Supporting Information for "Tensor Envelope Mixture Model For Simultaneous Clustering and Multiway Dimension Reduction" by Kai Deng and Xin Zhang

A Estimation algorithm and envelope dimension selection

A.1 EM-algorithm for TEMM

Algorithm 1 EM algorithm for TEMM (6).

1. Initialize $\{\widehat{\pi}_k^{(0)}, \widehat{\boldsymbol{\mu}}_k^{(0)}\}_{k=1}^K$, $\widehat{\boldsymbol{\Sigma}}^{(0)} = \widehat{\boldsymbol{\Sigma}}_M^{(0)} \otimes \cdots \otimes \widehat{\boldsymbol{\Sigma}}_1^{(0)}$.

2. For $s = 1, 2, \ldots$, repeat the following steps

   (a) E-step: Estimate $\eta_{ik}^{(s)}$ through (11) for all k and i.

   (b) M-step:

      i. For $k = 1, \ldots, K$, update $\widehat{\pi}_k^{(s)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(s)}/n$ and $\widetilde{\boldsymbol{\mu}}_k^{(s)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(s)} \mathbf{X}_i / \sum_{i=1}^n \widehat{\eta}_{ik}^{(s)}$.

      ii. Estimate $\widehat{\boldsymbol{\Gamma}}_m = \operatorname{argmin}_{\boldsymbol{\Gamma}_m} G_m^{(s)}(\boldsymbol{\Gamma}_m)$ for all m under $\boldsymbol{\Gamma}_m^T \boldsymbol{\Gamma}_m = \mathbf{I}_{u_m}$.

      iii. For $k = 1, \ldots, K$, $m = 1, \ldots, M$, update $\{\widehat{\boldsymbol{\alpha}}_k^{(s)}\}_{k=1}^K$, $\{\widehat{\boldsymbol{\Omega}}_m^{(s)}, \widehat{\boldsymbol{\Omega}}_{0m}^{(s)}\}_{m=1}^M$ as

$$\widehat{\boldsymbol{\alpha}}_k^{(s)} = [\![\widetilde{\boldsymbol{\mu}}_k^{(s)}; \widehat{\boldsymbol{\Gamma}}_1^T, \ldots, \widehat{\boldsymbol{\Gamma}}_M^T]\!], \quad \widehat{\boldsymbol{\Omega}}_m^{(s)} = \widehat{\boldsymbol{\Gamma}}_m^T \mathbf{M}_m^{(s)} \widehat{\boldsymbol{\Gamma}}_m, \quad \widehat{\boldsymbol{\Omega}}_{0m}^{(s)} = \widehat{\boldsymbol{\Gamma}}_{0m}^T \mathbf{N}_m^{(s)} \widehat{\boldsymbol{\Gamma}}_{0m}.$$

      iv. Update $\{\widehat{\boldsymbol{\mu}}_k^{(s)}\}_{k=1}^K$, $\widehat{\boldsymbol{\Sigma}}^{(s)}$, $\widehat{\mathbf{B}}^{(s)}$ through parameterization (6).

3. Until $\sum_k \|\widehat{\boldsymbol{\mu}}_k^{(s)} - \widehat{\boldsymbol{\mu}}_k^{(s-1)}\|_F / \sum_k \|\widehat{\boldsymbol{\mu}}_k^{(s-1)}\|_F < 10^{-3}$.

A.2 Envelope dimension selection

Selecting envelope dimensions $\{u_m\}_{m=1}^M$ is of critical importance in practice. For our likelihood-based approach, we can consistently select envelope dimensions using Bayesian information

criterion (BIC). However, the computation cost of the standard BIC can be expensive for $M \geq 2$ if we search over all possible $\prod_m p_m$ combinations of $(u_1, \ldots, u_M)$. We propose an alternative two-step procedure as follows.

First, for $m = 1, \ldots, M$, we select the envelope dimension $\widetilde{u}_m$ by minimizing the following BIC-type criterion over the set $\{0, 1, \ldots, p_m\}$,

$$\mathrm{BIC}_m(u_m) = \log |\boldsymbol{\Gamma}_m^T \widehat{\mathbf{M}}_m \boldsymbol{\Gamma}_m| + \log |\boldsymbol{\Gamma}_m^T \widehat{\mathbf{N}}_m^{-1} \boldsymbol{\Gamma}_m| + C \cdot u_m \log(n)/n. \tag{A1}$$

The above criterion contains the similar form of objective function in (14), which gives a log-pseudo-likelihood argument [see Zhang and Mai, 2018]. This separate selection over each mode m is estimating the envelope dimension of $\mathcal{E}_{\boldsymbol{\Sigma}_m}(\mathbf{B}_{(m)})$ while keeping the dimensions of other $(M-1)$ modes unreduced. Therefore, the dimension selection procedure is less sensitive to the complex interrelationships of each mode of the tensor. The constant C is chosen as 1 as suggested by Zhang and Mai [2018]. We provide further theoretical justification of this separate approach in Web Appendix F and show it can be used as a stand-alone dimension selection method.

Next, we refine the results by grid search over a neighbor of the selection result from the separate selection, denoted as $\widetilde{u} \equiv (\widetilde{u}_1, \ldots, \widetilde{u}_M)$. Specifically, we search over all the combinations of $u \equiv (u_1, \ldots, u_M)$ in the neighborhood of $\widetilde{u}$, $\mathcal{N}(\widetilde{u}) = \{u : \max(1, \widetilde{u}_m - 2) \leq u_m \leq \min(\widetilde{u}_m + 2, p_m),\ m = 1, \ldots, M\}$, that minimizes

$$\mathrm{BIC}(u) = -2 \sum_{i=1}^{n} \log(\sum_{k=1}^{K} \widehat{\pi}_k^u f_k(\mathbf{X}_i; \widehat{\boldsymbol{\theta}}^u)) + \log(n) \cdot K_u. \tag{A2}$$

In the above BIC, $K_u = (K-1)\prod_{m=1}^{M} u_m + \sum_{m=1}^{M} p_m(p_m+1)/2$ is the total number of parameters in TEMM, $\widehat{\pi}_k^u$ and $\widehat{\boldsymbol{\theta}}^u$ are the estimated parameters with envelope dimension fixed at u.

B Additional numerical results

B.1 Clustering analysis under the TEMM models

Parallel to the summarized results in Figures 2 and 3 of the paper, we summarize the averages (with standard errors) of the clustering error in Web Table 1, and the estimation error of cluster centroids in Web Table 2. The model setting and interpretation of results are exactly the same as in the paper. The Tables provide more detailed comparisons among the methods. In particular, TEMM achieved the smallest clustering error and estimation error among all competing methods, and is significantly better than all existing methods. There is no significant difference between the TEMM-BIC and TEMM (i.e. with true envelope dimension) for Models **M1–M3**. For **M4**, the average clustering errors achieve by TEMM-BIC are slightly lower than TEMM. However, as Figure 3 illustrates, the median clustering errors achieved by TEMM-BIC are slightly higher than that achieved by TEMM,

and there are fewer outliers from TEMM-BIC. This indicates TEMM-BIC could be more stable than TEMM.

Web Table 1: Average and standard error (in parenthesis) of clustering error R (%) over 100 data replications. The missing result of **M3** is because the source code for DTC can not be implemented in 4-way tensor. Because the implementation of MCFA is very time consuming, we only include 10 replications results for **M3–M4**.

Method	Bayes	**TEMM-BIC**	**TEMM**	TGMM	HD-GMM	KM(++)	PKM	SKM	AFPF	DTC	MCFA
M1	10.75 (–)	11.67 (0.25)	11.61 (0.25)	19.26 (0.42)	37.31 (0.75)	15.04 (0.45)	15.09 (0.47)	13.50 (0.36)	13.41 (0.32)	42.79 (0.56)	42.32 (0.76)
M2	14.47 (–)	15.83 (0.54)	17.65 (0.90)	29.98 (0.73)	40.02 (0.72)	27.39 (0.84)	27.71 (0.83)	33.92 (0.47)	20.72 (0.67)	33.88 (0.82)	47.42 (0.29)
M3	10.20 (–)	11.49 (0.47)	12.73 (0.76)	26.12 (0.89)	32.54 (0.99)	26.13 (0.93)	26.17 (0.94)	32.74 (0.77)	18.39 (0.87)	– (–)	47.70 (0.73)
M4(K=2)	1.11 (–)	4.25 (1.14)	4.45 (1.15)	20.30 (1.08)	24.68 (1.34)	20.42 (1.06)	20.49 (1.11)	30.30 (1.14)	13.25 (1.12)	37.70 (0.82)	46.70 (0.87)
M4(K=3)	5.16 (–)	8.62 (0.96)	9.17 (1.10)	44.04 (0.98)	49.47 (0.93)	43.54 (0.99)	43.74 (0.94)	51.13 (0.76)	41.10 (1.10)	56.15 (0.44)	60.60 (0.94)
M4(K=4)	9.05 (–)	14.50 (0.75)	17.61 (1.51)	54.38 (0.64)	56.47 (0.69)	54.21 (0.65)	53.43 (0.68)	60.92 (0.48)	52.86 (0.73)	63.42 (0.32)	69.10 (0.50)

Web Table 2: Average and standard error (in parenthesis) of estimation error in centroids over 100 data replications. The missing result of **M3** is because the source code for DTC can not be implemented in 4-way tensor. Because the implementation of MCFA is very time consuming, we only include 10 replications results for **M3–M4**.

