
```
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   detect_clusters = TRUE,
                   cluster_method = "cluster_louvain",
                   plot_whichstat = "cluster",
                   plot_responselabels = T, plot_ggplot = T,
                   left_anchor = "legal_abort_++",
                   plot_title = "",
                   color_palette = "RdBu", seed = 55)
```



Cluster detection: Infomap

```
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   detect_clusters = TRUE, cluster_method = "cluster_infomap",
                   plot_whichstat = "cluster",
                   plot_responselabels = T, plot_ggplot = F,
                   left_anchor = "legal_abort_++",
                   color_palette = "RdBu", seed = 55)
```

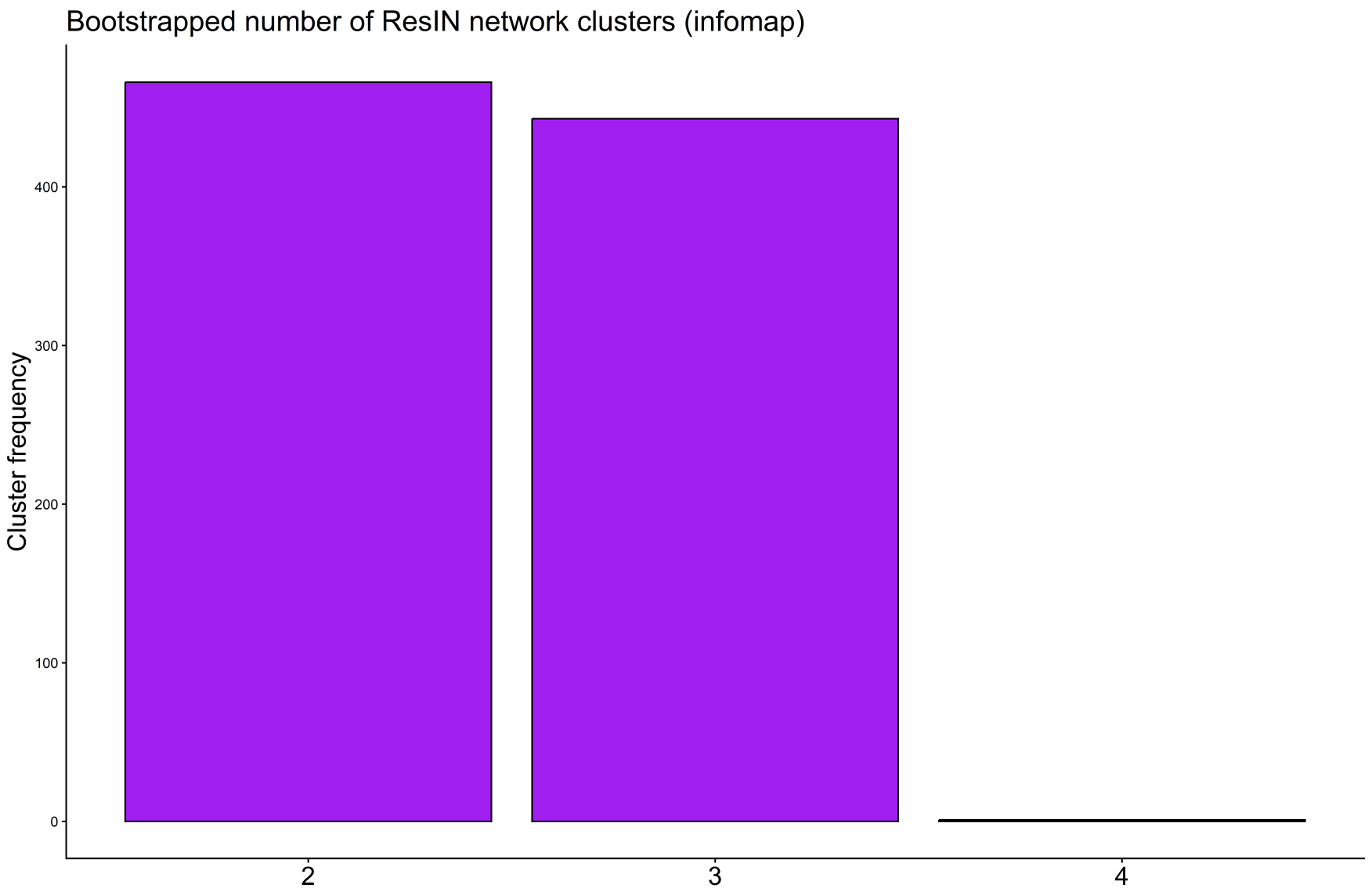
Cluster detection: Infomap



Bootstrapping uncertainty: Infomap

```r
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   detect_clusters = TRUE, cluster_method = "cluster_infomap",
                   generate_ggplot = F, plot_ggplot= F,
                   left_anchor = "legal_abort_++", seed = 22)

## First, prepare the bootstrap procedure (number of iterations and type)
  prepped_bst <- ResIN_boots_prepare(ResIN_out, n = 1000, boots_type = "resample")

## Second, carry out  procedure, optionally leveraging CPU parallelism
  #executed_bst <- ResIN_boots_execute(prepped_bst, parallel = T)
  #saveRDS(executed_bst, "executed_bst.RDS")
  #executed_bst <- readRDS("executed_bst.RDS")

## Third, extract the detected cluster (done via automatic forward search)
  # extracted_summarized <- ResIN_boots_extract(executed_bst, what = "cluster")
  # saveRDS(extracted_summarized, "extracted_summarized.RDS")
extracted_summarized <- readRDS("extracted_summarized.RDS")

```

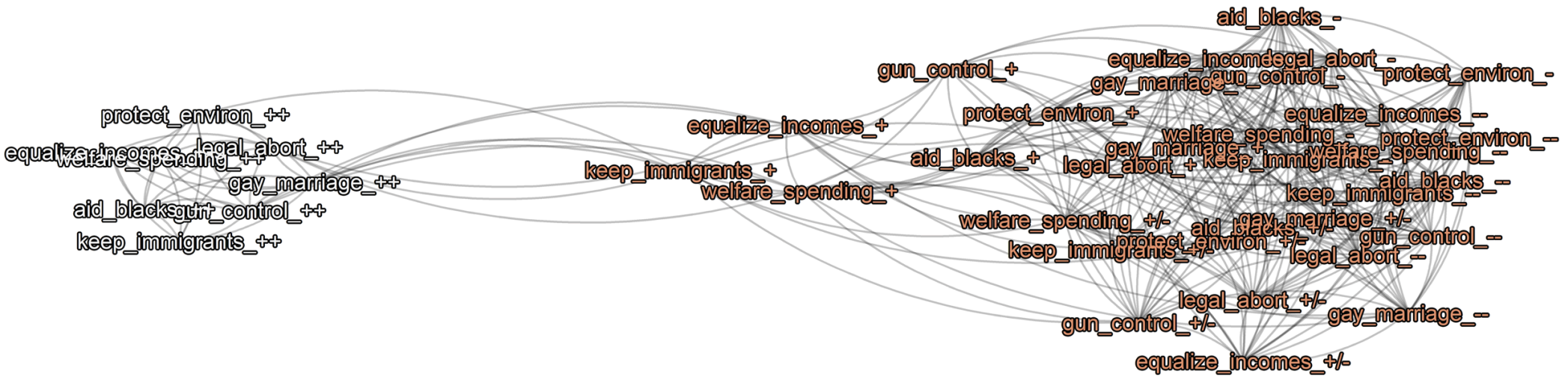
Bootstrapping uncertainty: Infomap



Cluster detection: Infomap

```
ResIN_out <- ResIN(df = Core_Items[20:380,], node_vars = node_vars,
                   detect_clusters = TRUE, cluster_method = "cluster_infomap",
                   generate_ggplot = T, plot_ggplot = F,
                   seed = 55, left_anchor = "legal_abort_++",
                   plot_whichstat = "cluster")
```

