Online Appendix

D Additional Simulations

D.1 Finite Sample Performance Using ARD

In this section, we study the small sample performance of the estimator presented in Section 3 when the researcher only has access to ARD (as in Example 4). First, we simulate network data using the model proposed by Breza et al. (2020) and simulate outcomes using the linear-in-means model (1) conditional on the simulated networks. Second, we estimate the network formation model using the Bayesian estimator proposed by Breza et al. (2020) (yielding $\hat{\boldsymbol{\rho}}_B$) and using the classical estimator proposed by Alidaee et al. (2020) (yielding $\hat{\boldsymbol{\rho}}_A$). Third, we estimate the linear-in-means model using the estimators presented in Proposition 2 and Theorem 1 based on $\hat{\boldsymbol{\rho}}_A$ and $\hat{\boldsymbol{\rho}}_B$.

Recall that

$$P(a_{ij} = 1) \propto \exp\{\nu_i + \nu_j + \zeta \mathbf{z}_i' \mathbf{z}_j\}, \tag{10}$$

where ν_i, ν_j, ζ, $\mathbf{z}_i$, and $\mathbf{z}_j$ are not observed by the econometrician but follow parametric distributions. We refer the interested reader to McCormick and Zheng (2015), Breza et al. (2020), and Breza et al. (2019) for a formal discussion of the model, including its identification and consistent estimation.

To study the finite sample performance of our instrumental strategy in this context, we simulate 20 groups, each having 250 individuals. Within each subpopulation, we simulate the ARD responses and a series of observable characteristics. The details of the Monte Carlo simulations can be found below in the Online Appendix F.

Importantly, the model in (10) is based on a single population framework. Thus, the network formation model must be estimated separately for each of the 20 groups. With only 250 individuals in each group, we therefore expect significant small-sample bias.

We contrast the estimator proposed by Breza et al. (2020) with that of Alidaee et al. (2020). Whereas Breza et al. (2020) present a parametric Bayesian estimator, Alidaee et al. (2020) propose a (nonparametric) penalized regression based on a low-rank assumption. One main advantage of the estimator proposed in Alidaee et al. (2020) is that it allows for a wider class of model and ensures that the estimation is fast and easily implementable.[31] Note, however, that their method only yields a consistent estimator of $\hat{P}(\mathbf{A})$ if the true network is effectively low rank.

Very intuitively, the low-rank assumption implies that linking probabilities were generated from a small number of parameters. Importantly, the model (10) is not necessarily low rank; for example, if the individuals' latent positions (i.e. the $\mathbf{z}_i$'s) are uniformly distributed, then the model may not be low rank and the method proposed by Alidaee et al. (2020) would perform poorly. If, however, individuals' latent positions are located around a few focal points, then the model might be low-rank because knowledge of these focal points may have high predictive power.

We compare the performance of both estimators as we vary the concentration parameter (that is, κ; see below in the Online Appendix F for details). This has the effect of changing the *effective rank* of the linking probabilities: increasing κ decreases the effective rank.[32] We therefore expect the estimator proposed by Alidaee et al. (2020) to perform better for larger values of κ.

A summary of the results is presented in Tables D.1 and D.2. The complete results can be found in Tables D.3 and D.4. Table D.1 presents the results for the special case where **GX** are observed in the data. The table displays the performance of our simulated GMM (see Corollary 1) when the network formation model is estimated by Breza et al. (2020) and Alidaee et al. (2020).

When $\kappa = 0$, the network formation is not low rank. This disproportionately affects the estimator of Alidaee et al. (2020). When $\kappa = 15$, the estimators proposed

[31] The authors developed user-friendly packages in R and Python. See Alidaee et al. (2020) for links and details.

[32] We refer the interested reader to Alidaee et al. (2020) for a formal discussion of the effective rank and its importance for their estimator.

by Breza et al. (2020) and Alidaee et al. (2020) perform similarly.

We now turn to the more general case where $\mathbf{GX}$ are not observed. Table D.2 presents the performance of our SGMM estimator (Theorem 1) when the network formation process is estimated using the estimators proposed by Breza et al. (2020) and Alidaee et al. (2020) and when we assume that the researcher knows the true distribution of the network.

We see that the performance of our estimator is strongly affected by the quality of the first-stage network formation estimator. When based on either the estimator proposed by Breza et al. (2020) or Alidaee et al. (2020), for $\kappa = 0$ or $\kappa = 15$, our SGMM estimator performs poorly.

The poor performance of our SGMM estimator in a context where both $\mathbf{Gy}$ and $\mathbf{GX}$ are unobserved was anticipated. This occurs for two main reasons. *First*, the consistency of the network formation estimator in Breza et al. (2019) holds as the size of each subpopulation goes to infinity, whereas the consistency of our estimator holds as the number of (bounded) subpopulations goes to infinity. This should affect the performance of our estimator, when based on *estimated* network formation models but not when based on the true distribution of the network.

Second, as discussed in Example 4, ARD provides very little information about the realized network structure in the data (as opposed to censoring issues, for example; see Example 2). Then, if the true distribution is vague in the sense that most predicted probabilities are away from 0 or 1, we expect the estimation to be imprecise. This is what happens when $\kappa = 15$, where our estimation based on the true distribution of the network is very imprecise in a context where the network affects the outcome through both $\mathbf{Gy}$ and $\mathbf{GX}$.

In the next section, we present a likelihood-based estimator, which uses more information on the data-generating process of the outcome to improve the precision of the estimation.

Table D.1: Simulation results with ARD and observed $\boldsymbol{GX}$

Parameter	Breza et al.		Alidaee et al.	
	Mean	Std. Dev	Mean	Std. Dev
		$\kappa = 0$		
		SGMM (Corollary 1)		
$\alpha = 0.4$	0.392	(0.010)	0.492	(0.057)
		$\kappa = 15$		
		SGMM (Corollary 1)		
$\alpha = 0.4$	0.400	(0.009)	0.428	(0.009)

Note: For each case, we generated 20 independent subnetworks of 250 individuals each. In each subnetwork, the spherical coordinates of individuals were generated from a von Mises–Fisher distribution with a location parameter $(1, 0, 0)$ and intensity parameter κ. Details of the simulation exercise can be found below in the Online Appendix F. Predicted probabilities were computed using the mean of the posterior distribution. We chose the weight associated with the nuclear norm penalty to minimize the RMSE through cross-validation. This value of $\lambda = 600$ is smaller than the recommended value in Alidaee et al. (2020). See Table D.3 for the estimated values of the other parameters.