Method	**TEMM-BIC**	**TEMM**	TGMM	HD-GMM	KM(++)	PKM	SKM	AFPF	DTC	MCFA
M1	0.23 (0.00)	0.21 (0.00)	0.29 (0.01)	0.64 (0.01)	0.33 (0.01)	0.33 (0.01)	0.28 (0.01)	0.28 (0.01)	1.94 (0.00)	0.77 (0.02)
M2	0.31 (0.02)	0.37 (0.03)	0.65 (0.03)	1.08 (0.02)	0.90 (0.03)	0.90 (0.03)	1.05 (0.02)	0.60 (0.03)	2.36 (0.01)	1.42 (0.03)
M3	0.44 (0.02)	0.47 (0.03)	1.00 (0.03)	1.16 (0.03)	1.11 (0.03)	1.10 (0.04)	1.18 (0.03)	0.78 (0.04)	– (–)	7.75 (0.08)
M4(K=2)	1.17 (0.04)	1.04 (0.04)	1.77 (0.06)	1.86 (0.06)	1.78 (0.06)	1.78 (0.06)	2.17 (0.06)	1.47 (0.06)	5.07 (0.00)	2.57 (0.06)
M4(K=3)	2.17 (0.04)	1.85 (0.07)	4.29 (0.07)	4.16 (0.06)	4.26 (0.07)	4.28 (0.07)	4.61 (0.06)	3.88 (0.07)	7.91 (0.01)	4.45 (0.11)
M4(K=4)	3.29 (0.05)	3.20 (0.13)	6.82 (0.07)	6.30 (0.06)	6.80 (0.06)	6.73 (0.07)	7.07 (0.06)	5.92 (0.06)	10.57 (0.01)	6.34 (0.11)

B.2 Clustering analysis under model mis-specification

Web Table 3: Average and standard error (in parenthesis) of clustering error R (%) over 100 data replications.

Method	Bayes	**TEMM-BIC**	TGMM	HD-GMM	KM(++)	PKM	SKM	APFP	DTC	MCFA
TGMM1	14.02 (–)	17.08 (0.28)	26.28 (0.44)	32.24 (0.78)	21.80 (0.48)	21.83 (0.49)	32.02 (0.42)	20.64 (0.30)	35.40 (0.72)	47.67 (0.19)
TGMM2	10.64 (–)	12.43 (0.18)	38.50 (0.72)	43.10 (0.55)	38.00 (0.67)	39.82 (0.68)	46.28 (0.27)	37.51 (0.69)	46.92 (0.38)	47.34 (0.21)
STGMM1	17.07 (–)	36.92 (0.94)	33.72 (0.56)	36.50 (0.73)	32.68 (0.48)	32.76 (0.48)	26.55 (0.74)	32.83 (0.48)	34.67 (0.47)	46.16 (0.30)
STGMM2	10.39 (–)	20.00 (0.50)	28.45 (0.57)	36.44 (0.83)	28.70 (0.63)	31.22 (0.66)	23.23 (0.55)	30.43 (1.42)	72.35 (0.29)	76.12 (0.15)
DTC	– (–)	1.04 (0.54)	21.76 (0.98)	19.52 (1.74)	22.38 (0.95)	34.92 (0.97)	39.24 (1.82)	23.48 (1.09)	11.56 (1.86)	52.84 (1.37)
APFP	10.38 (–)	22.36 (0.88)	20.61 (0.88)	27.74 (1.12)	20.01 (0.85)	22.44 (0.92)	23.78 (1.02)	21.30 (0.87)	24.79 (0.91)	61.26 (0.63)

In Web Table 3, we summarize the average and standard error of the clustering errors. The same simulation results are also summarized graphically in Figure 4 of the paper. For these six models with no envelope structures, note that the true envelope dimension implies no reduction in the population models. Therefore, we only have TEMM-BIC results. In other words, the TEMM (with true dimension) is reduced to TGMM. Clearly, TEMM-BIC achieved the smallest error under Models **TGMM1** and **TGMM2**. For sparse setting Model **STGMM1**, TEMM is outperformed by other methods. Model **STGMM2** can be viewed as a special TEMM where envelope basis $\mathbf{\Gamma}_m$ consists of 0's and 1's, and TEMM performed best under this sparse setting. In Model **DTC**, TEMM still achieved smallest clustering error that is close to 0. In Model **APFP**, TEMM is not significantly different from the best method KM/KM++.

B.3 Dimension selection

Web Table 4 summarizes the performance of envelope selection procedure by reporting the correct selection rate in each tensor mode. The left side of Web Table 4 illustrates the correct selection rate of TEMM models **M1–M4**. Under **M2** and **M3**, the proposed BIC can select envelope dimension with high accuracy; under **M1** and **M4**, the correct select rates are relatively low, as it becomes harder to select dimension correctly when the dimension p is large and/or the number of clusters increases. However, as we can see from Web Tables 1 and 2, the mis-selection of envelopes would not hurt the performance of TEMM much, and even may lead to improvement in clustering error.

The right side of Web Table 4 summarizes the BIC tuning of envelope dimensions under the mis-specified models based on 100 data replications, we can see that the BIC tends to select envelope spaces with relative low dimensions. This suggests that dimension reduction is helpful in finite sample, as seen in the improved clustering results from TGMM to TEMM-BIC, even when the true population model has not low-dimensional structure.

Web Table 4: On the left: Correct envelope selection rate in **M1–M4** based on 100 data replications. On the right: Average of BIC tuned envelope dimensions under mis-specified models based on 100 data replications.

	u_1	u_2	u_3	u_4
M1	27%	59%	–	–
M2	99%	94%	93%	–
M3	96%	93%	96%	93%
M4($K=2$)	81%	21%	–	–
M4($K=3$)	76%	19%	–	–
M4($K=4$)	62%	18%	–	–

	u_1	u_2	u_3
TGMM1	1.21	2.15	2.97
TGMM2	1.45	2.10	2.18
STGMM1	2.17	1.93	1.90
STGMM2	6.93	5.89	1.00
DTC	2.35	2.47	2.46
APFP	4.47	2.18	–

B.4 Subspace Estimation

In Web Table 5, we test the performance of TEMM in comparisons to the oracle TEMM estimator with true envelope basis $\mathbf{\Gamma}$'s. Under Model **M1**, we vary the sample size to see the difference between the two methods, and also to see the subspace estimation accuracy by Algorithm 1. When sample size n is reasonably large, e.g. $n = 400$ or 600, the estimated envelope subspace $\mathrm{span}(\mathbf{\Gamma})$ is very accurate. When n is around 200, although the average subspace distance between $\mathrm{span}(\widehat{\mathbf{\Gamma}})$ and $\mathrm{span}(\mathbf{\Gamma})$ is non-ignorable, the error of clustering as well as centroids estimation is considerably close between estimate and true envelope basis. We also remark that if using random basis $\mathbf{\Gamma}_r$ in TEMM, clustering error and subspace distance $d(\mathbf{\Gamma}_r, \mathbf{\Gamma})$ will result in 50% and 1 respectively, both are the highest error one can possibly obtain, centroids estimation error $d(\widehat{\boldsymbol{\mu}}, \boldsymbol{\mu})$ will be substantially greater than $\sum_k \|\overline{\mathbf{X}} - \boldsymbol{\mu}_k\|_F = 0.75$ in all sample size scenarios.