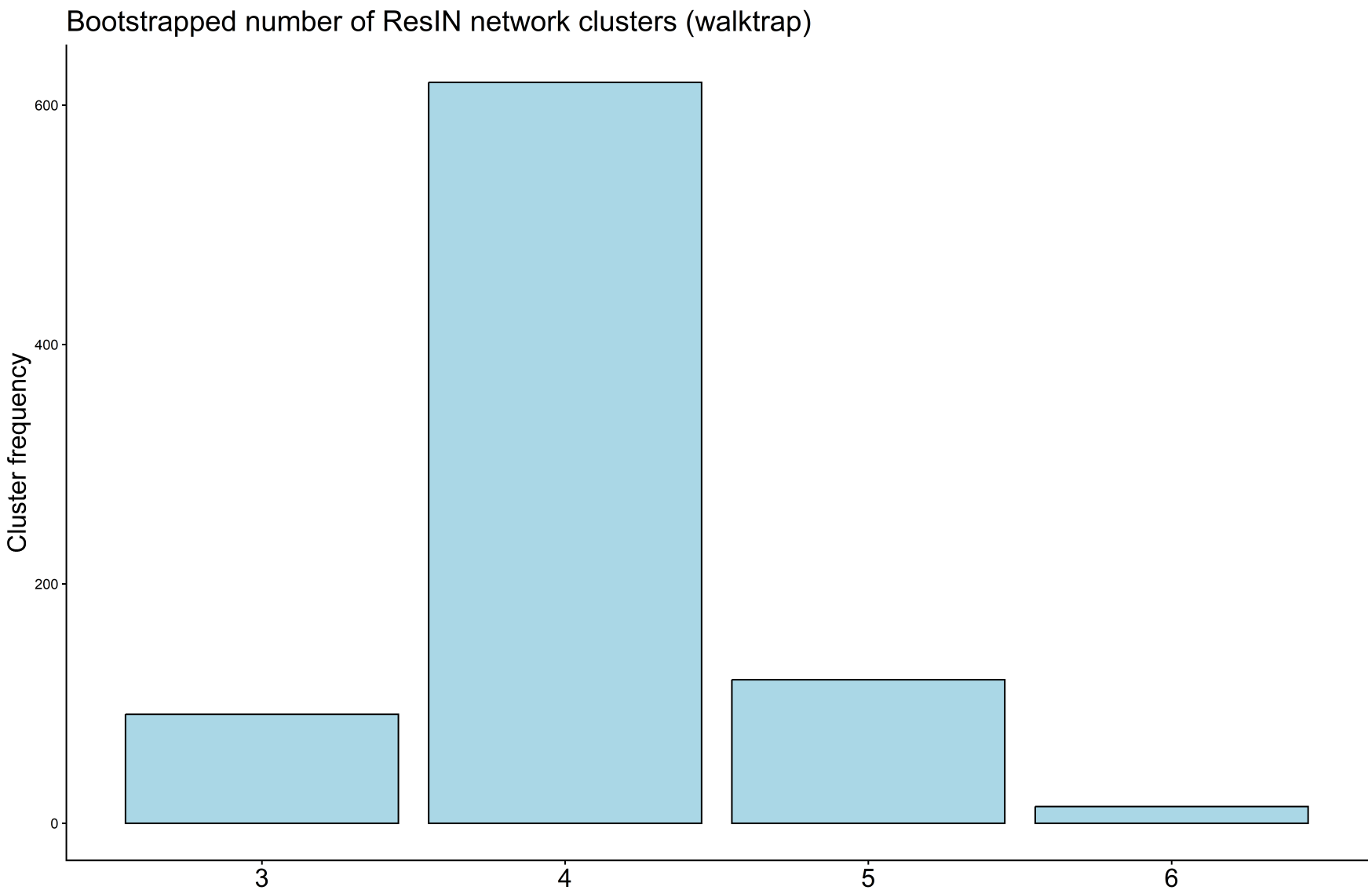
Same graph with plot_responselabels = T



Cluster detection: Walktrap

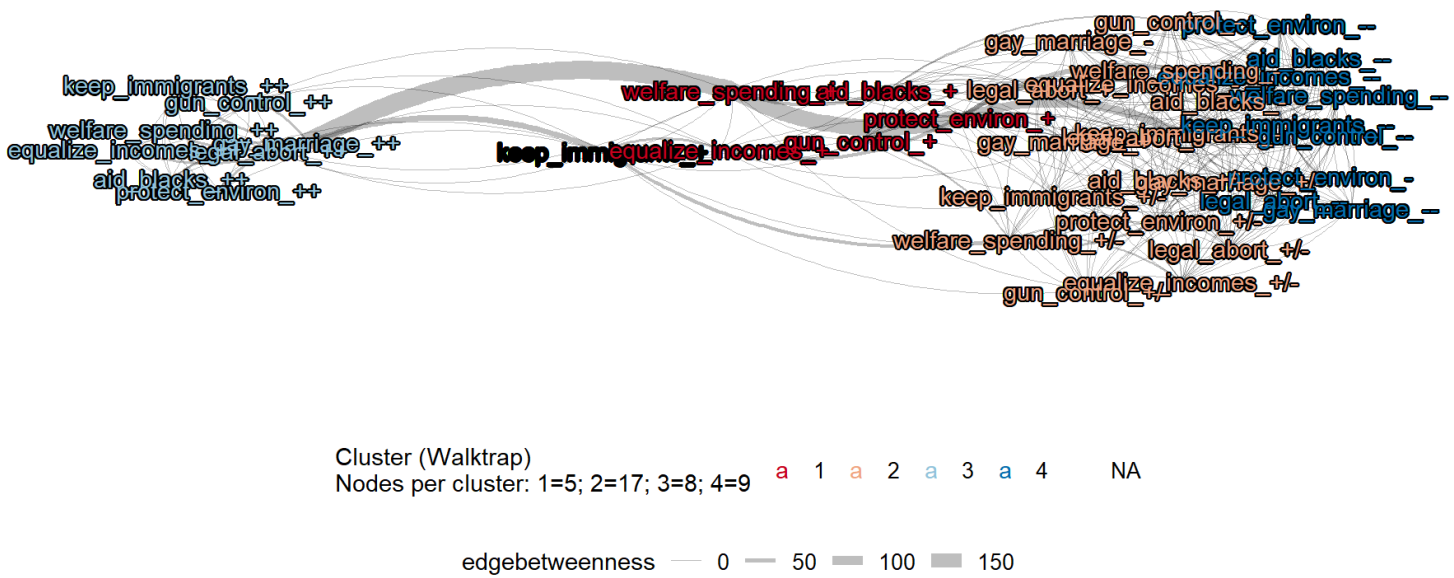
```r
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   detect_clusters = T,
                   cluster_method = "cluster_walktrap",
                   remove_nonsignificant = T,
                   generate_ggplot = F, plot_ggplot= F,
                   left_anchor = "legal_abort_++", seed = 55)

## First, prepare the bootstrap procedure (number of iterations and type)
prepped_bst <- ResIN_boots_prepare(ResIN_out, n = 1000, boots_type = "resample")

## Second, carry out  procedure, optionally leveraging CPU parallelism
 # executed_bst <- ResIN_boots_execute(prepped_bst, parallel = T)
 # saveRDS(executed_bst, "executed_bst_wt.RDS")

## Third, extract the detected cluster (done via automatic forward search)
 # executed_bst_wt <- readRDS("executed_bst_wt.RDS")
 # summarized_bst_wt <- ResIN_boots_extract(executed_bst_wt, what = "cluster")
 # saveRDS(summarized_bst_wt, "summarized_bst_wt.RDS")
```

Cluster detection: Walktrap



Cluster detection: Walktrap (network plot)

```
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   detect_clusters = T,
                   cluster_method = "cluster_walktrap",
                   generate_ggplot = T, plot_ggplot= T,
                   plot_whichstat = "cluster", plot_title = "",
                   plot_edgestat = "edgebetweenness",
                   left_anchor = "legal_abort_++", seed = 55)
```

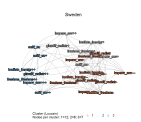


Exercise 1) Political attitude clustering in multi-party systems

```r
#### EXCERCISE: CLUSTER DETECTION IN THE 2018 ESS ####
ess_9 <- readRDS("ess_9.rds")

ess_9 <-  ess_9  %>% dplyr::select(
          cntry,    ## Country
          pspwght,  ## Post-stratification weights
          lrscale,  ## Left right scale
          impenv,   ## Environmental protection
          gincdif,  ## Reduce income differences
          freehms,  ## Gays and lesbians should be able to live life freely
          euftf,    ## Eu- further integration
          imdfetn,  ## Allow many/few immigrants of different race/ethnic group
          )

node_vars  <- c("impenv", "gincdif", "freehms", "euftf", "imdfetn")
```