Table D.2: Simulation results with ARD and unobserved $\boldsymbol{GX}$ (SGMM, Theorem 1)

Parameter	Breza et al.		Alidaee et al.		True distribution	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
			$\kappa = 0$			
$\alpha = 0.4$	0.717	(0.463)	0.700	(0.268)	0.400	(0.056)
			$\kappa = 15$			
$\alpha = 0.4$	0.603	(0.069)	0.870	(0.202)	0.434	(0.394)

Note: For each case, we generated 20 independent subnetworks of 250 individuals each. In each subnetwork, the spherical coordinates of individuals were generated from a von Mises–Fisher distribution with a location parameter $(1, 0, 0)$ and intensity parameter κ. Details of the simulation exercise can be found below in the Online Appendix F. Predicted probabilities were computed using the mean of the posterior distribution. See Table D.4 for the estimated values of the other parameters. Instruments were built using only second-degree peers, i.e. $\mathbf{G^2X}$.

Table D.3: Simulation results with ARD and observed $\boldsymbol{GX}$

Parameter	Breza et al.		Alidaee et al.	
	Mean	Std. Dev	Mean	Std. Dev
	SMM, $\kappa = 0$, $N = 250$, $M = 20$			
$\alpha = 0.4$	0.392	(0.01)	0.492	(0.057)
$\beta_1 = 1$	1.001	(0.004)	1.002	(0.009)
$\beta_2 = 1.5$	1.500	(0.007)	1.496	(0.016)
$\gamma_1 = 5$	5.013	(0.034)	3.884	(0.295)
$\gamma_2 = -3$	-2.993	(0.052)	-4.048	(0.354)
	SMM, $\kappa = 15$, $N = 250$, $M = 20$			
$\alpha = 0.4$	0.400	(0.009)	0.428	(0.009)
$\beta_1 = 1$	1.000	(0.004)	0.999	(0.004)
$\beta_2 = 1.5$	1.500	(0.008)	1.499	(0.008)
$\gamma_1 = 5$	4.996	(0.034)	4.677	(0.034)
$\gamma_2 = -3$	-3.005	(0.055)	-3.387	(0.055)

Note: In each subnetwork, the spherical coordinates of individuals are generated from a von Mises–Fisher distribution with a location parameter $(1, 0, 0)$ and intensity parameter κ. Predicted probabilities are computed using the mean of the posterior distribution. We chose the weight associated with the nuclear norm penalty to minimize the RMSE through cross-validation. This value of $\lambda = 600$ is smaller than the recommended value in Alidaee et al. (2020). Instruments are build using only second-degree peers, i.e. $\mathbf{G^2X}$.

Table D.4: Simulation results with ARD and unobserved $\boldsymbol{GX}$

Parameter	Breza et al.		Alidaee et al.		True distribution	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
	SMM, $\kappa = 0$, $N = 250$, $M = 20$					
$\alpha = 0.4$	0.717	(0.463)	0.700	(0.268)	0.400	(0.056)
$\beta_1 = 1$	0.988	(0.022)	0.995	(0.017)	1.000	(0.015)
$\beta_2 = 1.5$	1.505	(0.03)	1.503	(0.029)	1.501	(0.021)
$\gamma_1 = 5$	1.778	(4.473)	1.512	(2.37)	4.991	(0.455)
$\gamma_2 = -3$	-2.205	(1.24)	-0.405	(0.955)	-3.005	(0.287)
	SMM, $\kappa = 15$, $N = 250$, $M = 20$					
$\alpha = 0.4$	0.603	(0.069)	0.870	(0.202)	0.434	(0.394)
$\beta_1 = 1$	0.989	(0.014)	0.984	(0.015)	0.998	(0.021)
$\beta_2 = 1.5$	1.504	(0.029)	1.509	(0.029)	1.501	(0.023)
$\gamma_1 = 5$	2.866	(0.566)	0.246	(1.973)	4.638	(3.887)
$\gamma_2 = -3$	-2.458	(0.379)	-1.539	(0.602)	-2.913	(1.037)

Note: In each subnetwork, the spherical coordinates of individuals are generated from a von Mises–Fisher distribution with a location parameter $(1, 0, 0)$ and intensity parameter κ. Predicted probabilities are computed using the mean of the posterior distribution. We chose the weight associated with the nuclear norm penalty to minimize the RMSE through cross-validation. This value of $\lambda = 600$ is smaller than the recommended value in Alidaee et al. (2020). Instruments are build using only second-degree peers, i.e. $\mathbf{G^2X}$.

E Additional Results

E.1 Simple estimators

Proposition 1. *[Conditions] Suppose that* $\mathbf{GX}$ *and* $\mathbf{Gy}$ *are observed. Let* $\boldsymbol{H}$ *be a matrix such that (1) at least one column of* $\mathbf{H}^k\mathbf{X}$ *is (strongly) correlated with* $\mathbf{Gy}$*, conditional on* $[\mathbf{1}, \mathbf{X}, \mathbf{GX}]$ *for* $k \geq 2$*, and (2)* $\mathbb{E}[\boldsymbol{\varepsilon}|\mathbf{X}, \mathbf{H}] = \mathbf{0}$*. Finally, define the matrix* $\mathbf{Z} = [\mathbf{1}, \mathbf{X}, \mathbf{GX}, \mathbf{H}^2\mathbf{X}, \mathbf{H}^3\mathbf{X}...]$.

[Results] Then, under classical assumptions (e.g. Cameron and Trivedi (2005), Proposition 6.1), the (linear) GMM estimator based on the moment function $\frac{1}{N}\sum_i \mathbf{Z}_k'\boldsymbol{\varepsilon}_i$ *is consistent and asymptotically normally distributed with the usual asymptotic variance–covariance matrix.*

Condition (1) is the relevancy condition, whereas condition (2) is the exogeneity condition.[33] Although Proposition 1 holds for any matrix $\mathbf{H}$ such that conditions (1) and (2) hold, the most sensible example in our context is when $\mathbf{H}$ is constructed using a draw from $\hat{P}(\mathbf{G})$.