Web Table 5: Subspace estimation accuracy and its effects on clustering and estimation. Under Model **M1**, using estimated $\boldsymbol{\Gamma}$ and true $\boldsymbol{\Gamma}$ in every iteration of Algorithm 1, while model sample size range from 50 to 600. Reported are the average and standard error (in parenthesis) of clustering error R (%), estimation error of cluster centroids $d(\widehat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \sum_k \|\widehat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\|_F$, subspace estimation error $d(\widehat{\boldsymbol{\Gamma}}, \boldsymbol{\Gamma}) = \|\mathbf{P}_{\boldsymbol{\Gamma}} - \mathbf{P}_{\widehat{\boldsymbol{\Gamma}}}\|_F/\sqrt{2u}$ based on 100 data replications.

Criterion	$\widehat{\boldsymbol{\Gamma}}$	$n = 50$	$n = 100$	$n = 200$	$n = 400$	$n = 600$
R	Estimated $\boldsymbol{\Gamma}$	16.54(0.99)	14.28(0.79)	11.54(0.25)	11.17(0.18)	10.67(0.12)
	True $\boldsymbol{\Gamma}$	12.94(0.65)	11.95(0.40)	11.27(0.23)	11.11(0.17)	10.63(0.12)
$d(\widehat{\boldsymbol{\mu}}, \boldsymbol{\mu})$	Estimated $\boldsymbol{\Gamma}$	0.48(0.02)	0.34(0.01)	0.21(0.00)	0.15(0.00)	0.11(0.00)
	True $\boldsymbol{\Gamma}$	0.39(0.01)	0.28(0.01)	0.19(0.00)	0.14(0.00)	0.11(0.00)
$d(\widehat{\boldsymbol{\Gamma}}, \boldsymbol{\Gamma})$	Estimated $\boldsymbol{\Gamma}$	0.69(0.03)	0.55(0.03)	0.24(0.03)	0.12(0.02)	0.04(0.00)

B.5 Four-way tensor example

In Web Table 6, we study the clustering and estimation performance of TEMM based on **M3**, except for bigger variability $\sigma^2 = 2$ and that $p_4 = u_4$ varies from $\{2, 4, 6, 8, 10\}$. Since $\boldsymbol{\Sigma}_4^* = \mathbf{I}$, the fourth-mode of the tensor data is essentially sampling three-order tensors, thus we are essentially increasing the sample size in a sense and gaining more information by increasing $p_4 = u_4$, leads to significant accuracy improvement in both clustering and envelope estimation as demonstrated in Web Table 6.

Web Table 6: Clustering of **M3**, change $\sigma^2 = 2$, and fourth mode dimensions $p_4 = u_4$ vary from 2 to 10. Reported are the average and standard error (in parenthesis) of clustering error R (%) and subspace distance $d(\widehat{\boldsymbol{\Gamma}}, \boldsymbol{\Gamma}) = \|\mathbf{P}_{\boldsymbol{\Gamma}} - \mathbf{P}_{\widehat{\boldsymbol{\Gamma}}}\|_F/\sqrt{2u}$ based on 100 data replications. Since $\boldsymbol{\Sigma}_4^* = \mathbf{I}$, increasing $p_4 = u_4$ is essentially sampling three-order tensors and increasing the sample size in a sense, leads to information gain and significant accuracy improvement in both clustering and envelope estimation.

Criterion	$p_4 = 2$	$p_4 = 4$	$p_4 = 6$	$p_4 = 8$	$p_4 = 10$
Bayes error	32.66	15.69	12.54	5.97	1.84
R	46.76(0.25)	41.58(0.77)	29.98(1.21)	7.64(0.72)	2.00(0.05)
$d(\widehat{\boldsymbol{\Gamma}}, \boldsymbol{\Gamma})$	0.84(0.01)	0.77(0.02)	0.56(0.04)	0.03(0.01)	0.006(0.00)

B.6 Computation time comparison

In Web Table 7, we report the computation time of TEMM and all competing methods under Models **M1** and **TGMM1** based on 100 data replications, as well as the computation time for clustering of Gene Time Course data. The computations are conducted on High Performance Computing Cluster at Florida State University with 16 GB memory, which should be similar to a standard laptop computer. As showed in Web Table 7, TEMM converges in reasonable time. Recall that, in **M1**, $p = (20, 20)$, $u = (2, 2)$, sample size $n = 200$, number of cluster $K = 2$; in **TGMM1**, $p = (10, 10, 10)$, average BIC tuned envelope dimension $u = (1.21, 2.15, 2.97)$, sample size $n = 500$, number of clusters $K = 2$; in the gene time course data, $p = (76, 7)$, BIC selected $u = (7, 1)$, sample size $n = 53$, number of clusters $K = 2$. The three-way tensor with higher total dimension $p = 10^3$ in **TGMM1** is the reason why it takes a longer time than the other two examples.

Web Table 7: Average and standard error (in parenthesis) of computation time (s) under **M1**, **TGMM1** and Gene Time Course data (GTC).

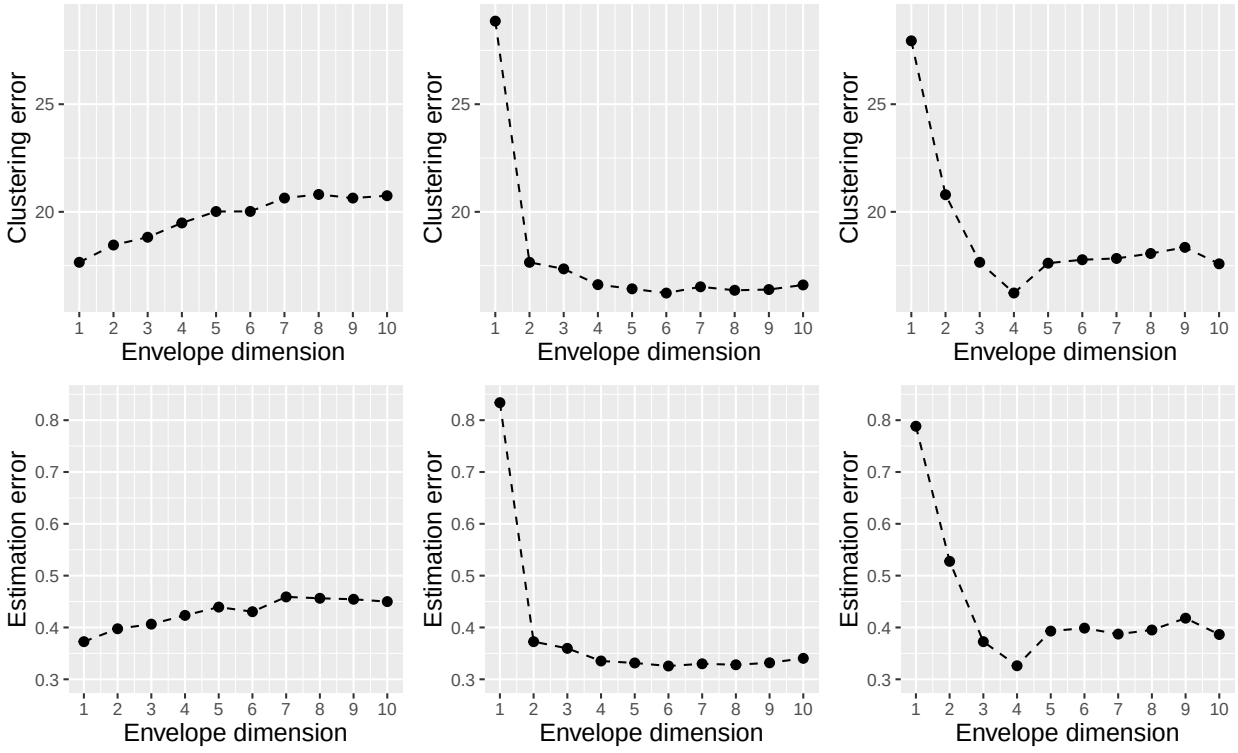
Method	**TEMM-BIC**	TGMM	HD-GMM	KM(++)	PKM	SKM	APFP	DTC	MCFA
M1	4.01 (0.14)	2.45 (0.10)	0.66 (0.02)	0.05 (0.00)	0.14 (0.00)	0.34 (0.01)	106.54 (9.55)	0.65 (0.02)	515.00 (6.24)
TGMM1	35.23 (1.99)	51.79 (1.83)	6.72 (0.36)	1.15 (0.02)	4.00 (0.01)	12.14 (0.29)	4031.14 (227.14)	35.70 (0.28)	6004.04 (21.67)
GTC	7.91	0.45	0.02	0.03	0.01	0.04	12.90	1.23	1689.25