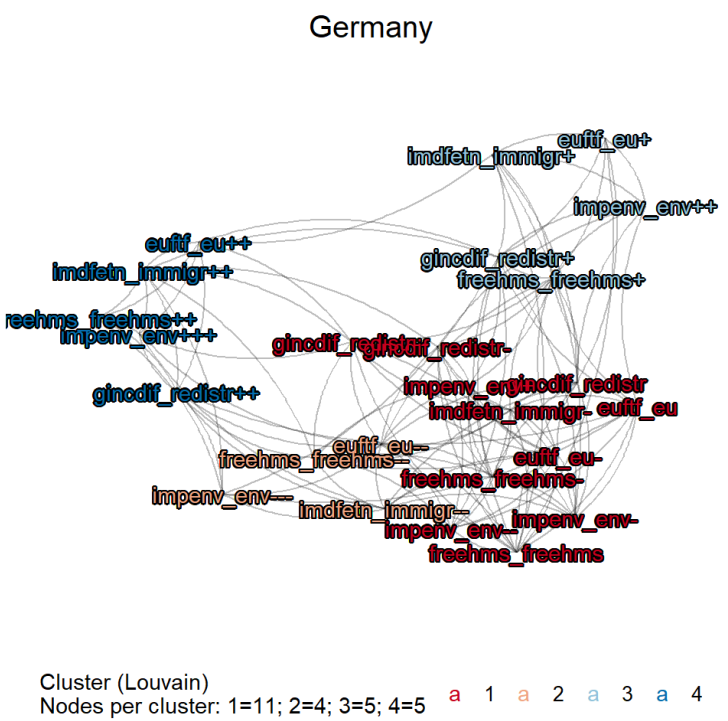
Exercise 1) Political attitude clustering in multi-party systems

```r
ResIN_ESS_SE <- ess_9 %>% filter(cntry == "SE") %>% ResIN(., node_vars = node_va
                                                         detect_clusters = TRUE,
                                                         cluster_method = "cluster_louvain",
                                                         plot_whichstat = "cluster",
                                                         left_anchor = "impenv_env+++",
                                                         weights = "pspwght",
                                                         plot_title = "Sweden",
                                                         seed = 22)
```



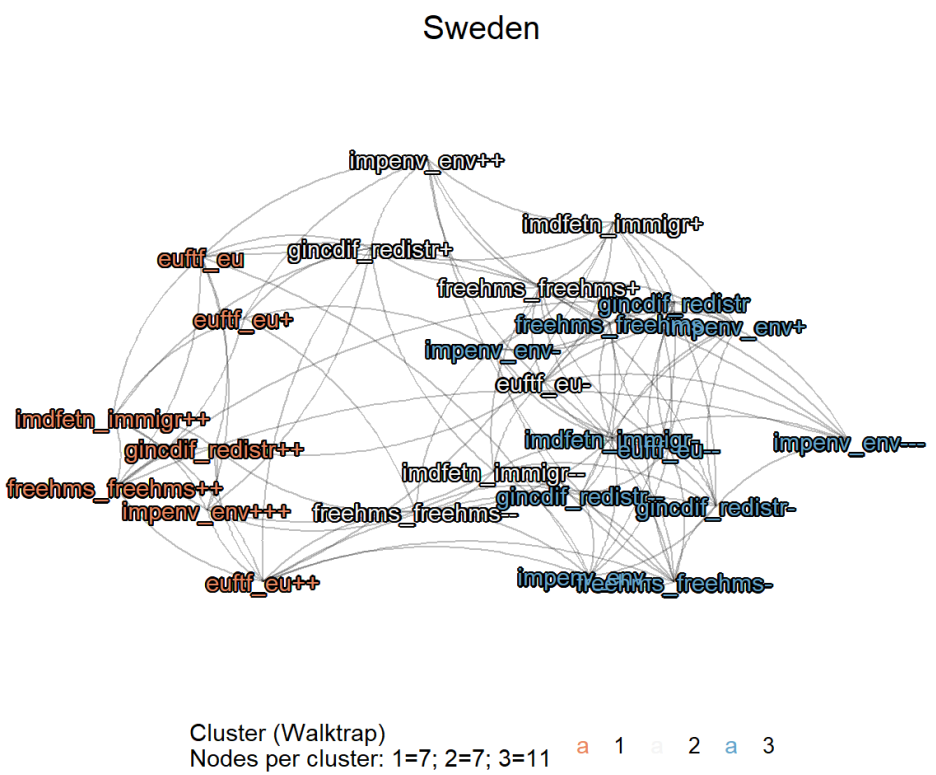
Exercise 1) Possible solution I

```r
ResIN_ESS_DE <-ess_9 %>% filter(cntry == "DE") %>% ResIN(., node_vars = node_vars,
                                                         detect_clusters = TRUE,
                                                         cluster_method = "cluster_louvain",
                                                         plot_whichstat = "cluster",
                                                         left_anchor = "impenv_env+++",
                                                         weights = "pspwght",
                                                         plot_title = "Germany",
                                                         seed = 22)
```



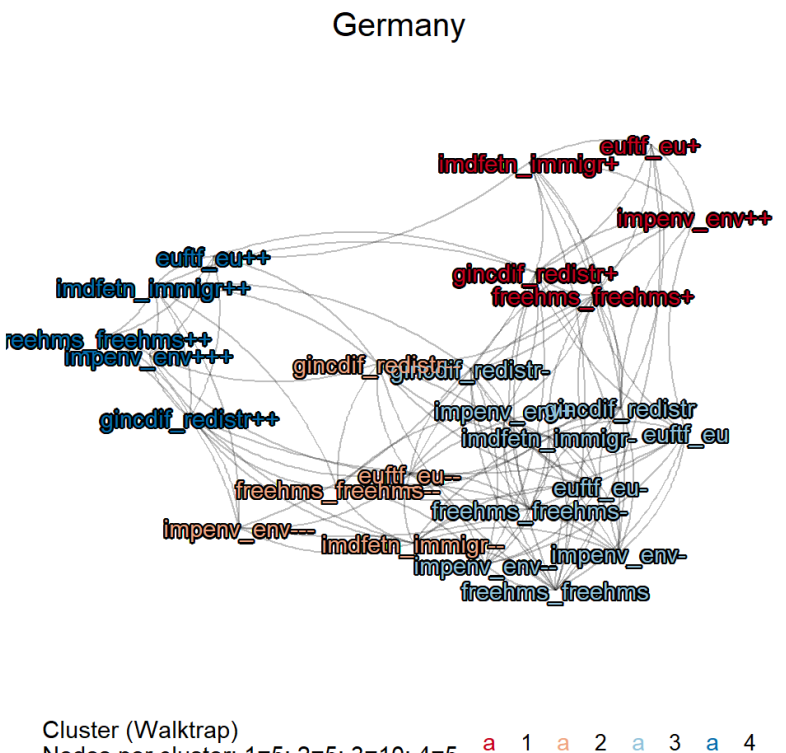
Exercise 1) Possible solution II

```r
ResIN_ESS_SE <- ess_9 %>% filter(cntry == "SE") %>% ResIN(., node_vars = node_vars
                                                         detect_clusters = TRUE,
                                                         cluster_method = "cluster_walktrap",
                                                         plot_whichstat = "cluster",
                                                         left_anchor = "impenv_env+++",
                                                         weights = "pspwght",
                                                         plot_title = "Sweden", seed = 22)
```



Exercise 1) Possible solution II

```
ResIN_ESS_DE <- ess_9 %>% filter(cntry == "DE") %>% ResIN(., node_vars = node_vars
                                                         detect_clusters = TRUE,
                                                         cluster_method = "cluster_walktrap",
                                                         plot_whichstat = "cluster",
                                                         left_anchor = "impenv_env+++",
                                                         weights = "pspwght",
                                                         plot_title = "Germany", seed = 22)
```



Exercise 2) Checking robustness of your solution via bootstrap

- **Task 3**: For a single country, try to specify a bootstrapping procedure to assess the robustness of your detected solution
- The ResIN package vignette provides a comprehensive exposition of the three ResIN bootstrapping functions: ResIN_boots_prepare(), ResIN_boots_execute(), and ResIN_boots_extract(): https://pwarncke77.github.io/ResIN/articles/ResIN-VIGNETTE.html#bootstrapping