Importantly, the moment conditions remain valid even when the researcher uses a *mispecified* estimator of the distribution $P(\mathbf{G})$, as long as the specification error on $P(\mathbf{G})$ does not induce a correlation with $\boldsymbol{\varepsilon}$.[34] This could be of great practical importance, especially if the estimation of $\hat{P}(\mathbf{G})$ suffers from a small sample bias.

We present a simple, asymptotically biased, linear GMM estimator. The presentation of such this estimator is useful for two reasons. First, simulations show that the asymptotic bias turns out to be negligible in many cases, especially for moderate group sizes. Moreover, the estimator is computationally attractive because the estimator can be written in a closed form. Second, the estimator helps to understand

[33] Although (for simplicity) in Proposition 1, we use the entire matrix $\mathbf{X}$ to generate the instruments $\mathbf{HX}$, in practice, one should avoid including instruments (i.e. columns of $\mathbf{HX}$) that are weakly correlated with $\mathbf{Gy}$.

[34] We would like to thank Chih-Sheng Hsieh and Arthur Lewbel for discussions on this important point.

the logic underlying the estimator defined in Theorem 1, which we can view as a bias-corrected version of this simple linear GMM estimator. Proposition 2 formalizes.

Proposition 2. *[Conditions] Assume that* $\mathbf{GX}$ *is observed. Let* $\ddot{\mathbf{S}} = [\mathbf{1}, \mathbf{X}, \mathbf{GX}, \ddot{\mathbf{G}}\mathbf{X}, \ddot{\mathbf{G}}\mathbf{y}]$ *and* $\dot{\mathbf{Z}} = [\mathbf{1}, \mathbf{X}, \mathbf{GX}, \ddot{\mathbf{G}}\mathbf{X}, \dot{\mathbf{G}}^2\mathbf{X}, \dot{\mathbf{G}}^3\mathbf{X}, ...]$*. Denote by* $\hat{\boldsymbol{\theta}}$ *the linear GMM estimator based on the (pseudo) moment function* $\frac{1}{N}\sum_i \dot{\mathbf{Z}}_i[\boldsymbol{\eta}_i + \boldsymbol{\varepsilon}_i]$ *and define the sensitivity matrix*

$\mathbf{M}_N = [(\ddot{\mathbf{S}}'\dot{\mathbf{Z}}/N)\mathbf{W}(\dot{\mathbf{Z}}'\ddot{\mathbf{S}}/N)]^{-1}(\ddot{\mathbf{S}}'\dot{\mathbf{Z}}/N)\mathbf{W}$.

[Result] Then, under classical assumptions (see proof), the asymptotic bias of $\hat{\boldsymbol{\theta}}$ *is given by* $\alpha\mathbf{M}_0 \operatorname{plim}[\dot{\mathbf{Z}}'(\mathbf{G} - \ddot{\mathbf{G}})\mathbf{y}/N]$*. Moreover, letting* $\mathbf{W} = \mathbf{I}$ *minimizes the asymptotic bias in the sense of minimizing the Frobenius norm of* $\mathbf{M}$*.*

Although there are no obvious way to obtain a consistent estimate of the asymptotic bias (because $\mathbf{y}$ is a function of $\mathbf{G}$), simulations show that the bias is very small in practice.[35]

The intuition behind Proposition 2 comes from the literature on error-in-variable models with repeated observations (e.g. Bound et al. (2001)). The instrumental variable uses two independent draws from the (estimated) distribution of the true network. One draw is used to proxy the unobserved variable (i.e. $\mathbf{Gy}$), whereas the other is used to proxy the instrument (i.e. $\mathbf{GX}$). This approach greatly reduces the bias compared with a situation in which only one draw would be used.[36]

The argument in Proposition 2 is very similar to the one in Andrews et al. (2017), although here perturbation with respect to the true model is not *local*.[37] We also show that we expect $\mathbf{W} = \mathbf{I}$ to minimize the asymptotic bias. Our result therefore provides a theoretical justification for the simulations in Onishi and Otsu (2021) who show that using the identity matrix to weight the moments greatly reduces the bias in the context studied by Andrews et al. (2017).

[35] See Section D.1 below.

[36] Simulations available uppon request.

[37] See page 1562 in Andrews et al. (2017).

E.1.1 Proof of Proposition 2

Part 1: Asymptotic bias Define $\boldsymbol{\Delta} = \mathbf{G} - \ddot{\mathbf{G}}$ and rewrite $\boldsymbol{\eta} = \alpha[\mathbf{G} - \ddot{\mathbf{G}}]\mathbf{y} = \alpha\boldsymbol{\Delta}\mathbf{y}$. Let $\boldsymbol{\theta}_0^+ = (\boldsymbol{\theta}_0', 0, 0, \dots, 0)'$ be the true value the parameter when regressors are defined as $\ddot{\mathbf{S}} = [\mathbf{1}, \mathbf{X}, \mathbf{G}\mathbf{X}, \ddot{\mathbf{G}}\mathbf{X}, \ddot{\mathbf{G}}\mathbf{y}]$. The Linear GMM estimator can be written as

$$\hat{\boldsymbol{\theta}} = \left[\left(\frac{\ddot{\mathbf{S}}'\dot{\mathbf{Z}}}{N}\right)\left(\frac{\dot{\mathbf{Z}}'\ddot{\mathbf{S}}}{N}\right)\right]^{-1}\left(\frac{\ddot{\mathbf{S}}'\dot{\mathbf{Z}}}{N}\right)\left(\frac{\dot{\mathbf{Z}}'\ddot{\mathbf{S}}}{N}\boldsymbol{\theta}_0^+ + \frac{\dot{\mathbf{Z}}'[\boldsymbol{\eta}+\boldsymbol{\varepsilon}]}{N}\right),$$
$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_0^+ + \mathbf{M}_N\left(\frac{\dot{\mathbf{Z}}'\boldsymbol{\eta}+\dot{\mathbf{Z}}'\boldsymbol{\varepsilon}}{N}\right),$$