B.7 Mis-specification of envelope dimension and number of clusters

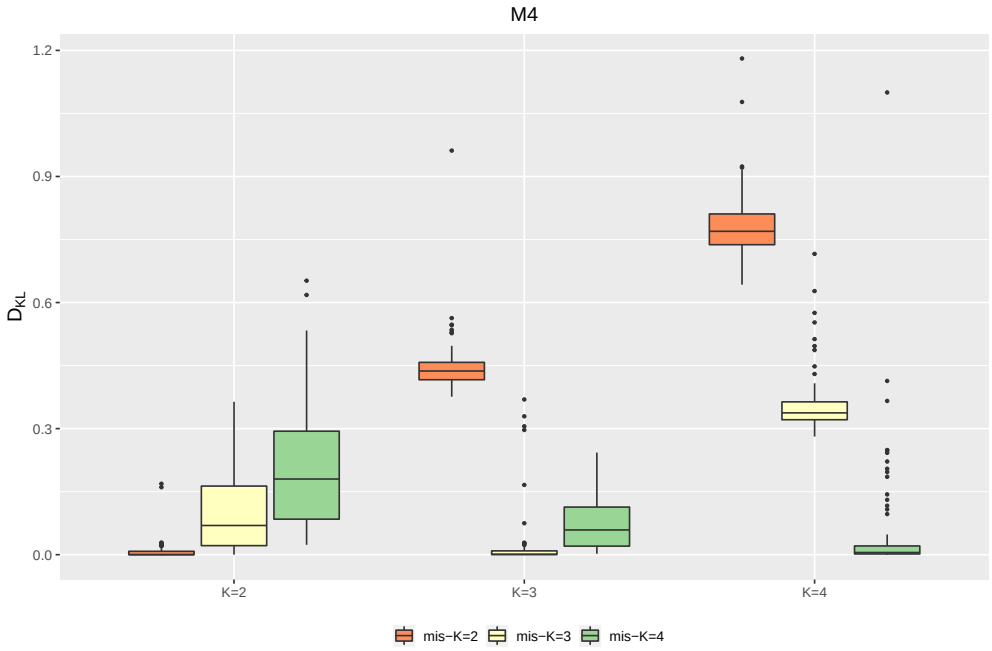
To test the performance of TEMM when envelope dimensions are mis-specified, we specify each the envelope dimension $\widehat{u}_m$, $m = 1, 2, 3$, from 1 to 10 under **M2**, where the true dimension is $u = (1, 2, 3)$. The results are summarized in Web Figure 1. As expected, the clustering error and parameter estimation are much higher if we under-specify the envelope dimensions, while the loss in accuracy and efficiency is less serious if we over-specify the envelope dimensions. For mode-3 of the tensor, if we slightly over-specify the envelope dimension as $\widehat{u}_3 = 4 > 3 = u_3$, there is a noticeable improvement.

We further test the performance of TEMM under mis-specified number of clusters K. Under **M4**, the true number of clusters were set as $K \in \{2, 3, 4\}$. For each true K, we fit TEMM with mis-$K = 2, 3, 4$ to see the effect of mis-specification. Within each data replication, we create a testing set of size 1000 with true latent labels $Y_i \in \{1, \ldots, K\}$. The predicted labels $\widehat{Y}_i$ are then estimated by TEMM parameters fitted from a training set of size $n = 200$. We then evaluate the performance of TEMM under mis-specified number of clusters by calculating the Kullback-Leibler

divergence between the estimated distributions of $\widehat{Y}$ and Y. The results are summarized in Web Figure 2. When mis-K equals to K, the KL divergence would be close to 0, which indicates TEMM accurately predicts the labels. When the number of clusters are under-selected or over-selected, the KL divergence would be bigger. In particular, similar to the envelope dimension mis-specification, we see a more severe loss in prediction accuracy when under-specify the number of clusters and a less serious issue if slightly over-specify the number of clusters.



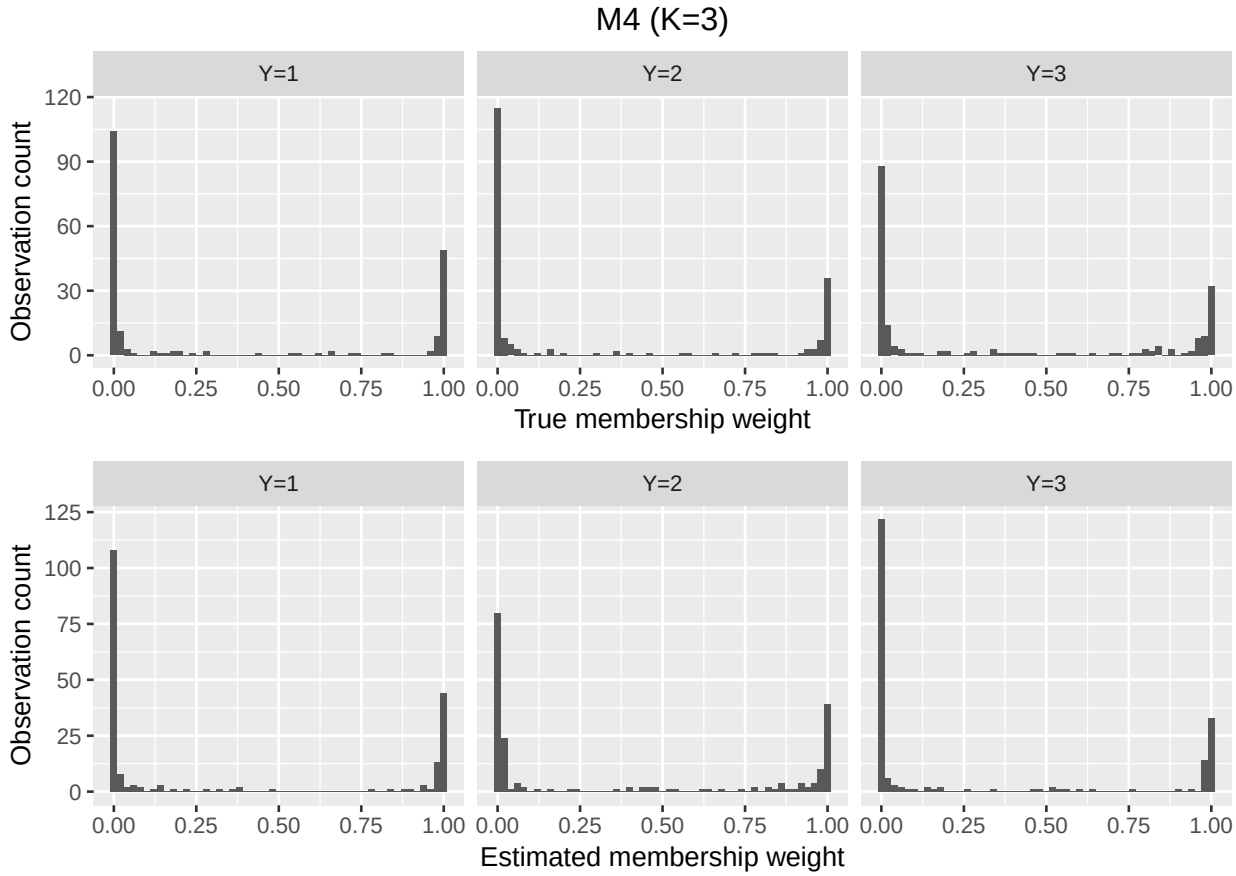
Web Figure 1: Clustering error and parameter estimation error under **M2**. In the left two plots, we vary $\widehat{u}_1$ from 1 to 10 while using true u_2 and u_3. Similarly, only $\widehat{u}_2$ varied in the middle and only $\widehat{u}_3$ varied in the right panels.



Web Figure 2: KL divergence under under **M4** when the number of clusters is mis-specified.