Exercise 2) Possible solution

```r
## 3.1: Generate the ResIN network we'd like to resample:
res_outDE <- ess_9 %>% filter(cntry == "DE") %>% ResIN(., node_vars = node_vars,
                                                       detect_clusters = TRUE,
                                                       cluster_method = "cluster_walktrap",
                                                       left_anchor = "impenv_env+++",
                                                       generate_ggplot = F, plot_ggplot = F,
                                                       weights = "pspwght", seed = 22)

## Adding survey weights here
wt_DE <- ess_9 %>% filter(cntry == "DE") %>% dplyr::select(pspwght)
```

Exercise 2) Possible solution

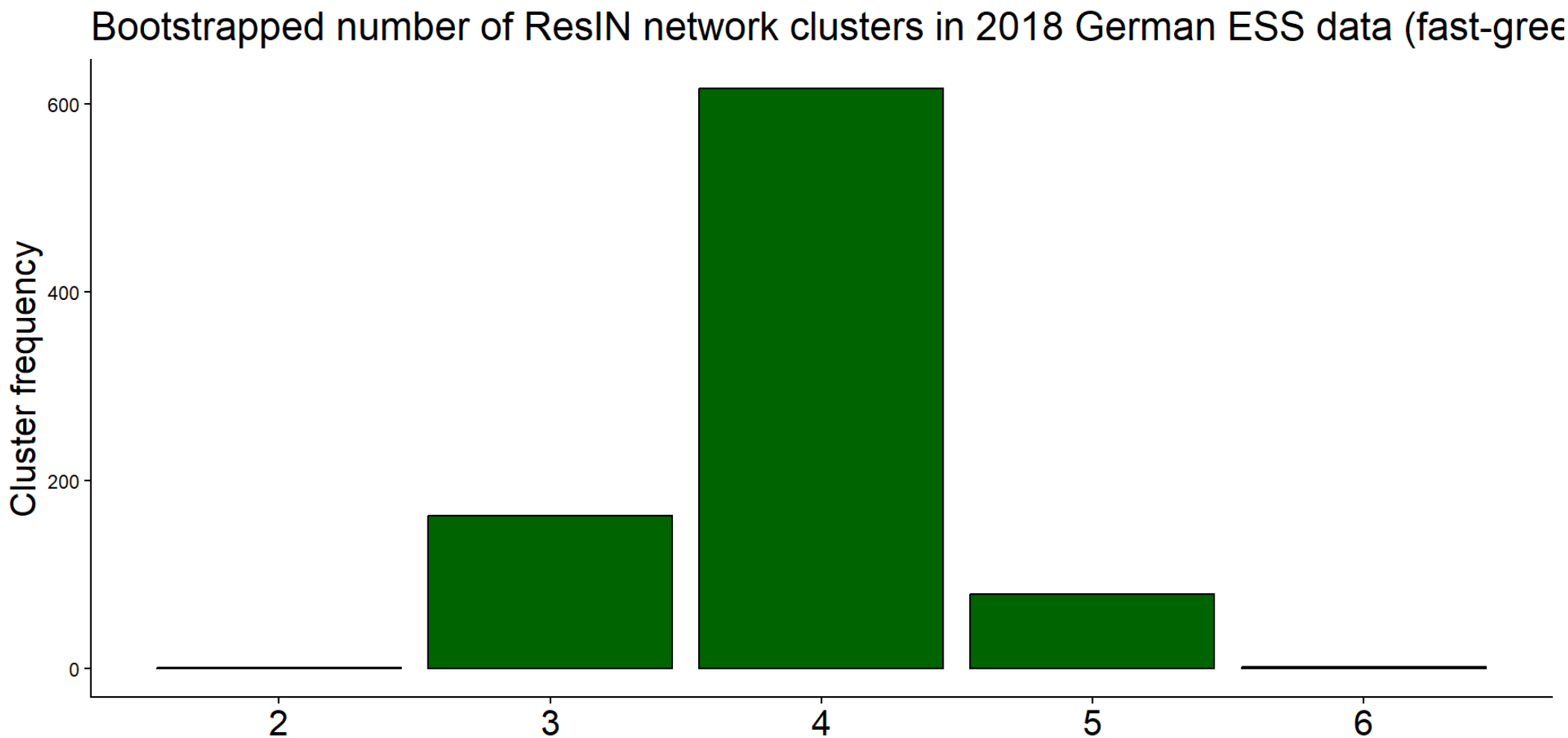
```
## 3.2: Preparing the bootstraping proceedure by supplying the input parameters
res_outDE_prepped <- ResIN_boots_prepare(res_outDE, n = 1000, boots_type = "resamp

## 3.3. Executing the proceedure (using parallel processing here)
# res_outDE_executed <- ResIN_boots_execute(res_outDE_prepped, parallel = T)

## 3.4. Extracting the detected communities with the what = "cluster" argument
# res_outDE_summarized <- ResIN_boots_extract(res_outDE_executed, what = "cluster"
# saveRDS(res_outDE_summarized, "res_outDE_summarized.RDS")
res_outDE_summarized <- readRDS("res_outDE_summarized.RDS")
```

Exercise 2) Possible solution

```r
## Unlisting the max number of detected cluster
max_cl_DE <- unlist(lapply(res_outDE_summarized, function(x) max(unique(x)))) %>%

## Creating a basic ggplot of the cluster distribution
res_plot_DE <- ggplot(max_cl_DE, aes(x = .))+
  geom_bar(fill = "darkgreen", color = "black")+
  labs(y = "Cluster frequency", x = "",
       title = "Bootstrapped number of ResIN network clusters in 2018 German ESS d
  theme_classic()+
  theme(axis.text.x = element_text(size = 16),
        axis.title.y = element_text(size = 16),
        plot.title = element_text(size = 18))
```