where $\mathbf{M}_N = \left[\left(\frac{\ddot{\mathbf{S}}'\dot{\mathbf{Z}}}{N}\right)\left(\frac{\dot{\mathbf{Z}}'\ddot{\mathbf{S}}}{N}\right)\right]^{-1}\left(\frac{\ddot{\mathbf{S}}'\dot{\mathbf{Z}}}{N}\right)$.
The asymptotic bias of $\hat{\boldsymbol{\theta}}$ is then given by $\operatorname{plim}\boldsymbol{\theta}_N^* = \alpha\mathbf{M}_0 \operatorname{plim}[\dot{\mathbf{Z}}'(\mathbf{G} - \ddot{\mathbf{G}})\mathbf{y}/N]$.
Part 2: Choice of W Let $\mathbf{K} = \dot{\mathbf{Z}}'\boldsymbol{\Delta}\mathbf{G}^2/N$, if $\boldsymbol{\gamma} = \mathbf{0}$, and $\mathbf{K} = \dot{\mathbf{Z}}'\boldsymbol{\Delta}/N$ otherwise. Consider $\|\mathbf{MK}\|_F = \sqrt{trace(\mathbf{K}'\mathbf{M}'\mathbf{MK})} = \sqrt{trace(\mathbf{KK}'\mathbf{M}'\mathbf{M})}$. We have

$$(1/N^2)\mathbf{MM}' = [\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\mathbf{W}\dot{\mathbf{Z}}'\ddot{\mathbf{S}}]^{-1}\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\mathbf{WW}\dot{\mathbf{Z}}'\ddot{\mathbf{S}}[\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\mathbf{W}\dot{\mathbf{Z}}'\ddot{\mathbf{S}}]^{-1}.$$

Let $\mathbf{W} = \mathbf{C}'\mathbf{C}$ and let $\mathbf{B} = \ddot{\mathbf{S}}'\dot{\mathbf{Z}}\mathbf{C}'$. We have

$$(1/N^2)\mathbf{MM}' = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{CC}'\mathbf{B}(\mathbf{B}'\mathbf{B})^{-1}.$$

Now, define $\mathbf{J}' = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{C} - (\mathbf{B}'(\mathbf{C}')^{-1}\mathbf{C}^{-1}\mathbf{B})^{-1}\mathbf{B}'(\mathbf{C}')^{-1}$. We have

$$(1/N^2)\mathbf{MM}' = \mathbf{J}'\mathbf{J} + (\mathbf{B}'(\mathbf{C}')^{-1}\mathbf{C}^{-1}\mathbf{B})^{-1} = \mathbf{J}'\mathbf{J} + (\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\dot{\mathbf{Z}}'\ddot{\mathbf{S}})^{-1}.$$

Therefore, we have

$$(1/N^2)\|\mathbf{M}\|_F = \sqrt{trace(\mathbf{J}'\mathbf{J} + (\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\dot{\mathbf{Z}}'\ddot{\mathbf{S}})^{-1})} = \sqrt{trace(\mathbf{J}'\mathbf{J}) + trace((\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\dot{\mathbf{Z}}'\ddot{\mathbf{S}})^{-1})}.$$

When $\mathbf{W} = \mathbf{I}$, we have that $\mathbf{J} = \mathbf{0}$ and the Frobenius norm of $\mathbf{M}$ is given by $N^2\sqrt{trace((\ddot{\mathbf{S}}'\dot{\mathbf{Z}}\dot{\mathbf{Z}}'\ddot{\mathbf{S}})^{-1})}$.

E.1.2 Bias correction for GX observed

Let us replace $\ddot{\mathbf{V}}$ with $\mathbf{V}$ in (5). We have

$$\dot{\mathbf{Z}}_i^{(r)\prime}\left[(\mathbf{I}-\alpha\dddot{\mathbf{G}}^{(t)})_i\left((\mathbf{I}-\alpha\mathbf{G})^{-1}-(\mathbf{I}-\alpha\ddot{\mathbf{G}}^{(s)})^{-1}\right)\mathbf{V}\tilde{\boldsymbol{\theta}}\right]$$

which is equal to

$$\dot{\mathbf{Z}}_i^{(r)\prime}\left[(\mathbf{I}-\alpha\dddot{\mathbf{G}}^{(t)})_i\left((\mathbf{I}-\alpha\mathbf{G})^{-1}-(\mathbf{I}-\alpha\ddot{\mathbf{G}}^{(s)})^{-1}\right)((\mathbf{I}-\alpha\mathbf{G})\mathbf{y}+\boldsymbol{\varepsilon})\right],$$

which simplifies to

$$\dot{\mathbf{Z}}_i^{(r)\prime}(\mathbf{I}-\alpha\dddot{\mathbf{G}}^{(t)})_i(\mathbf{I}-\alpha\ddot{\mathbf{G}}^{(s)})^{-1}[\alpha(\mathbf{G}-\ddot{\mathbf{G}}^{(s)})]\mathbf{y}+\dot{\mathbf{Z}}_i^{(r)\prime}\left[(\mathbf{I}-\alpha\dddot{\mathbf{G}}^{(t)})_i\left((\mathbf{I}-\alpha\mathbf{G})^{-1}-(\mathbf{I}-\alpha\ddot{\mathbf{G}}^{(s)})^{-1}\right)\boldsymbol{\varepsilon}\right].$$

Note that the expectation of the term $\dot{\mathbf{Z}}_i^{(r)\prime}\left[(\mathbf{I}-\alpha\dddot{\mathbf{G}}^{(t)})_i\left((\mathbf{I}-\alpha\mathbf{G})^{-1}-(\mathbf{I}-\alpha\ddot{\mathbf{G}}^{(s)})^{-1}\right)\boldsymbol{\varepsilon}\right]$ is null.