B.8 Cluster assignment probabilities

As suggested by one referee, we exported the membership weight η and $\widehat{\eta}$. Recall that, the membership weight η_{ik} is the probability for an observation $\mathbf{X}_i$ to be in cluster k and is defined as $\eta_{ik} = \Pr(Y_i = k \mid \mathbf{X}_i, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the true parameters in mixture model. The estimated membership weight $\widehat{\eta}_{ik} = \Pr(Y_i = k \mid \mathbf{X}_i, \widehat{\boldsymbol{\theta}})$, with $\widehat{\boldsymbol{\theta}}$ estimated by TEMM. Web Figure 3 shows the histograms of η_{ik} and $\widehat{\eta}_{ik}, k = 1, 2, 3, i = 1, \ldots, 200$ under **M4** with $K = 3$, we can clearly see that TEMM assigns most simulated observations to one cluster with probabilities close to 0 or 1. The distribution of $\widehat{\eta}_{ik}$ is highly close to η_{ik}. In high-dimensional model-based clustering, it is common that the estimated weights η_{ik} will be close to either 0 or 1. This overconfident estimation in our simulation examples are partially because the optimal clustering error is small enough (see Web Table 1), so that when the parameter estimations of TEMM are accurate, the probability of assigning observations into one of the clusters would usually dominate the probabilities of assigning into the other two clusters.



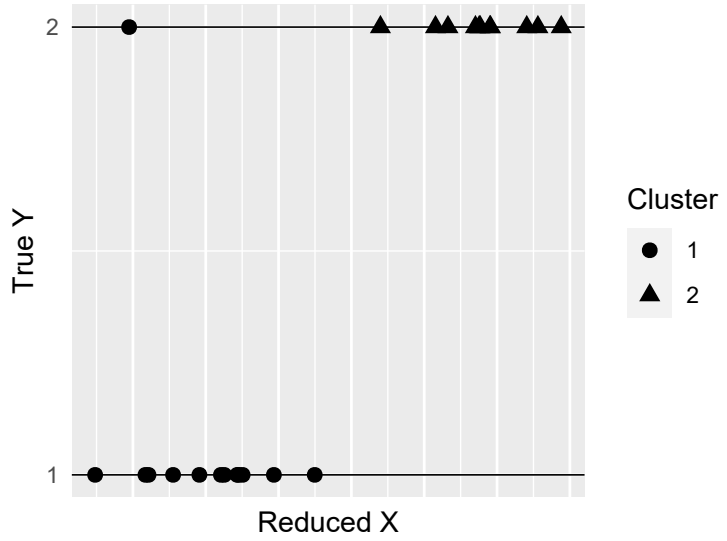
Web Figure 3: Distribution of membership weight under **M4** ($K = 3$) for each cluster. The histograms are the true membership weights η_{ik} (top row) and the estimated $\widehat{\eta}_{ik}$ (bottom row).

C Additional real data analysis

C.1 Finger-Tapping fMRI

The finger-tapping fMRI data is provided in [Maitra et al., 2002], the data contains 12 functional magnetic resonance imaging (fMRI) scans of the brain of a right hand dominant male subject during a right-hand finger-thumb opposition activity and 12 similar scans using the left-hand, with each pair of scan collected on 12 different occasions over 2 months. As suggested by Thompson et al. [2020], we focus our analysis on the 20th slice of the image volume with 128×128 pixels that previous work [Maitra, 2009, 2010] indicated as adequate to distinguish the activation between the left- and right-hand finger-tapping, there are total 24 slices acquired according to [Maitra et al., 2002], slice thickness is 6mm, with no gap between each of the slices. Again, follow [Thompson et al., 2020], we select a 20×20 section of the 20th slice having the left-topmost pixel at $(33, 67)$, which was the 20×20 section of the slice displaying the highest average activation in the left-hand activation

images as determined by the FAST-fMRI algorithm in [Almodóvar-Rivera and Maitra, 2019]. The goal of our analysis is to separate the $n = 24$ images into $K = 2$ clusters. BIC selected $u = (6, 6)$ for TEMM. As Web Table 8 demonstrates, 23 out of 24 images were correctly clustered by most clustering methods, we also notice that implement TEMM with $u = (1, 1)$ results in the same error 0.417, the one mislabeled case was identified as an outlier by Maitra [2010]. To further visualize the data and identify the outlier, we reduce the data to 1-dimensional with envelope basis estimated under $u = (1, 1)$ in TEMM. As illustrated in Figure 4, the two clusters estimated by TEMM are well separated, all observations are correctly clustered except for 1 outlier, which performs extremely different from its true cluster population and thus was misclassified.

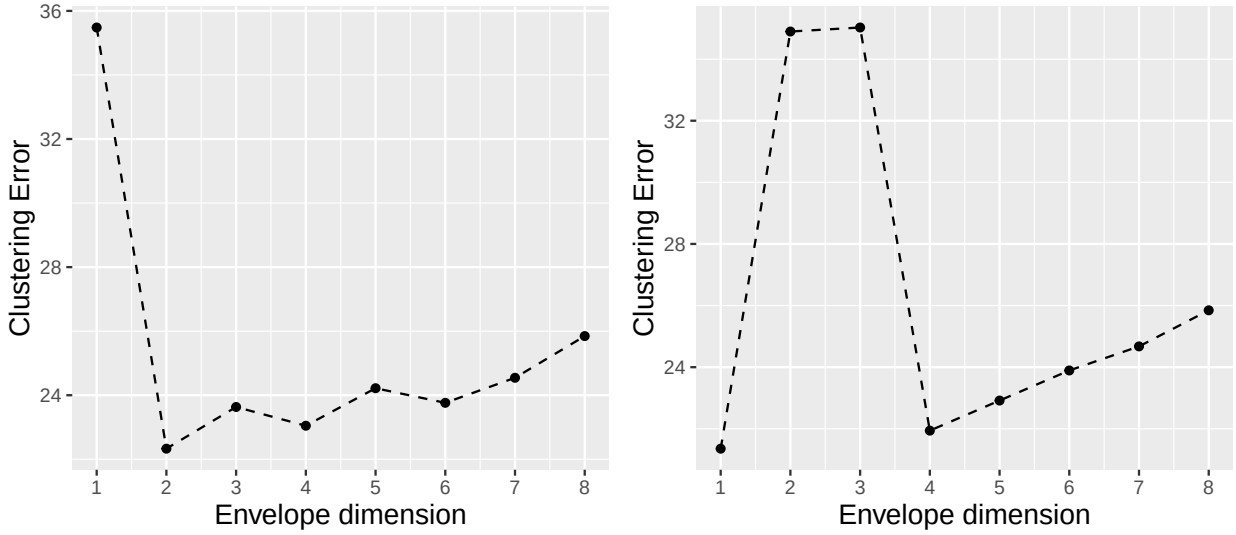


Web Figure 4: Scatter plot of true label Y versus 1-dimensional reduced data $\mathbf{X}_\star$ in TEMM clustering of Finger-Tapping fMRI data. The points on the bottom line $Y = 1$ represents observations from the first cluster, the points on the top line $Y = 2$ represents observations from the second cluster. Circle shaped points denote observations labeled into the first cluster by TEMM, triangle shaped points denote observations labeled into the second cluster by TEMM. The circle point on the top left is identified as an outlier.

C.2 Optical Recognition of Handwritten Digits Data

The Handwritten Digits data [Dua and Graff, 2017] collected bitmaps of handwritten digits from 30 people with a total sample size of $3,823$. A pre-processing programs extracted the normalized bitmaps of handwritten digits from a pre-printed form, and 32×32 bitmaps are divided into non-overlapping blocks of 4×4 and the number of on pixels are counted in each block. This generates an input matrix of 8×8 where each element is an integer in the range from 0 to 16. We analyze the $K = 4$ clusters problem using digits $\{1, 2, 4, 8\}$, results in a subset of $n = 1,536$ samples from the

original dataset. The proposed BIC tuning select envelope dimensions as $u = (8, 8)$ for TEMM, suggesting no reduction on the data. To assess how further dimension reduction would affect the clustering performance, we summarize the clustering results with $u_1 = 8$ and u_2 ranges from 1 to 8 in the left panel of Web Figure 5, clustering results with $u_2 = 1$ and u_1 ranges from 1 to 8 in the right panel of Web Figure 5. As we can see, when fix $u_1 = 8$ and increasing u_2 the clustering error tends to first decrease and then increase, indicating more envelope reduction impose on the second mode remove most irrelevant information and helps clustering, while the extreme reduction to 1-dimension would result in major information loss. The right panel in Web Figure 5 shares a similar tendency as left panel, except the deep clustering error drop when the first mode is reduced to 1-dimension, this may be caused by elimination of irrelevant information and clustering is thus improved. In Web Table 8, we include clustering result of TEMM with $u = (8, 8)$ as selected by BIC.



Web Figure 5: Clustering error (%) of Optical Recognition of Handwritten Digits dataset. Left panel keeps $u_1 = 8$ while u_2 ranges from 1 to 8, right panel keeps $u_2 = 8$ while u_1 ranges from 1 to 8.