Exercise 2) Possible solution

```
1 res_plot_DE
```

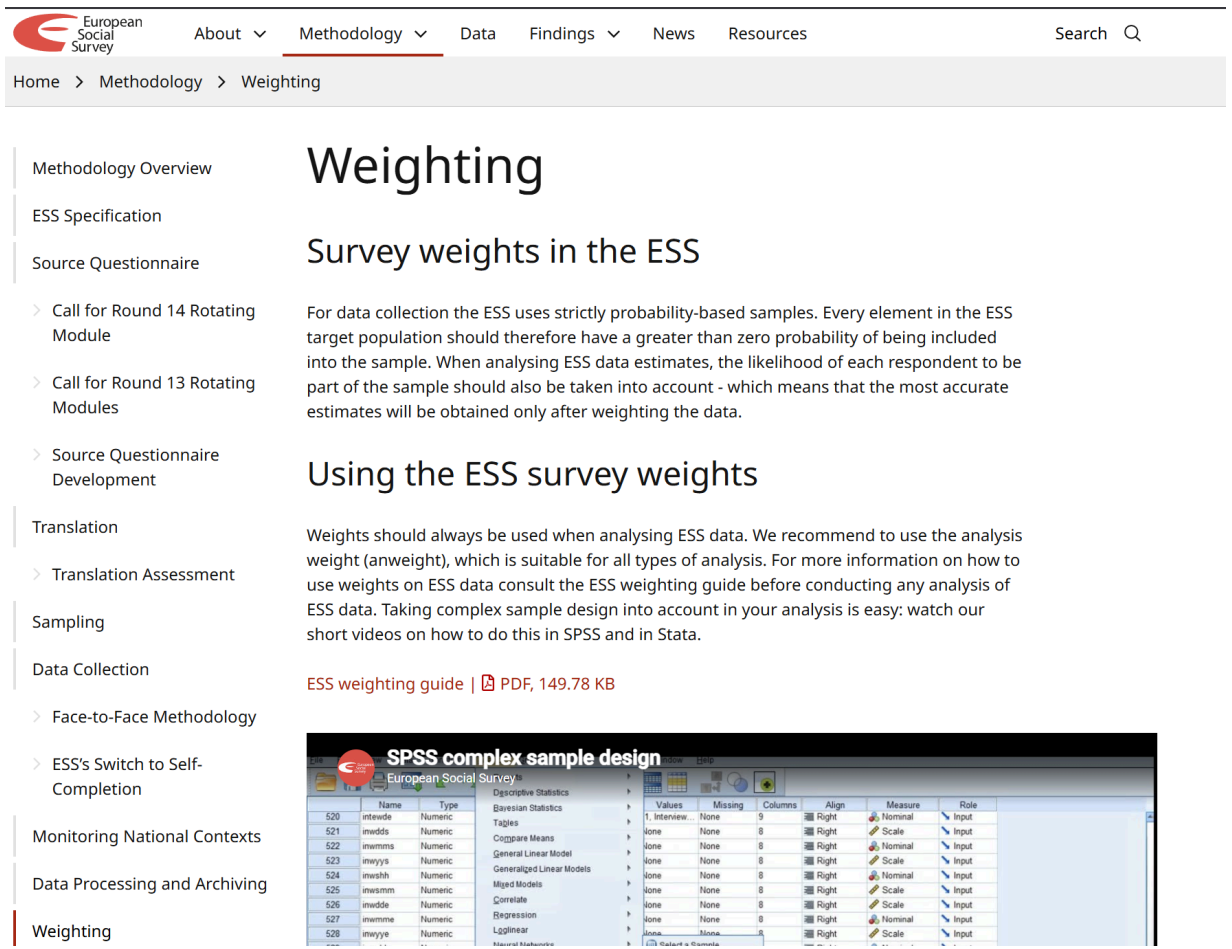


Day 3: Advanced features of the ResIN package

- Work with survey weights
- Network pruning
- Multimodal ResIN networks

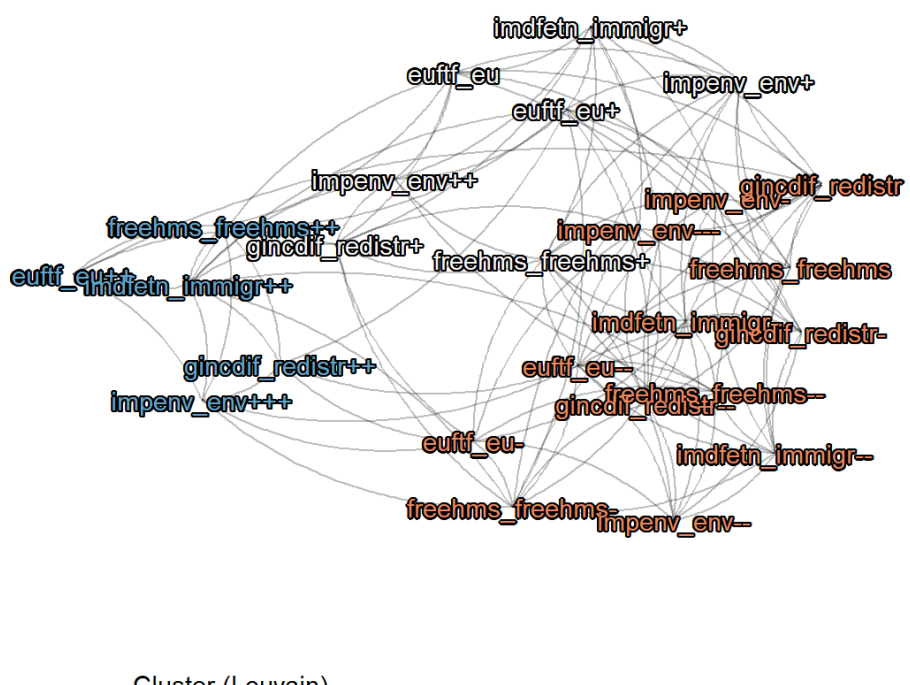
Work with survey weights

https://europeansocialsurvey.org/methodology/ess-methodology/data-processing-and-archiving/weighting



Work with survey weights

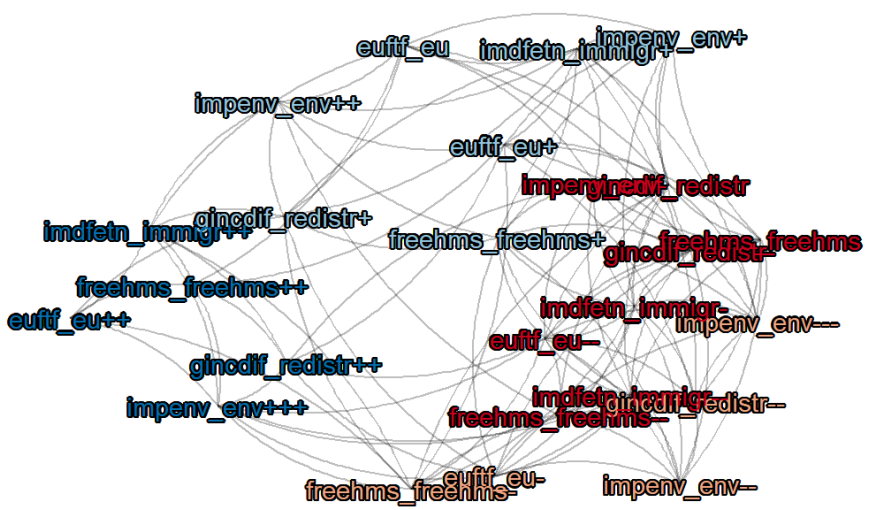
```
ResIN_ESS_NO <- ess_9 %>% filter(cntry == "NO") %>% ResIN(., node_vars = node_vars
                                 detect_clusters = TRUE,
                                 cluster_method = "cluster_louvain",
                                 plot_whichstat = "cluster",
                                 left_anchor = "impenv_env+++",
                                 plot_title = "Norway without weights", seed = 22
```



Work with survey weights

```
ResIN_ESS_NO <- ess_9 %>% filter(cntry == "NO") %>% ResIN(., node_vars = node_vars
                                 detect_clusters = TRUE, cluster_method = "cluster
                                 plot_whichstat = "cluster",
                                 left_anchor = "impenv_env+++",
                                 weights = "pspwght",  ## Note the inclusion of th
                                 plot_title = "Norway with weights", seed = 22)
```

Norway with weights



Cluster (Louvain)
Nodes per cluster: 1=8; 2=5; 3=7; 4=5

a 1 a 2 a 3 a 4

Bootstrapping with survey weights:

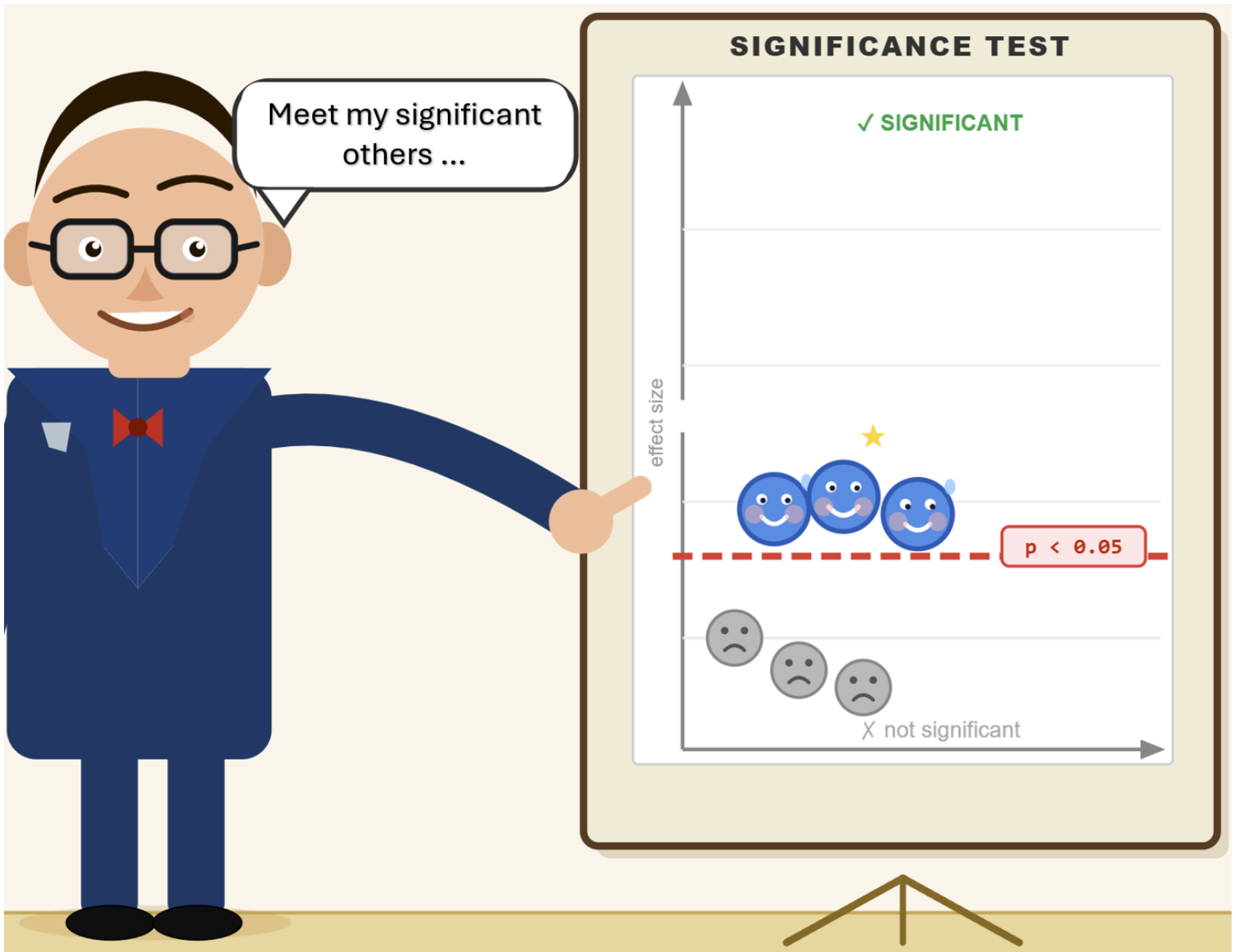
```
1 ResIN_boots_prepare(ResIN_ESS_NO, n = 1000, boots_type = "resample", weights = NUL
```

```
<ResIN bootstrap plan>
  type:          resample
  iterations:    1000
  resample_size:1406
  weights:       none
  save_input:    FALSE
  seed:          42
```

```
1 wghts_NO <- ess_9 %>% filter(cntry == "NO") %>% dplyr::select("pspwght")
2
3 ResIN_boots_prepare(ResIN_ESS_NO, n = 1000, boots_type = "resample", weights = wgh
```