E.1.3 Corollaries

Corollary 1. *Assume that* $\mathbf{GX}$ *is observed but that* $\mathbf{Gy}$ *is not observed. Let* $\dot{\mathbf{Z}}^{(s)}=[\mathbf{1},\mathbf{X},\mathbf{GX},\dot{\mathbf{G}}^{s}\mathbf{GX},(\dot{\mathbf{G}}^{(s)})^2\mathbf{GX},...]$, $\ddot{\mathbf{Z}}^{(s,t)}=[\mathbf{1},\mathbf{X},\ddot{\mathbf{G}}^{(t)}\mathbf{X},\dot{\mathbf{G}}^{s}\ddot{\mathbf{G}}^{(t)}\mathbf{X},(\dot{\mathbf{G}}^{(s)})^2\ddot{\mathbf{G}}^{(t)}\mathbf{X},...]$, $\mathbf{V}=[\mathbf{1},\mathbf{X},\mathbf{GX}]$, *and* $\ddot{\mathbf{V}}^{(t)}=[\mathbf{1},\mathbf{X},\ddot{\mathbf{G}}^{(t)}\mathbf{X}]$. *Then, the results from Theorem 1 hold for the following (simulated) moment function:*

$$\frac{1}{S}\sum_{s=1}^{S}\dot{\mathbf{Z}}_i^{(s)\prime}(\mathbf{I}-\alpha\dot{\mathbf{G}}^{(s)})_i\mathbf{y}-\frac{1}{ST}\sum_{t=1}^{T}\sum_{s=1}^{S}(\dot{\mathbf{Z}}^{(s)\prime}\mathbf{V}-\ddot{\mathbf{Z}}^{(s,t)\prime}\ddot{\mathbf{V}}^{(t)})_i\tilde{\boldsymbol{\theta}}$$
$$-\frac{1}{ST}\sum_{t=1}^{T}\sum_{s=1}^{S}\ddot{\mathbf{Z}}_i^{(s,t)\prime}(\mathbf{I}-\alpha\dot{\mathbf{G}}^{(s)})_i(\mathbf{I}-\alpha\ddot{\mathbf{G}}^{(t)})^{-1}\ddot{\mathbf{V}}^{(t)}\tilde{\boldsymbol{\theta}}. \qquad (11)$$

Corollary 2. *Assume that* $\mathbf{Gy}$ *is observed but that* $\mathbf{GX}$ *is not observed. Let* $\dot{\mathbf{Z}}^{(s)}=[\mathbf{1},\mathbf{X},\dot{\mathbf{G}}^{(s)}\mathbf{X},(\dot{\mathbf{G}}^{(s)})^2\mathbf{X},...]$, *and* $\ddot{\mathbf{V}}^{(t)}=[\mathbf{1},\mathbf{X},\ddot{\mathbf{G}}^{(t)}\mathbf{X}]$. *Then, the results from Theorem*

1 hold for the following (simulated) moment function:

$$\frac{1}{S}\sum_{s=1}^{S}\dot{\mathbf{Z}}_i^{(s)\prime}(\mathbf{I}-\alpha\mathbf{G})_i\mathbf{y}-\frac{1}{ST}\sum_{t=1}^{T}\sum_{s=1}^{S}\dot{\mathbf{Z}}_i^{(s)\prime}\ddot{\mathbf{V}}_i^{(t)}\tilde{\boldsymbol{\theta}} \tag{12}$$

F ARD Details

This section provides details about ARD simulation and model estimation using a MCMC method. We simulate the network for a population of 5000 individuals divided into $m = 20$ groups of $n = 250$ individuals. Within each group, the probability of a link is

$$P(a_{ij} = 1) \propto \exp\{\nu_i + \nu_j + \zeta \mathbf{z}_i' \mathbf{z}_j\}. \tag{13}$$

As there is no connection between the groups, the networks are simulated and estimated independently. We first present how we simulate the data following the model (10).

F.1 ARD Simulation

The parameters are defined as follows: $\zeta = 1.5, \quad \nu_i \sim \mathcal{N}(-1.25, 0.37)$, and the $\mathbf{z}_i$ are distributed uniformly according to a von Mises–Fisher distribution. We use a hypersphere of dimension 3. We set the same values for the parameter for the 20 groups. We generate the probabilities of links in each network following Breza et al. (2020).

$$P(a_{ij} = 1 | \nu_i, \nu_j, \zeta, \mathbf{z}_i, \mathbf{z}_j) = \frac{\exp\{\nu_i + \nu_j + \zeta \mathbf{z}_i' \mathbf{z}_j\} \sum_{i=1}^N d_i}{\sum_{ij} \exp\{\nu_i + \nu_j + \zeta \mathbf{z}_i' \mathbf{z}_j\}}, \tag{14}$$

where d_i is the degree defined by $d_i \approx \dfrac{C_p(0)}{C_p(\zeta)} \exp{(\nu_i)} \displaystyle\sum_{i=1}^N \exp(\nu_i)$, and the function $C_p(.)$ is the normalization constant in the von Mises–Fisher distribution density function. After computing the probability of a link for any pair in the population, we sample the entries of the adjacency matrix using a Bernoulli distribution with probability (14).

To generate the ARD, we require the "traits" (e.g. cities) for each individual. We set $K = 12$ traits on the hypersphere. Their location $\mathbf{v}_k$ is distributed uniformly according to the von Mises–Fisher distribution. The individuals having the trait k are assumed to be generated by a von Mises–Fisher distribution with the location parameter $\mathbf{v}_k$ and the intensity parameter $\eta_k \sim |\mathcal{N}(4,1)|$, $k = 1, \ldots, 12$.

We attribute traits to individuals given their spherical coordinates. We first define N_k, the number of individuals having the trait k:

$$N_k = \left\lfloor r_k \frac{\sum_{i=1}^{N} f_{\mathcal{M}}(\mathbf{z}_i|\mathbf{v}_k, \eta_k)}{\max_i f_{\mathcal{M}}(\mathbf{z}_i|\mathbf{v}_k, \eta_k)} \right\rfloor,$$

where $\lfloor x \rfloor$ represents the greatest integer less than or equal to x, r_k is a random number uniformly distributed over $(0.8; 0.95)$, and $f_{\mathcal{M}}(\mathbf{z}_i|\mathbf{v}_k, \eta_k)$ is the von Mises–Fisher distribution density function evaluated at $\mathbf{z}_i$ with the location parameter $\mathbf{v}_k$ and the intensity parameter η_k.

The intuition behind this definition for N_k is that when many $\mathbf{z}_i$ are close to $\mathbf{v}_k$, many individuals should have the trait k.