C.3 Landsat Satellite Data

The Landsat satellite data [Dua and Graff, 2017] describes a set of multi-spectral satellite images that are in two visible and two infrared bands. We focus on the testing set of size $n = 845$ analyzed by Thompson et al. [2020] that consists of 3×3 pixel segments labeled according to the terrain type (397 gray soil, 211 damp gray soil, and 237 soil with vegetation stubble) of their middle pixel. The data is regarded as 4×9 matrices, and the problem is to separate the image observations into $K = 3$ soil types from the pixel values. The BIC selected envelope dimension as $u = (4, 1)$. As

Web Table 8 illustrates, the performance of TEMM and TGMM is the best among all clustering methods compared. The achieved clustering error is very close to the best classification error 0.107 achieved in [Thompson et al., 2020] while we include no information of training set in our analysis.

C.4 Cambridge Hand Gestures Data

The Cambridge hand gestures data contains 80 images of hand gestures extracted from the Cambridge hand gestures database [Kim et al., 2007]. Following Thompson et al. [2020], we analysis the data processed by Molstad and Rothman [2019] into 80×60 pixel gray-scale images. There are $K = 4$ classes in this problem: the images show a hand gesture in one of two shapes and one of two orientations: in each image, the hand is either in a flat or "V" shape and is located either in the center of the image or to the left side of the image. The BIC selected $u = (15, 15)$ for TEMM. As can be seen from Web Table 8, TEMM significantly outperforms other competing methods.

Web Table 8: Clustering of Gene Time Course dataset (GTC), Finger-Tapping fMRI dataset (FTF), Optical Recognition of Handwritten Digits dataset (ORHD), Landsat Satellite dataset (LS) and Cambridge Hand Gestures dataset (CHG). Reported are the clustering error ($\%$). Source code of AFPF failed to be implemented in ORHD and CHG datasets, source code of MCFA failed to be implemented in ORHD dataset.

Dataset	**TEMM-BIC**	TGMM	HD-GMM	KM(++)	PKM	SKM	AFPF	DTC	MCFA
GTC	24.53	37.74	33.96	37.74	37.74	37.74	37.74	39.62	35.85
FTF	4.17	4.17	4.17	4.17	4.17	4.17	8.33	4.17	16.67
ORHD	25.85	25.85	19.68	19.08	19.08	15.69	–	28.13	–
LS	12.90	12.90	40.72	16.33	16.33	14.79	16.21	44.38	36.57
CHG	13.75	27.50	18.75	28.75	30.00	32.50	–	23.75	23.75

D Proofs

D.1 Proof for Proposition 1

First, in TGMM, we can see there are $(K-1)\prod_{m=1}^{M} p_m + \sum_{m=1}^{M} p_m(p_m+1)/2$ free parameters in $\{\{\boldsymbol{\mu}_k\}_{k=1}^{K}, \{\boldsymbol{\Sigma}_m\}_{m=1}^{M}\}$. Then in TEMM, there are $(K-1)\prod_{m=1}^{M} u_m + \sum_{m=1}^{M}\{u_m(u_m+1)/2 + (p_m - u_m)(p_m - u_m + 1)/2 + (p_m - u_m)u_m\}$ free parameters in $\{\{\boldsymbol{\alpha}_k\}_{k=1}^{K}, \{\boldsymbol{\Gamma}_m, \boldsymbol{\Omega}_m, \boldsymbol{\Omega}_{0m}\}_{m=1}^{M}\}$. Finally, we have the total reduction in the number of free parameters is $(K-1)(\prod_{m=1}^{M} p_m - \prod_{m=1}^{M} u_m)$.

D.2 Proof for Proposition 2

Recall that Under the TGMM (4) that $\mathbf{X} \sim \sum_{k=1}^{K} \pi_k \mathrm{TN}(\boldsymbol{\mu}_k, \mathfrak{M})$, we want to show the equivalence between (6):$\boldsymbol{\mu}_k = [\![\boldsymbol{\alpha}_k; \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_M]\!]$, $\boldsymbol{\Sigma}_m = \boldsymbol{\Gamma}_m \boldsymbol{\Omega}_m \boldsymbol{\Gamma}_m^T + \boldsymbol{\Gamma}_{0m} \boldsymbol{\Omega}_{0m} \boldsymbol{\Gamma}_{0m}^T$ and (8): $\mathrm{span}(\boldsymbol{\beta}) \subseteq \mathcal{G}, \boldsymbol{\Sigma} = \mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{P}_{\mathcal{G}} + \mathbf{Q}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{Q}_{\mathcal{G}}$, where $\mathcal{G} = \mathcal{G}_M \otimes \cdots \otimes \mathcal{G}_1 \equiv \mathrm{span}(\boldsymbol{\Gamma}_M \otimes \cdots \otimes \boldsymbol{\Gamma}_1) \subseteq \mathbb{R}^p$ and $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}_M \otimes \cdots \otimes \boldsymbol{\Sigma}_1$.

First, we assume the envelope pursuit (6) is true, and try to prove (8) is true.

Let $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M \otimes \cdots \otimes \boldsymbol{\Sigma}_1$, $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_M \otimes \cdots \otimes \boldsymbol{\Gamma}_1$, by definition of $\mathbf{B}_k$ we can write $\boldsymbol{\beta}_k = \mathrm{vec}(\mathbf{B}_k) = \boldsymbol{\Sigma}^{-1} \mathrm{vec}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_1) = \boldsymbol{\Sigma}^{-1} \boldsymbol{\Gamma} \mathrm{vec}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_1)$, because $\boldsymbol{\Sigma}_m = \boldsymbol{\Gamma}_m \boldsymbol{\Omega}_m \boldsymbol{\Gamma}_m^T + \boldsymbol{\Gamma}_{0m} \boldsymbol{\Omega}_{0m} \boldsymbol{\Gamma}_{0m}^T = \boldsymbol{\Gamma}_m \boldsymbol{\Gamma}_m^T \boldsymbol{\Sigma}_m \boldsymbol{\Gamma}_m \boldsymbol{\Gamma}_m^T + \boldsymbol{\Gamma}_{0m} \boldsymbol{\Gamma}_{0m}^T \boldsymbol{\Sigma}_{0m} \boldsymbol{\Gamma}_{0m} \boldsymbol{\Gamma}_{0m}^T$, we can re-write $\boldsymbol{\Sigma}^{-1} \boldsymbol{\Gamma} = \boldsymbol{\Gamma} \boldsymbol{\Gamma}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Gamma}$, thus

$$\boldsymbol{\beta}_k = \boldsymbol{\Gamma} \boldsymbol{\Gamma}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Gamma} \mathrm{vec}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_1) = \boldsymbol{\Gamma} \boldsymbol{a}_k,$$

where $\boldsymbol{a}_k = \boldsymbol{\Gamma}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Gamma} \mathrm{vec}(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_1) \in \mathbb{R}^u, u = \prod_{m=1}^{M} u_m$, which implies $\boldsymbol{\beta}_k \in \mathcal{G} = \mathrm{span}\{\boldsymbol{\Gamma}\}$ for $k = 2, \ldots, K$, thus $\mathrm{span}\{\boldsymbol{\beta}\} \subseteq \mathcal{G}$. To prove $\boldsymbol{\Sigma} = \mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{P}_{\mathcal{G}} + \mathbf{Q}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{Q}_{\mathcal{G}}$, we show $\mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{Q}_{\mathcal{G}} = 0$. By $\boldsymbol{\Sigma}_m = \boldsymbol{\Gamma}_m \boldsymbol{\Omega}_m \boldsymbol{\Gamma}_m^T + \boldsymbol{\Gamma}_{0m} \boldsymbol{\Omega}_{0m} \boldsymbol{\Gamma}_{0m}^T$, we have $\boldsymbol{\Sigma}_m = \mathbf{P}_m \boldsymbol{\Sigma}_m \mathbf{P}_m + \mathbf{Q}_m \boldsymbol{\Sigma}_m \mathbf{Q}_m$, where $\mathbf{P}_m$ is the projection matrix onto $\mathcal{G}_m$, $\mathbf{Q}_m = \mathbf{I}_{p_m} - \mathbf{P}_m$ is the projection matrix onto complement space $\mathcal{G}_m^{\perp}$, therefore $\mathbf{P}_m \boldsymbol{\Sigma}_m = \mathbf{P}_m \boldsymbol{\Sigma}_m \mathbf{P}_m$, then

$$\begin{aligned}
\mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{Q}_{\mathcal{G}} &= \mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} - \mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{P}_{\mathcal{G}} \\
&= (\mathbf{P}_M \boldsymbol{\Sigma}_M) \otimes \cdots \otimes (\mathbf{P}_1 \boldsymbol{\Sigma}_1) - (\mathbf{P}_M \boldsymbol{\Sigma}_M \mathbf{P}_M) \otimes \cdots \otimes (\mathbf{P}_1 \boldsymbol{\Sigma}_1 \mathbf{P}_1) \\
&= 0.
\end{aligned}$$

Next, we suppose (8) is true and obtain (6) in the following.