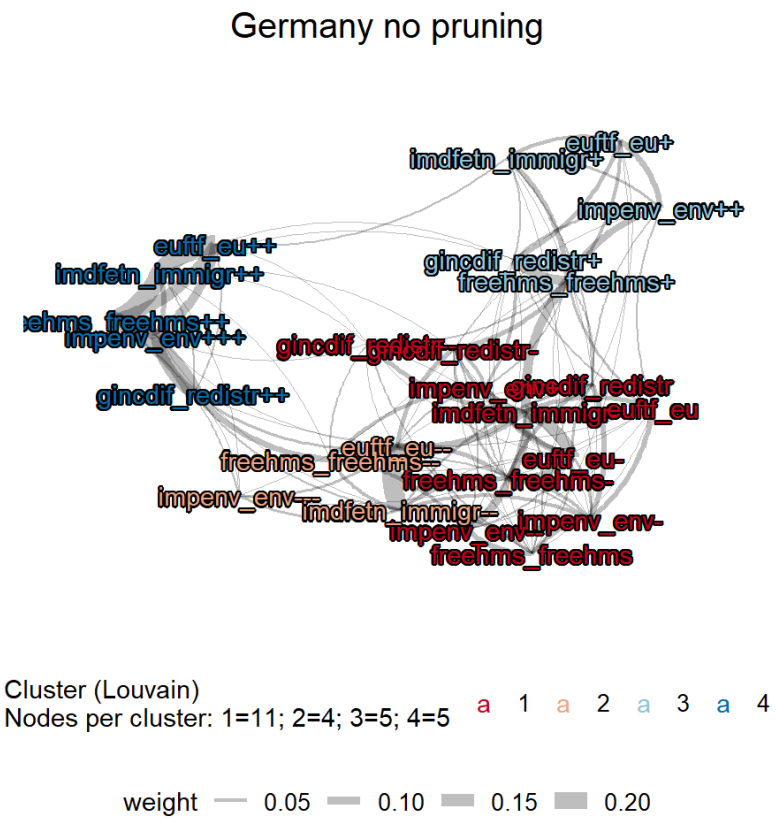
```
<ResIN bootstrap plan>
  type:          resample
  iterations:    1000
  resample_size:1406
  weights:       provided
  save_input:    FALSE
  seed:          42
```

Network pruning (p-value thresholding)



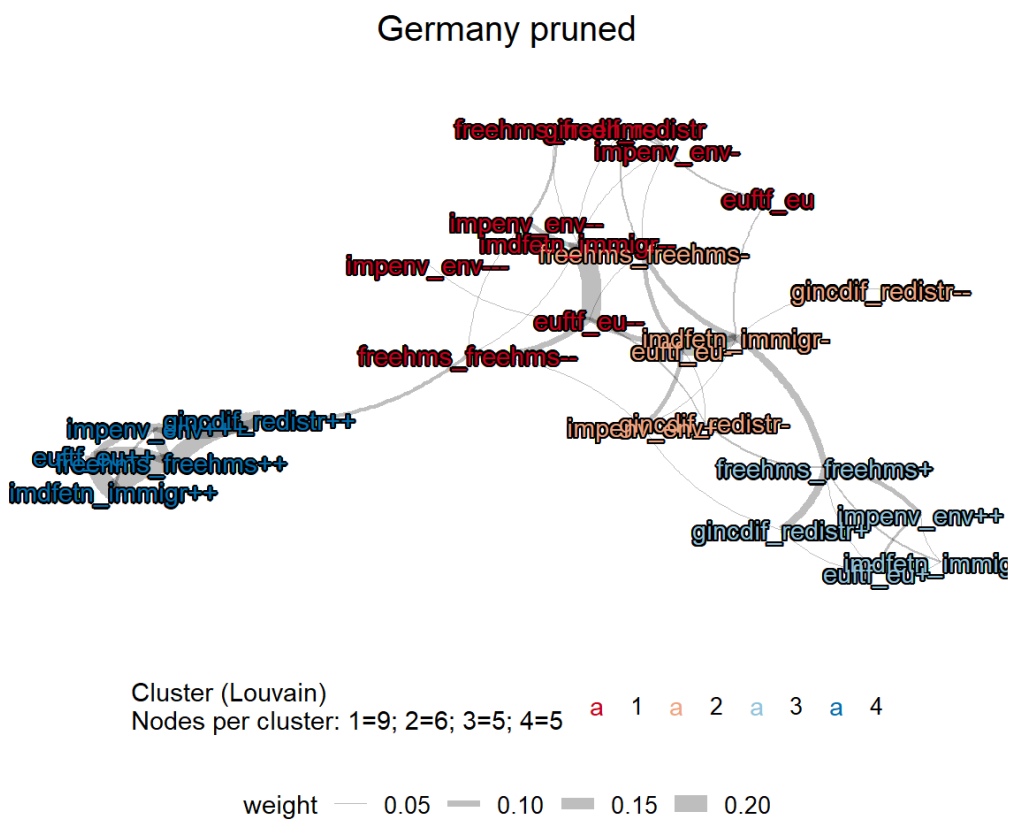
Network pruning (p-value thresholding)

```
ResIN_ESS_DE <- ess_9 %>% filter(cntry == "DE") %>% ResIN(., node_vars = node_vars
                                 detect_clusters = TRUE, cluster_method = "cluster
                                 plot_whichstat = "cluster", plot_edgestat = "weig
                                 left_anchor = "impenv_env+++",
                                 weights = "pspwght",
                                 plot_title = "Germany no pruning", seed = 22)
```



Network pruning (p-value thresholding)

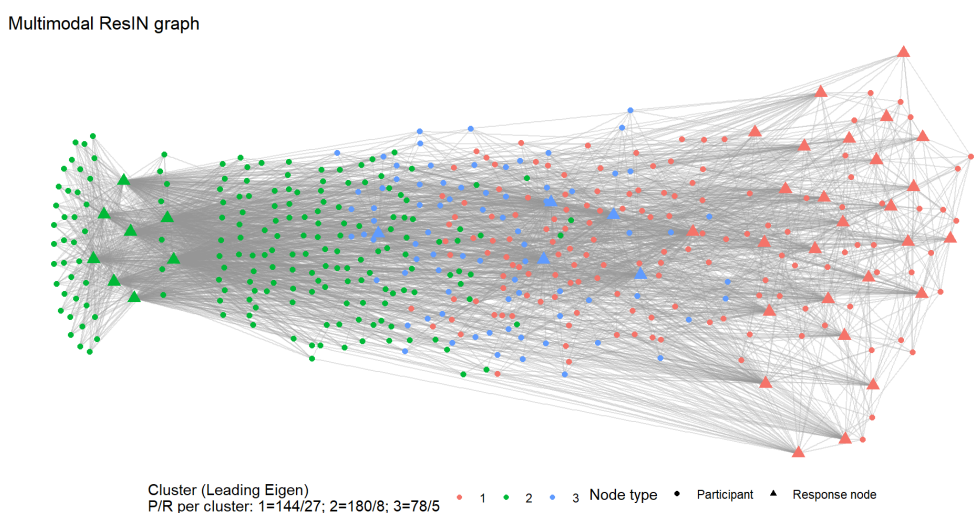
```
ResIN_ESS_DE <- ess_9 %>% filter(cntry == "DE") %>% ResIN(., node_vars = node_vars
                                 detect_clusters = TRUE, cluster_method = "cluster
                                 plot_whichstat = "cluster", left_anchor = "impenv
                                 remove_nonsignificant = TRUE, plot_edgestat = "we
                                 sign_threshold = 0.1,
                                 plot_title = "Germany pruned", seed = 22)
```