We can finally attribute trait k to individual i by sampling a Bernoulli distribution with the probability f_{ik} given by

$$f_{ik} = N_k \frac{f_{\mathcal{M}}(\mathbf{z}_i|\mathbf{v}_k, \eta_k)}{\sum_{i=1}^{N} f_{\mathcal{M}}(\mathbf{z}_i|\mathbf{v}_k, \eta_k)}.$$

The probability of having a trait depends on the proximity of the individuals to the trait's location on the hypersphere.

F.2 Model Estimation

In practice, we only have the ARD and the traits for each individual. McCormick and Zheng (2015) propose a MCMC approach to infer the parameters in the model (13).

However, the spherical coordinates and the degrees in this model are not identified. The authors solve this issue by fixing some $\mathbf{v}_k$ and use the fixed positions to rotate the latent surface back to a common orientation at each iteration of the MCMC using a Procrustes transformation. In addition, the total size of a subset b_k is constrained in the MCMC.

As discussed by McCormick and Zheng (2015), the number of $\mathbf{v}_k$ and b_k to be set as fixed depends on the dimensions of the hypersphere. In our simulations, $\mathbf{v}_1$, $\mathbf{v}_2$, $\dots$, $\mathbf{v}_5$ are set as fixed to rotate back the latent space. When simulating the data, we let $\mathbf{v}_1 = (1, 0, 0)$, $\mathbf{v}_2 = (0, 1, 0)$, and $\mathbf{v}_3 = (0, 0, 1)$. This ensures that the fixed positions on the hypersphere are spaced, as suggested by the authors, to use as much of the space as possible, maximizing the distance between the estimated positions. We also constrain b_3 to its true value. The results do not change when we constrain a larger set of b_k

Following Breza et al. (2020), we estimate the link probabilities using the parameters' posterior distributions. The gregariousness parameters are computed from the degrees d_i and the parameter ζ using the following equation:

$$\nu_i = \log(d_i) - \log\left(\sum_{i=1}^{N} d_i\right) + \frac{1}{2}\log\left(\frac{C_p(\zeta)}{C_p(0)}\right).$$

G Bayesian Inference

G.1 Posterior Distributions for Algorithm 1.

To compute the posterior distributions, we set prior distributions on $\tilde{\alpha}$, $\mathbf{\Lambda}$, and σ^2, where $\tilde{\alpha} = \log(\frac{\alpha}{1-\alpha})$ and $\mathbf{\Lambda} = [\boldsymbol{\beta}, \boldsymbol{\gamma}]$. In Algorithm 1, we therefore sample $\tilde{\alpha}$ and compute α, such that $\alpha = \frac{\exp(\tilde{\alpha})}{1+\exp(\tilde{\alpha})}$. Using this functional form for computing α ensures that $\alpha \in (0,1)$. The prior distributions are set as follows:

$$
\begin{aligned}
\tilde{\alpha} &\sim \mathcal{N}(\mu_{\tilde{\alpha}}, \sigma^2_{\tilde{\alpha}}),\\
\mathbf{\Lambda}|\sigma^2 &\sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{\Lambda}}, \sigma^2 \mathbf{\Sigma}_{\mathbf{\Lambda}}),\\
\sigma^2 &\sim IG(\frac{a}{2}, \frac{b}{2}).
\end{aligned}
$$

For the simulations and estimations in this paper, we set $\mu_{\tilde{\alpha}} = -1$, $\sigma^{-2}_{\tilde{\alpha}} = 2$, $\boldsymbol{\mu}_{\mathbf{\Lambda}} = \mathbf{0}$, ${\mathbf{\Sigma}_{\mathbf{\Lambda}}}^{-1} = \frac{1}{100}\boldsymbol{I}_K$, $a = 4$, and $b = 4$, where $\boldsymbol{I}_K$ is the identity matrix of dimension K and $K = \dim(\mathbf{\Lambda})$.

Following Algorithm 1, α is updated at each iteration t of the MCMC by drawing $\tilde{\alpha}^*$ from the proposal $\mathcal{N}(\tilde{\alpha}_{t-1}, \xi_t)$, where the jumping scale ξ_t is also updated at each t following Atchadé and Rosenthal (2005) for an acceptance rate of a^* targeted at 0.44. As the proposal is symmetrical, $\alpha^* = \frac{\exp(\tilde{\alpha}^*)}{1+\exp(\tilde{\alpha}^*)}$ is accepted with the probability

$$
\min\left\{1, \frac{\mathcal{P}(\mathbf{y}|\mathbf{A}_t, \mathbf{\Lambda}_{t-1}, \alpha^*)P(\tilde{\alpha}^*)}{\mathcal{P}(\mathbf{y}|\mathbf{A}_t, \boldsymbol{\theta}_{t-1})P(\tilde{\alpha}_t)}\right\}.
$$

The parameters $\mathbf{\Lambda}_t = [\boldsymbol{\beta}_t,\ \boldsymbol{\gamma}_t]$ and σ^2_t are drawn from their posterior conditional distributions, given as follows:

$$
\begin{aligned}
\mathbf{\Lambda}_t|\boldsymbol{y}, \boldsymbol{A}_t, \alpha_t, \sigma^2_{t-1} &\sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{\mathbf{\Lambda}_t}, \sigma^2_{t-1}\hat{\mathbf{\Sigma}}_{\mathbf{\Lambda}_t}),\\
\sigma^2_t|\boldsymbol{y}, \boldsymbol{A}_t, \boldsymbol{\theta}_t &\sim IG\left(\frac{\hat{a}_t}{2}, \frac{\hat{b}_t}{2}\right),
\end{aligned}
$$

where,

$$
\begin{aligned}
\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\Lambda}_t}^{-1} &= \mathbf{V}_t'\mathbf{V}_t + \boldsymbol{\Sigma}_{\boldsymbol{\Lambda}}^{-1}, \\
\hat{\boldsymbol{\mu}}_{\boldsymbol{\Lambda}_t} &= \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\Lambda}_t}\left(\mathbf{V}_t'(\boldsymbol{y} - \alpha_t \boldsymbol{G}_t \boldsymbol{y}) + \boldsymbol{\Sigma}_{\boldsymbol{\Lambda}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\Lambda}}\right), \\
\hat{a}_t &= a + N, \\
\hat{b}_t &= b + (\boldsymbol{\Lambda}_t - \boldsymbol{\mu}_{\boldsymbol{\Lambda}})'\boldsymbol{\Sigma}_{\boldsymbol{\Lambda}}^{-1}(\boldsymbol{\Lambda}_t - \boldsymbol{\mu}_{\boldsymbol{\Lambda}}) + (\boldsymbol{y} - \alpha_t \boldsymbol{G}_t \boldsymbol{y} - \mathbf{V}_t\boldsymbol{\Lambda}_t)'(\boldsymbol{y} - \alpha_t \boldsymbol{G}_t \boldsymbol{y} - \mathbf{V}_t\boldsymbol{\Lambda}_t), \\
\mathbf{V}_t &= [\mathbf{1},\ \boldsymbol{X},\ \boldsymbol{G}_t\boldsymbol{X}].
\end{aligned}
$$