To prove $\boldsymbol{\Sigma}_m = \boldsymbol{\Gamma}_m \boldsymbol{\Omega}_m \boldsymbol{\Gamma}_m^T + \boldsymbol{\Gamma}_{0m} \boldsymbol{\Omega}_{0m} \boldsymbol{\Gamma}_{0m}^T$, we show $\mathbf{P}_m \boldsymbol{\Sigma}_m = \mathbf{P}_m \boldsymbol{\Sigma}_m \mathbf{P}_m$, from (8) we have

$$\mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} - \mathbf{P}_{\mathcal{G}} \boldsymbol{\Sigma} \mathbf{P}_{\mathcal{G}} = (\mathbf{P}_M \boldsymbol{\Sigma}_M) \otimes \cdots \otimes (\mathbf{P}_1 \boldsymbol{\Sigma}_1) - (\mathbf{P}_M \boldsymbol{\Sigma}_M \mathbf{P}_M) \otimes \cdots \otimes (\mathbf{P}_1 \boldsymbol{\Sigma}_1 \mathbf{P}_1) = 0,$$

then we right-multiply $\mathbf{P}_M \otimes \cdots \otimes \mathbf{P}_{m+1} \otimes \mathbf{I}_{\mathbf{p}_m} \otimes \mathbf{P}_{m-1} \otimes \cdots \otimes \mathbf{P}_1$ to obtain:

$$(\mathbf{P}_M \boldsymbol{\Sigma}_M \mathbf{P}_M) \otimes \cdots \otimes (\mathbf{P}_m \boldsymbol{\Sigma}_m - \mathbf{P}_m \boldsymbol{\Sigma}_m \mathbf{P}_m) \otimes \cdots \otimes (\mathbf{P}_1 \boldsymbol{\Sigma}_1 \mathbf{P}_1) = 0,$$

which implies $\mathbf{P}_m \boldsymbol{\Sigma}_m \mathbf{P}_m = \mathbf{P}_m \boldsymbol{\Sigma}_m$, thus we have $\boldsymbol{\Sigma}_m = \boldsymbol{\Gamma}_m \boldsymbol{\Omega}_m \boldsymbol{\Gamma}_m^T + \boldsymbol{\Gamma}_{0m} \boldsymbol{\Omega}_{0m} \boldsymbol{\Gamma}_{0m}^T$, where $\boldsymbol{\Omega}_m = \boldsymbol{\Gamma}_m^T \boldsymbol{\Sigma}_m \boldsymbol{\Gamma}_m$, $\boldsymbol{\Omega}_{0m} = \boldsymbol{\Gamma}_{0m}^T \boldsymbol{\Sigma}_m \boldsymbol{\Gamma}_{0m}$, $m = 1, \ldots, M$. Now, we show there exits $\boldsymbol{\alpha}_k \in \mathbb{R}^{u_1 \times \cdots \times u_M}$ such that $\boldsymbol{\mu}_k = [\![\boldsymbol{\alpha}_k; \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_M]\!]$, $k = 1, \ldots, K$. From (8) we know that $\mathrm{span}\{\boldsymbol{\beta}\} \subseteq \mathcal{G}$, which gives us $\mathbf{B}_k = [\![\mathbf{B}_k; \boldsymbol{\Gamma}_1 \boldsymbol{\Gamma}_1^T, \ldots, \boldsymbol{\Gamma}_M \boldsymbol{\Gamma}_M^T]\!]$, $k = 2, \ldots, K$, therefore by definition we have

$$\begin{aligned}
\boldsymbol{\mu}_k - \boldsymbol{\mu}_1 &= [\![\mathbf{B}_k; \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M]\!] \\
&= [\![[\![\mathbf{B}_k; \boldsymbol{\Gamma}_1 \boldsymbol{\Gamma}_1^T, \ldots, \boldsymbol{\Gamma}_M \boldsymbol{\Gamma}_M^T]\!]; \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M]\!] \\
&= [\![[\![\mathbf{B}_k; \boldsymbol{\Gamma}_1 \boldsymbol{\Gamma}_1^T, \ldots, \boldsymbol{\Gamma}_M \boldsymbol{\Gamma}_M^T]\!]; \boldsymbol{\Gamma}_1 \boldsymbol{\Omega}_1 \boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_{01} \boldsymbol{\Omega}_{01} \boldsymbol{\Gamma}_{01}^T, \ldots, \boldsymbol{\Gamma}_M \boldsymbol{\Omega}_M \boldsymbol{\Gamma}_M^T + \boldsymbol{\Gamma}_{0M} \boldsymbol{\Omega}_{0M} \boldsymbol{\Gamma}_{0M}^T]\!] \\
&= [\![\mathbf{B}_k; \boldsymbol{\Gamma}_1 \boldsymbol{\Omega}_1 \boldsymbol{\Gamma}_1^T, \ldots, \boldsymbol{\Gamma}_M \boldsymbol{\Omega}_M \boldsymbol{\Gamma}_M^T]\!] \\
&= [\![[\![\mathbf{B}_k; \boldsymbol{\Omega}_1 \boldsymbol{\Gamma}_1^T, \ldots, \boldsymbol{\Omega}_M \boldsymbol{\Gamma}_M^T]\!]; \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_M]\!],
\end{aligned}$$

we also know $\sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k = 0$ since $\mathrm{E}(\mathbf{X}) = 0$, which implies that $\boldsymbol{\mu}_k$ can be written as $\boldsymbol{\mu}_k = [\![\boldsymbol{\alpha}_k; \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_M]\!]$, $k = 1, \ldots, K$.

D.3 Proof for Proposition 3

The proof is analogous to the proof of Proposition 3 in [Zhang and Li, 2017] and thus omitted.

D.4 Proof for Theorem 1(i)

We prove the results in a more general setting that $\overline{\boldsymbol{\mu}}$ and $\overline{\mathbf{X}}$ are no longer zeros. Let $\boldsymbol{\mu}_k - \overline{\boldsymbol{\mu}} = [\![\boldsymbol{\alpha}_k; \boldsymbol{\Gamma}_1, ..., \boldsymbol{\Gamma}_M]\!]$, it is straightforward to show the Q-function we defined in (9) has the following form:

$$Q(\boldsymbol{\theta}(\boldsymbol{\phi}) \mid \widehat{\boldsymbol{\theta}}^{(s-1)}) \simeq -\frac{n}{2}\log|\boldsymbol{\Sigma}| + \sum_{i=1}^{n}\sum_{k=1}^{K} \eta_{ik}^{(s)}\{\log \pi_k - \frac{1}{2}(\mathrm{vec}(\mathbf{X}_i - \overline{\boldsymbol{\mu}}) - \boldsymbol{\Gamma}\mathrm{vec}(\boldsymbol{\alpha}_k))^T \boldsymbol{\Sigma}^{-1}(\mathrm{vec}(\mathbf{X}_i - \overline{\boldsymbol{\mu}}) - \boldsymbol{\Gamma}\mathrm{vec}(\boldsymbol{\alpha}_k))\},$$

where symbol "$\simeq$" means equal up to an additive constant. Recall $\boldsymbol{\phi} = (\{\pi_k, \boldsymbol{\alpha}_k\}_{k=1}^K, \{\boldsymbol{\Gamma}_m, \boldsymbol{\Omega}_m, \boldsymbol{\Omega}_{0m}\}_{m=1}^M)$, $\boldsymbol{\theta} = (\{\boldsymbol{\mu}_k\}_{k=1}^K, \mathfrak{M}, \mathbf{B})$, our goal is to optimize the above Q-function over $\boldsymbol{\phi}$ by setting partial derivative to zero except for $\boldsymbol{\Gamma}_m$, which is treated as known, for $m = 1, \ldots, M$.