Multimodal ResIN networks (people + attitudes)

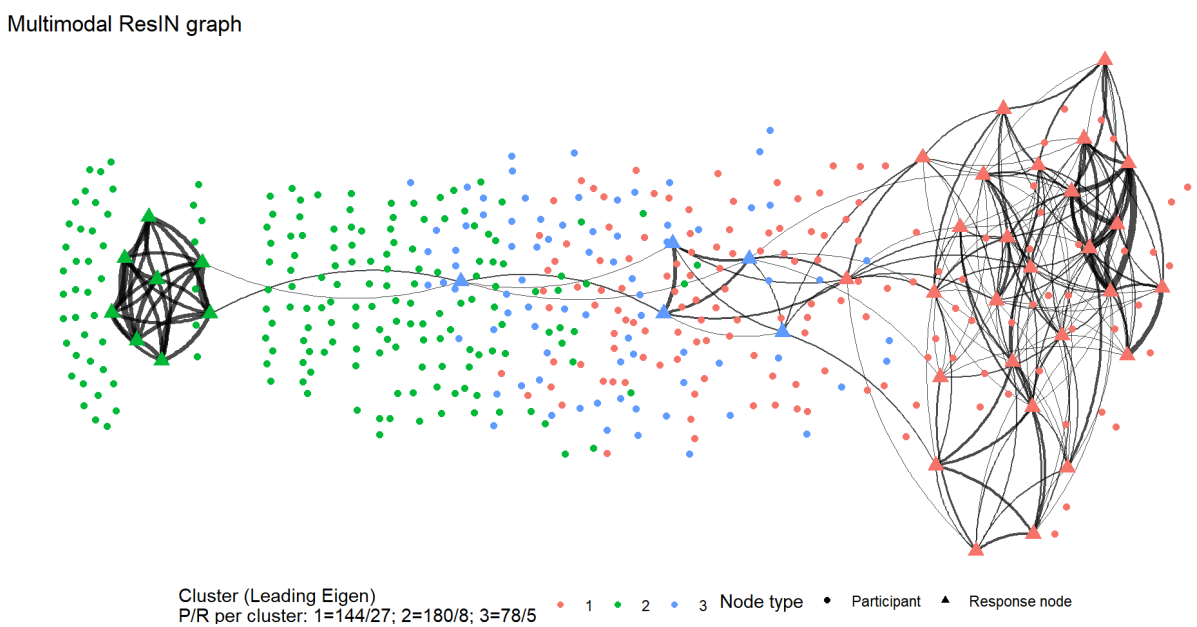
Multimodal ResIN networks (people + attitudes)

```r
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   multimodal = TRUE, detect_clusters = T,
                   remove_nonsignificant = TRUE,
                   sign_threshold = 0.1,
                   cluster_method = "cluster_leading_eigen",
                   generate_ggplot = T, plot_ggplot= F,
                   plot_whichstat = "cluster",
                   plot_edgestat = "weight",
                   left_anchor = "legal_abort_++", seed = 55,
                   multimodal_edge_overlay = "multimodal")

ResIN_out$multimodal_output$multimodal_ggraph
```



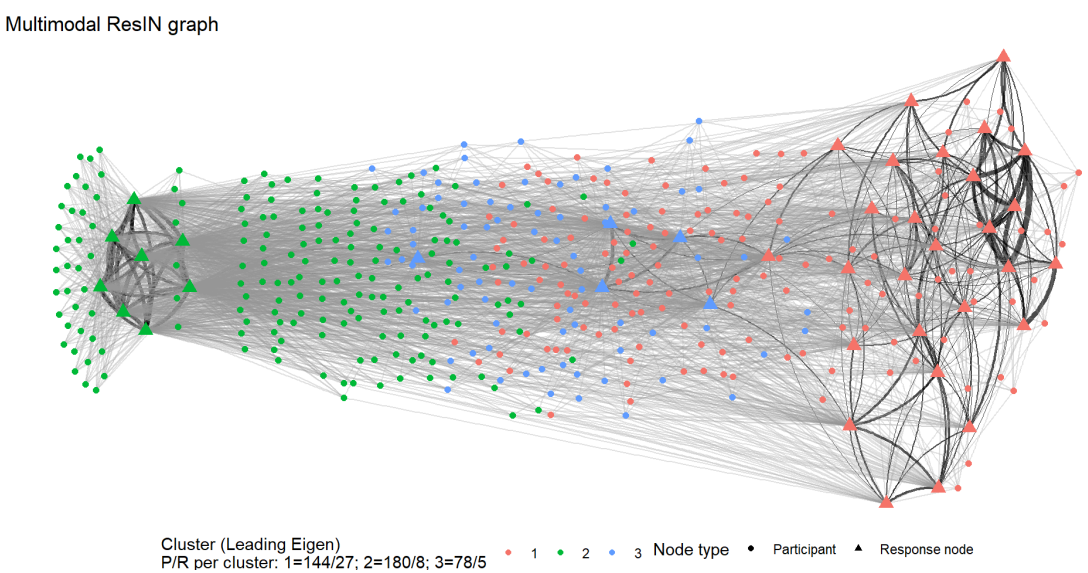
Multimodal ResIN networks (people + attitudes)

```r
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   multimodal = TRUE, detect_clusters = T,
                   remove_nonsignificant = TRUE,
                   sign_threshold = 0.1,
                   cluster_method = "cluster_leading_eigen",
                   generate_ggplot = T, plot_ggplot= F,
                   plot_whichstat = "cluster", plot_edgestat = "weight",
                   left_anchor = "legal_abort_++", seed = 55,
                   multimodal_edge_overlay = "ResIN")
ResIN_out$multimodal_output$multimodal_ggraph
```



