Graph sparsity/ Low dimensional embeddings (d)	GridCV on embedding dimensions	Information theory measure (penalize number of parameters = f(d)	Sparse matrix factorization	Non-negative matrix factorization	Normalized matrix factorization
	Select best dimension that improves predictive performance on independent test set	Select dimension that compromises loglikelihood and model complexity	Encourages unused components to move towards zero Set dimension of embeddings to maximum Lasso shrinks the unused dimensions => effective dimension if smaller than max	Rejects negative values of embedding elements, ensures sparsity	Uses projected gradients to ensure the L2 norm=1 for all embeddings
Boosting	(-) Computationally costly (many dimensions are tested * #folds * number of bootstrap samples) (-) The embeddings are undetermined by rotation => unidentifiability Computing classical confidence intervals is not possible (+) However, we can obtain a	(-) Computationally costly (range of dimensions * bootstrap) Could be done in two-times: estimate dimension using information criteria Use Bootstrap to get uncertainty over associations (complexity of one fit * (range of dimensions + bootstrap)) (+) Sparse	1)Lasso on embeddings (+) Allows the selection of the dimension (lower-dimensional embeddings) => (+) could lead to sparse associations or => (-) could simply cancel some unused dimensions but still yield dense associations (No theoretical	(+) Induce sparsity (-) No possible negative associations (-) Still requires the selection of the embedding dimension	(+) Normalized association matrix (Only in the symmetric case) Adds an additional constraint on the embeddings => plays against the sparsity of the embeddings

Graph sparsity/ Low dimensional embeddings (d)	**GridCV on embedding dimensions**	**Information theory measure (penalize number of parameters = f(d))**	**Sparse matrix factorization**	**Non-negative matrix factorization**	**Normalized matrix factorization**
	confidence interval for the association matrix => (+) Sparse associations (the product of the embeddings which does not suffer from the same issue) We cannot aggregate the embeddings or quantify the uncertainty over their values (but any statistic that is not sensitive to rotation of their dimension) can be aggregated across bootstrap samples.	associations (CINF filtering) + small dimensions (Information criteria) (-) AIC has been shown to favor complex models (-) BIC has been shown to favor too simple models with few data	guideline for using this with bootstrap) 2) Lasso on associations => Confidence intervals should be computed using selective inference from Tibshirani et al 2016 to compute confidence intervals, classical formula of Mean+-1.96SE is not correct (lasso reduces the variance of the estimators of the parameters) => (+) Associations are sparse But, (-) Embeddings could be high dimensional (with negative + positive components) to achieve		

Graph sparsity/ Low dimensional embeddings (d)	GridCV on embedding dimensions	Information theory measure (penalize number of parameters = f(d)	Sparse matrix factorization	Non-negative matrix factorization	Normalized matrix factorization
			sparsity (zeros) in their dot product = association matrix		
Generate N subsamples with replacement of the same size than the number of observations. Fit the model on each subsample. Compute confidence intervals on estimated associations. Drop edges with CI containing 0. NB: This procedure allows to have a quantification of uncertainty (confidence intervals) on inferred edges					

Graph sparsity/ Low dimensional embeddings (d)	**GridCV on embedding dimensions**	**Information theory measure (penalize number of parameters = f(d)**	**Sparse matrix factorization**	**Non-negative matrix factorization**	**Normalized matrix factorization**
CV based selection of lambda	To select both d and lambda (Computational cost)	/	/	/	/
Vanilla graphical Lasso / GLMNET approach	CV overfits in regression problems	(-) AIC + BIC perform poorly with glasso		/	/
Create a lasso path = a list of decreasing values of a penalty parameter λ For each value of lambda, fit the estimator on all the dataset For the highest value of lambda the graph has zero edges (density=0) As we decrease lambda, more edges are added to the graph Lambda is selected					

Graph sparsity/ Low dimensional embeddings (d)	GridCV on embedding dimensions	Information theory measure (penalize number of parameters = f(d)	Sparse matrix factorization	Non-negative matrix factorization	Normalized matrix factorization
based on various criteria: * Predictive log likelihood, R^2 * AIC, BIC, (number of parameters is computed using number of non zero elements of the precision matrix) NB: This approach infers a graph contained in the true graph (minimizes false positives)					
StARS = Lasso + stability				/	/
Generate N subsamples without replacement of size $10\sqrt(n)$ Fit the model on each subsample for each value of the lambda					

Graph sparsity/ Low dimensional embeddings (d)	GridCV on embedding dimensions	Information theory measure (penalize number of parameters = f(d)	Sparse matrix factorization	Non-negative matrix factorization	Normalized matrix factorization
path For each value of lambda along the lasso path: - Obtains N graphs per value of lambda - Define for each pair of nodes a Bernoulli indicator of whether an edge is inferred (needs a threshold or an epsilon value to select edges here) - Compute using the population of size N the variance of this indicator - Average this variance across all pairs (edges) => obtain a measure of instability With highest					

Graph sparsity/ Low dimensional embeddings (d)	GridCV on embedding dimensions	Information theory measure (penalize number of parameters = f(d)	Sparse matrix factorization	Non-negative matrix factorization	Normalized matrix factorization
regularization, all graphs are empty (Highest sparsity, high stability = low instability) As we decrease lambda, edges are added but they are not necessarily the same => increase instability At some point, the graphs get dense => decrease instability Networks on subsamples will then tend to converge towards a common structure (up to an accepted level of instability recommended Beta=0.05) This procedure selects					

Graph sparsity/ Low dimensional embeddings (d)	GridCV on embedding dimensions	Information theory measure (penalize number of parameters = f(d)	Sparse matrix factorization	Non-negative matrix factorization	Normalized matrix factorization
lambda only. Then, the model is trained with all the data with the selected penalty to obtain the final graph. NB: StARS infers a graph that contains the true graph (minimizes false positives)					