G.2 Network Sampling

This section explains how we sample the network in Algorithm 1 using Gibbs sampling. As discussed above, a natural solution is to update only one entry of the adjacency matrix at every step t of the MCMC. The entry (i, j) is updated according to its conditional posterior distribution. For each entry, however, we need to compute $\mathcal{P}(\mathbf{y}|0, \mathbf{A}_{-ij})$ and $\mathcal{P}(\mathbf{y}|1, \mathbf{A}_{-ij})$, which are the respective likelihoods of replacing a_{ij} by 0 or by 1. The likelihood computation requires the determinant of $(\mathbf{I} - \alpha\mathbf{G})$, which has a complexity $O(N^3)$ where N is the dimension of $\mathbf{G}$. This implies that we must compute $2N(N-1)$ times $\det(\mathbf{I} - \alpha\mathbf{G})$ to update the adjacency matrix at each step of the MCMC. As $\boldsymbol{G}$ is row-normalized, alternating any off-diagonal entry (i, j) in $\boldsymbol{A}$ between 0 and 1 perturbs all off-diagonal entries of the row i in $(\mathbf{I} - \alpha\mathbf{G})$. We show that $\boldsymbol{A}_{ij}$ and $\det(\mathbf{I} - \alpha\mathbf{G})$ can be updated by computing a determinant of an auxiliary matrix that requires only updating two entries.

Assume that we want to update the entry (i, j). Let h be the function defined in $\mathbb{N}$ such that $\forall\ x \in \mathbb{N}^*$, $h(x) = x$, and $h(0) = 1$. Let $\mathbf{L}$ be an N × N diagonal matrix, where $\mathbf{L}_{ii} = h(n_i)$, and n_i stands for the degree of i, while $\mathbf{L}_{kk} = 1$ for all $k \neq i$, and $\mathbf{W}$ is the matrix $\boldsymbol{G}$ where the row i of $\mathbf{W}$ is replaced by the row i of $\boldsymbol{A}$. Then, as the determinant is linear in each row, we can obtain $\mathbf{I} - \alpha\mathbf{G}$ by dividing the row i of

$\mathbf{L} - \alpha\mathbf{W}$ by $h(n_i)$. We get:

$$\det(\mathbf{I} - \alpha\mathbf{G}) = \frac{1}{h(n_i)}\det(\mathbf{L} - \alpha\mathbf{W}).$$

When a_{ij} changes (from 0 to 1, or 1 to 0), note that only the entries (i, i) and (i, j) change in $\mathbf{L} - \alpha\mathbf{W}$. Two cases can be distinguished.

- If $a_{ij} = 0$ before the update, then the new degree of i will be $n_i + 1$. Thus, the entry (i, i) in $\mathbf{L} - \alpha\mathbf{W}$ will change from $h(n_i)$ to $h(n_i + 1)$ (as the diagonal of $\mathbf{W}$ equals 0), and the entry (i, j) will change from 0 to $-\alpha$. The new determinant is therefore given by

  $$\det(\mathbf{I} - \alpha\mathbf{G}^*) = \frac{1}{h(n_i + 1)}\det(\mathbf{L}^* - \alpha\mathbf{W}^*),$$

  where $\mathbf{G}^*$, $\mathbf{L}^*$, and $\alpha\mathbf{W}^*$ are the new matrices once a_{ij} has been updated.

- If $a_{ij} = 1$ before the update, then the new degree of k will be $n_i - 1$. Thus, the entry (i, i) in $\mathbf{L} - \alpha\mathbf{W}$ will change from $h(n_i)$ to $h(n_i - 1)$, and the entry (i, j) will change from $-\alpha$ to 0. The new determinant is therefore given by

  $$\det(\mathbf{I} - \alpha\mathbf{G}^*) = \frac{1}{h(n_i - 1)}\det(\mathbf{L}^* - \alpha\mathbf{W}^*).$$

Then, to update $\det(\mathbf{L} - \alpha\mathbf{W})$ when only the entries (i, i) and (i, j) change, we adapt the Lemma 1 in Hsieh et al. (2019) as follows:

Proposition 3. *Let $\mathbf{e}_i$ be the i'th unit basis vector in $\mathbb{R}^N$. Let $\mathbf{M}$ denote an $N \times N$ matrix and $\mathbf{B}_{ij}(\mathbf{Q}, \epsilon)$ an $N \times N$ matrix as a function of an $N \times N$ matrix $\mathbf{Q}$ and a real value ϵ, such that*

$$\mathbf{B}_{ij}(\mathbf{Q}, \epsilon) = \frac{\mathbf{Q}\mathbf{e}_i\mathbf{e}_j'\mathbf{Q}}{1 + \epsilon\mathbf{e}_j'\mathbf{Q}\mathbf{e}_i}. \tag{15}$$

Adding a perturbation ϵ_1 in the (i, i)th position and a perturbation ϵ_2 in the (i, j)th position to the matrix $\mathbf{M}$ can be written as $\tilde{\mathbf{M}} = \mathbf{M} + \epsilon_1\mathbf{e}_i\mathbf{e}_i' + \epsilon_2\mathbf{e}_i\mathbf{e}_j'$.