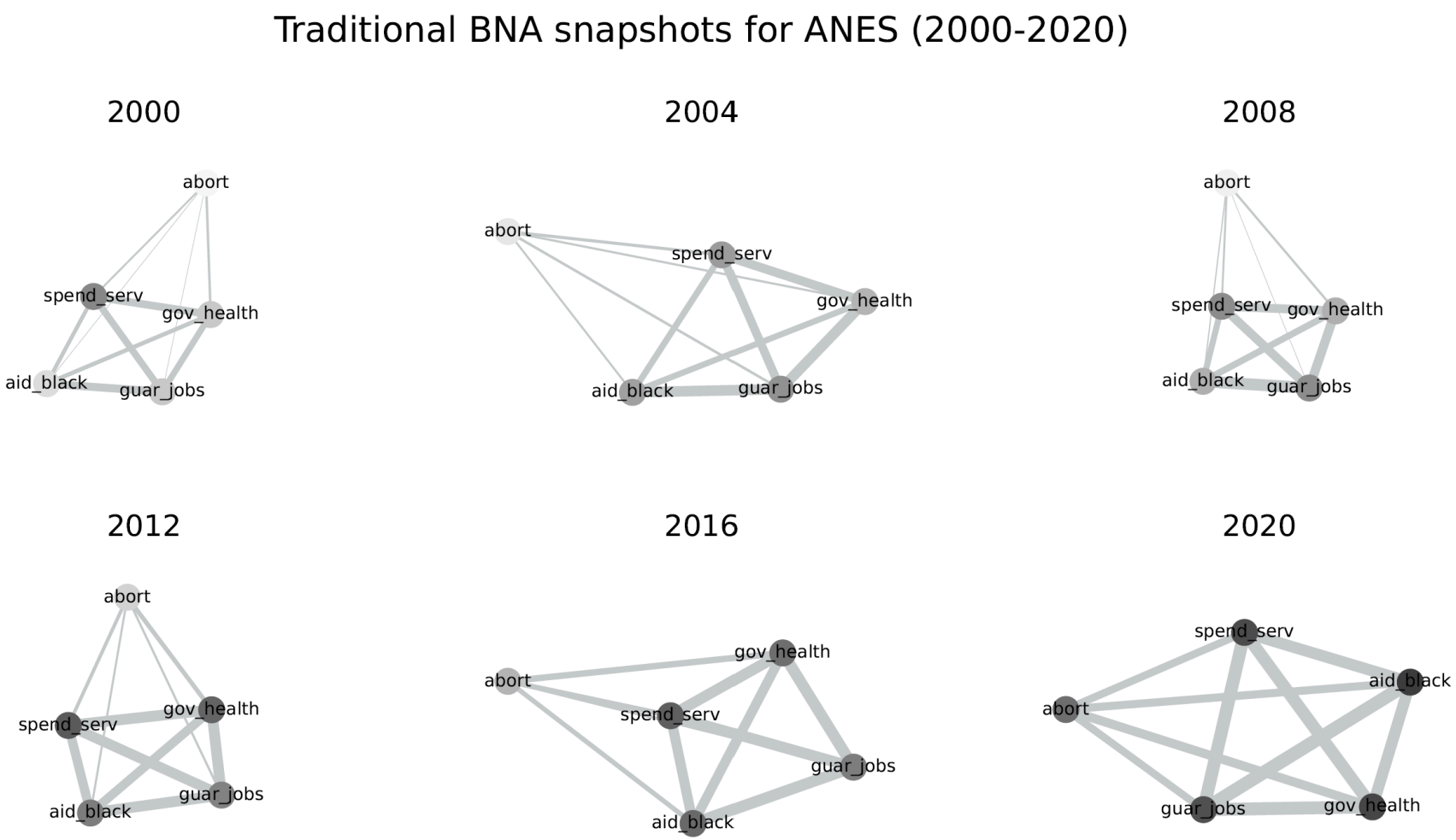
Multimodal ResIN networks (people + attitudes)

```
ResIN_out <- ResIN(df = Core_Items, node_vars = node_vars,
                   multimodal = TRUE, detect_clusters = T,
                   remove_nonsignificant = TRUE,
                   sign_threshold = 0.1,
                   cluster_method = "cluster_leading_eigen",
                   generate_ggplot = T, plot_ggplot= F,
                   plot_whichstat = "cluster",plot_edgestat = "weight",
                   left_anchor = "legal_abort_++", seed = 55,
                   multimodal_edge_overlay = "both")

ResIN_out$multimodal_output$multimodal_ggraph
```



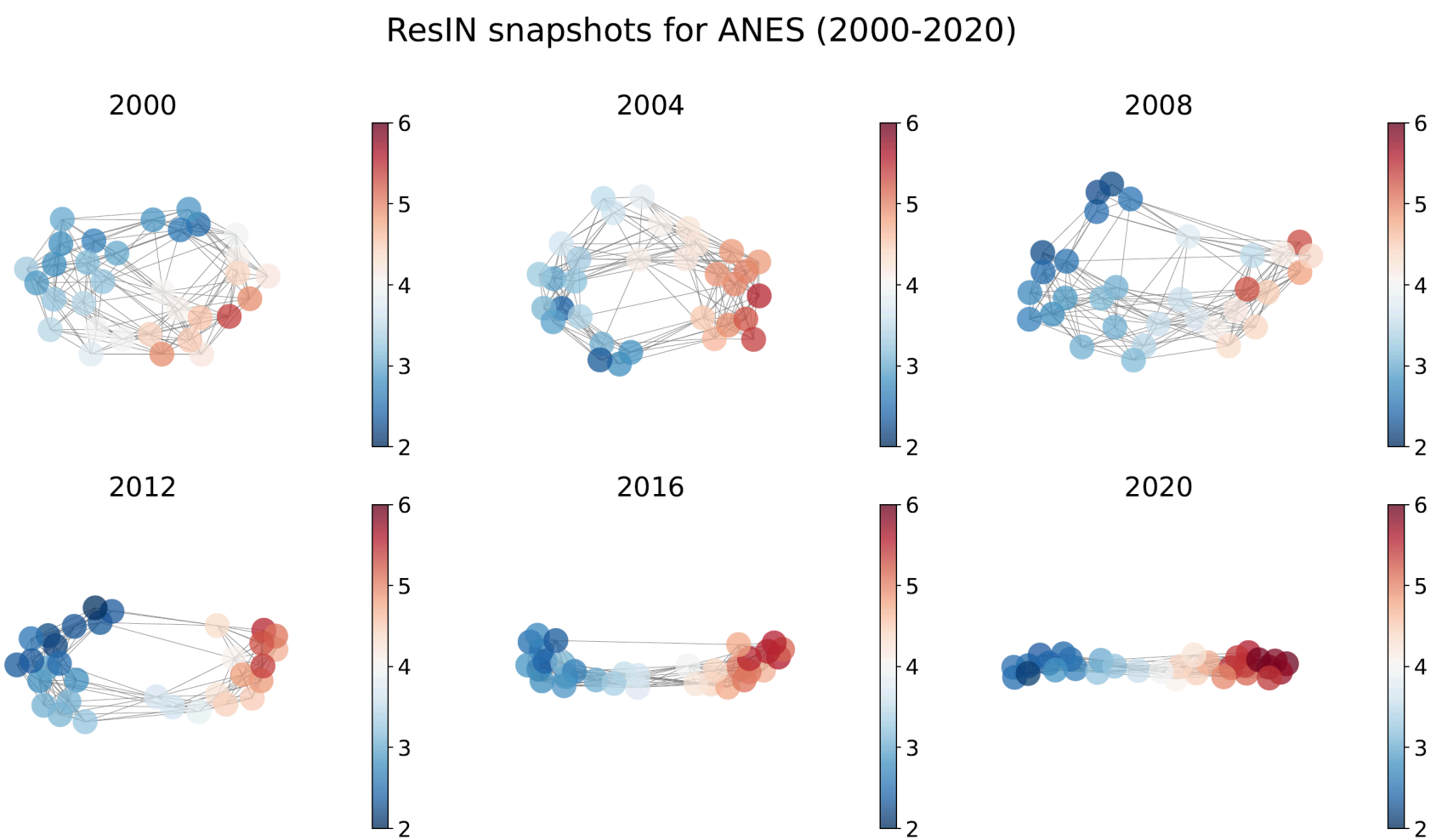
More ResIN: Differential belief system centrality

- Chen et.al. 2025



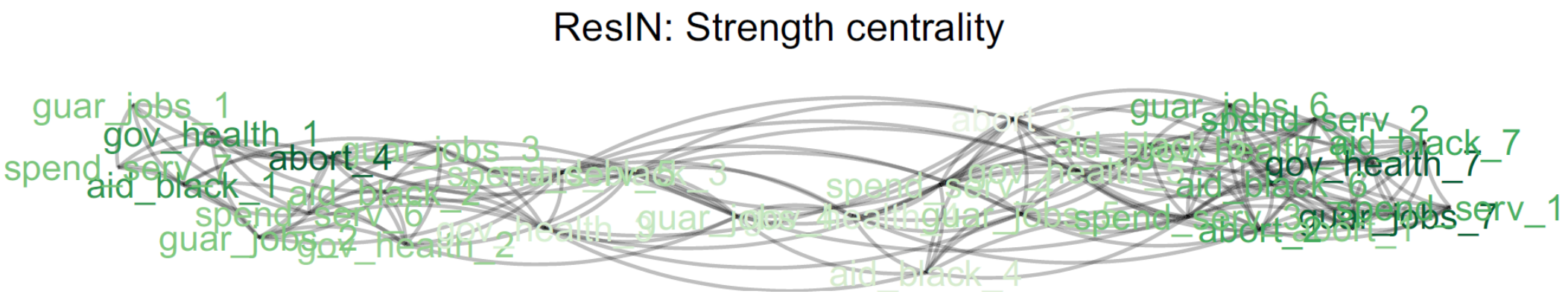
More ResIN: Differential belief system centrality

- Chen et.al. 2025



More ResIN: Differential belief system centrality

- Chen et.al. 2025



ResIN - still lost?

- Further theoretical and tutorial resources available at:
  - Carpentras et.al. 2024
  - Warncke et.al. 2026
  - https://pwarncke77.github.io/ResIN/
  - https://www.resinmethod.net/
  - https://www.youtube.com/watch?v=vIqUcI6GuzE