1. *The inverse of the perturbed matrix can be written as*

$$\tilde{\mathbf{M}}^{-1} = \mathbf{M}^{-1} - \epsilon_1 \mathbf{B}_{ii}(\mathbf{M}^{-1}, \epsilon_1) - \epsilon_2 \mathbf{B}_{ij}\left(\mathbf{M}^{-1} - \epsilon_1 \mathbf{B}_{ii}(\mathbf{M}^{-1}, \epsilon_1), \epsilon_2\right).$$

2. *The determinant of the perturbed matrix can be written as*

$$det\left(\tilde{\mathbf{M}}\right) = \left(1 + \epsilon_2 \mathbf{e}_j' \left(\mathbf{M}^{-1} - \epsilon_1 \mathbf{B}_{ii}(\mathbf{M}^{-1}, \epsilon_1)\mathbf{e}_i\right)\right)(1 + \epsilon_1 \mathbf{e}_i' \mathbf{M}^{-1} \mathbf{e}_i)\, det\,(\mathbf{M}).$$

Proof. 1. By the Sherman–Morrison formula (Mele, 2017), we have

$$\left(\mathbf{M} + \epsilon \mathbf{e}_i \mathbf{e}_j'\right)^{-1} = \mathbf{M}^{-1} - \epsilon \frac{\mathbf{M}^{-1}\mathbf{e}_i \mathbf{e}_j' \mathbf{M}^{-1}}{1 + \epsilon \mathbf{e}_j' \mathbf{M}^{-1} \mathbf{e}_i} = \mathbf{M}^{-1} - \epsilon \mathbf{B}_{ij}(\mathbf{M}, \epsilon).$$

Thus,

$$\begin{aligned}
\tilde{\mathbf{M}}^{-1} &= \left((\mathbf{M} + \epsilon_1 \mathbf{e}_i \mathbf{e}_i') + \epsilon_2 \mathbf{e}_i \mathbf{e}_j'\right)^{-1}, \\
\tilde{\mathbf{M}}^{-1} &= (\mathbf{M} + \epsilon_1 \mathbf{e}_i \mathbf{e}_i')^{-1} - \epsilon_2 \mathbf{B}_{ij}((\mathbf{M} + \epsilon_1 \mathbf{e}_i \mathbf{e}_i')^{-1}, \epsilon_2), \\
\tilde{\mathbf{M}}^{-1} &= \mathbf{M}^{-1} - \epsilon_1 \mathbf{B}_{ii}(\mathbf{M}^{-1}, \epsilon_1) - \epsilon_2 \mathbf{B}_{ij}\left(\mathbf{M}^{-1} - \epsilon_1 \mathbf{B}_{ii}(\mathbf{M}^{-1}, \epsilon_1), \epsilon_2\right).
\end{aligned}$$

2. By the matrix determinant lemma (Johnson and Horn, 1985), we have

$$\det\left(\mathbf{M} + \epsilon \mathbf{e}_i \mathbf{e}_j'\right) = (1 + \epsilon \mathbf{e}_j' \mathbf{M}^{-1} \mathbf{e}_i) \det(\mathbf{M}).$$

It follows that

$$\begin{aligned}
\det\left(\tilde{\mathbf{M}}\right) &= \det\left((\mathbf{M} + \epsilon_1 \mathbf{e}_i \mathbf{e}_i') + \epsilon_2 \mathbf{e}_i \mathbf{e}_j'\right), \\
\det\left(\tilde{\mathbf{M}}\right) &= (1 + \epsilon_2 \mathbf{e}_j' (\mathbf{M} + \epsilon_1 \mathbf{e}_i \mathbf{e}_i')^{-1} \mathbf{e}_i) \det(\mathbf{M} + \epsilon_1 \mathbf{e}_i \mathbf{e}_i'), \\
\det\left(\tilde{\mathbf{M}}\right) &= \left(1 + \epsilon_2 \mathbf{e}_j' \left(\mathbf{M}^{-1} - \epsilon_1 \mathbf{B}_{ii}(\mathbf{M}^{-1}, \epsilon_1)\mathbf{e}_i\right)\right)(1 + \epsilon_1 \mathbf{e}_i' \mathbf{M}^{-1} \mathbf{e}_i) \det(\mathbf{M}).
\end{aligned}$$

□

The method proposed above becomes computationally intensive when many entries must be updated simultaneously. We also propose an alternative method that allows updating the block for entries in $\boldsymbol{A}$. Let $\mathbf{D} = (\mathbf{I} - \alpha\mathbf{G})$; we can write

$$\det(\mathbf{D}) = \sum_{j=1}^{N}(-1)^{i+j}\mathbf{D}_{\mathrm{ij}}\delta_{\mathrm{ij}}, \tag{16}$$

where i denotes any row of $\mathbf{D}$ and δ_{ij} is the minor[38] associated with the entry (i, j). The minors of row i do not depend on the values of entries in row i. To update any block in row i, we therefore compute the N minors associated with i and use this minor within the row. We can then update many entries simultaneously without increasing the number of times that we compute $\det(\mathbf{D})$.

One possibility is to update multiple links simultaneously by randomly choosing the number of entries to consider and their position in the row. As suggested by Chib and Ramamurthy (2010), this method would help the Gibbs to converge more quickly. We can summarize how we update the row i as follows:

1. Compute the N minors $\delta_{i1}, \ldots, \delta_{in}$.

2. Let $\Omega_{\boldsymbol{G}}$ be the entries to update in the row i, and $n_{\boldsymbol{G}} = |\Omega_{\boldsymbol{G}}|$ the number of entries in $\Omega_{\boldsymbol{G}}$.

    (a) Choose r, the size of the block to update, as a random integer number such that $1 \leq r \leq n_{\boldsymbol{G}}$. In practice, we choose $r \leq \min(5, n_{\boldsymbol{G}})$ because the number of possibilities of links to consider grows exponentially with r.

    (b) Choose the r random entries from $\Omega_{\boldsymbol{G}}$. These entries define the block to update.

    (c) Compute the posterior probabilities of all possibilities of links inside the block and update the block (there are 2^r possibilities). Use the minors calculated at 1 and the formula (16) to quickly compute $det(\mathbf{D})$.

[38] The determinant of the submatrix of $\mathbf{M}$ by removing row i and column j.

(d) Remove the r drawn positions from $\Omega_{\boldsymbol{G}}$ and let $n_{\boldsymbol{G}} = n_{\boldsymbol{G}} - r$. Replicate 2a, 2b, and 2c until $n_{\boldsymbol{G}} = 0$.