First, we maximize Q with respect to $\mathrm{vec}(\overline{\boldsymbol{\mu}})$ and obtain $\mathrm{vec}(\widehat{\overline{\boldsymbol{\mu}}}) = \mathrm{vec}(\overline{\mathbf{X}})$.

Next, for $\boldsymbol{\alpha}_k$, plug $\mathrm{vec}(\widehat{\overline{\boldsymbol{\mu}}})$ back into Q and maximize it with respect to $\mathrm{vec}(\boldsymbol{\alpha}_k)$, which results in $\mathrm{vec}(\widehat{\boldsymbol{\alpha}}_k^{(s)}) = \boldsymbol{\Gamma}^T \frac{\sum_{i=1}^n \eta_{ik}^{(s)} \mathrm{vec}(\mathbf{X}_i - \overline{\mathbf{X}})}{\sum_{i=1}^n \eta_{ik}^{(s)}} = \boldsymbol{\Gamma}^T \mathrm{vec}(\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}})$, where $\widetilde{\boldsymbol{\mu}}_k^{(s)} = \frac{\sum_{i=1}^n \eta_{ik}^{(s)} \mathbf{X}_i}{\sum_{i=1}^n \eta_{ik}^{(s)}}$.

Then we consider $\boldsymbol{\Omega}_m$ and $\boldsymbol{\Omega}_{0m}$ for each m. We first need to re-organize the terms in Q-function as follows.

Let $\boldsymbol{\nu}_k^{(s)} = [\![\widehat{\boldsymbol{\alpha}}_k^{(s)}; \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_M]\!] + \widehat{\overline{\boldsymbol{\mu}}} = [\![\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}; \boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T, \ldots, \boldsymbol{\Gamma}_M\boldsymbol{\Gamma}_M^T]\!] + \overline{\mathbf{X}}$, plug the estimates $\widehat{\overline{\boldsymbol{\mu}}}$ and $\widehat{\boldsymbol{\alpha}}_k^{(s)}$ in Q we have

$$\begin{aligned} Q(\boldsymbol{\theta}(\boldsymbol{\phi}) \mid \widehat{\boldsymbol{\theta}}^{(s-1)}) \simeq & -\frac{n}{2}(p_{-M}\log|\boldsymbol{\Sigma}_M| + \cdots p_{-1}\log|\boldsymbol{\Sigma}_1|) + \sum_{i=1}^{n}\sum_{k=1}^{K} \eta_{ik}^{(s)}\{\log \pi_k - \\ & \frac{1}{2}(\mathrm{vec}(\mathbf{X}_i - \overline{\mathbf{X}}) - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\mathrm{vec}(\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}))^T \boldsymbol{\Sigma}^{-1}(\mathrm{vec}(\mathbf{X}_i - \overline{\mathbf{X}}) - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\mathrm{vec}(\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}))\} \\ = & -\frac{n}{2}(p_{-M}\log|\boldsymbol{\Sigma}_M| + \cdots p_{-1}\log|\boldsymbol{\Sigma}_1|) + \\ & \sum_{i=1}^{n}\sum_{k=1}^{K} \eta_{ik}^{(s)}\{\log \pi_k - \frac{1}{2}\mathrm{vec}(\mathbf{X}_i - \boldsymbol{\nu}_k^{(s)})^T \boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{X}_i - \boldsymbol{\nu}_k^{(s)})\}, \end{aligned}$$

where $p_{-j} = \prod_{i \neq j} p_i$. Denote $\mathrm{vec}(\mathbf{X}_i - \boldsymbol{\nu}_k^{(s)})^T \boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{X}_i - \boldsymbol{\nu}_k^{(s)})$ as H_{ik}, we further rewrite

Q-function as

$$Q(\boldsymbol{\theta}(\boldsymbol{\phi}) \mid \widehat{\boldsymbol{\theta}}^{(s-1)}) \simeq -\frac{n}{2}(p_{-M}\log|\boldsymbol{\Sigma}_M| + \cdots p_{-1}\log|\boldsymbol{\Sigma}_1|) + \sum_{i=1}^{n}\sum_{k=1}^{K}\eta_{ik}^{(s)}\{\log\pi_k - \frac{1}{2}H_{ik}\}$$

Then, let $\boldsymbol{e}_{ik}^{(s)} = \mathbf{X}_i - \boldsymbol{\nu}_k^{(s)}$, we have the following representation of H_{ik}.

$$\begin{aligned}
H_{ik} &= \mathrm{vec}(\boldsymbol{e}_{ik}^{(s)})^T\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\boldsymbol{e}_{ik}^{(s)}) \\
&= \mathrm{vec}(\boldsymbol{e}_{ik}^{(s)})^T\mathrm{vec}(\llbracket\boldsymbol{e}_{ik}^{(s)}; \boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_M^{-1}\rrbracket) \\
&= \mathrm{vec}((\boldsymbol{e}_{ik}^{(s)})_{(m)})^T\mathrm{vec}(\llbracket\boldsymbol{e}_{ik}^{(s)}; \boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_M^{-1}\rrbracket_{(m)}) \\
&= \mathrm{tr}\big\{(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\llbracket\boldsymbol{e}_{ik}^{(s)}; \boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_M^{-1}\rrbracket_{(m)}\big\} \\
&= \mathrm{tr}\big\{(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}\big\} \\
&= \mathrm{tr}\big\{\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\big\} \\
&= \mathrm{tr}\big\{\boldsymbol{\Gamma}_m\boldsymbol{\Omega}_m^{-1}\boldsymbol{\Gamma}_m^T(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\big\} \\
&\quad + \mathrm{tr}\big\{\boldsymbol{\Gamma}_{0m}\boldsymbol{\Omega}_{0m}^{-1}\boldsymbol{\Gamma}_{0m}^T(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\big\} \\
&= \mathrm{tr}\big\{\boldsymbol{\Omega}_m^{-1}\boldsymbol{\Gamma}_m^T(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\boldsymbol{\Gamma}_m\big\} \\
&\quad + \mathrm{tr}\big\{\boldsymbol{\Omega}_{0m}^{-1}\boldsymbol{\Gamma}_{0m}^T(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\boldsymbol{\Gamma}_{0m}\big\},
\end{aligned}$$

recall $\boldsymbol{\Sigma}_{-m} = \boldsymbol{\Sigma}_M \otimes \cdots \otimes \boldsymbol{\Sigma}_{m+1} \otimes \boldsymbol{\Sigma}_{m-1} \otimes \cdots \otimes \boldsymbol{\Sigma}_1$.

Note $|\boldsymbol{\Sigma}_m| = |\boldsymbol{\Omega}_m||\boldsymbol{\Omega}_{0m}|$, now we can write the first derivative of Q-function w.r.t. $\boldsymbol{\Omega}_m$ as

$$\begin{aligned}
\frac{\partial Q}{\partial\boldsymbol{\Omega}_m} &= -\frac{np_{-m}}{2}\frac{\partial\log|\boldsymbol{\Omega}_m|}{\partial\boldsymbol{\Omega}_m} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\eta_{ik}^{(s)}\frac{\partial H_{ik}}{\partial\boldsymbol{\Omega}_m} \\
&= -\frac{np_{-m}}{2}\boldsymbol{\Omega}_m^{-1} + \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\eta_{ik}^{(s)}\boldsymbol{\Omega}_m^{-1}\boldsymbol{\Gamma}_m^T(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T\boldsymbol{\Gamma}_m\boldsymbol{\Omega}_m^{-1}.
\end{aligned}$$

Let the first derivative equals to zero leads to $\widehat{\boldsymbol{\Omega}}_m = \boldsymbol{\Gamma}_m^T\mathbf{S}_m^{(s)}\boldsymbol{\Gamma}_m$, where

$$\begin{aligned}
\mathbf{S}_m^{(s)} &= \frac{1}{np_{-m}}\sum_{i=1}^{n}\sum_{k=1}^{K}\eta_{ik}^{(s)}(\boldsymbol{e}_{ik}^{(s)})_{(m)}\boldsymbol{\Sigma}_{-m}^{-1}(\boldsymbol{e}_{ik}^{(s)})_{(m)}^T, \\
\boldsymbol{e}_{ik}^{(s)} &= \mathbf{X}_i - \boldsymbol{\nu}_k^{(s)} = \mathbf{X}_i - \overline{\mathbf{X}} - \llbracket\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}; \mathbf{P}_1, \ldots, \mathbf{P}_M\rrbracket.
\end{aligned}$$

Similarly, we have $\widehat{\boldsymbol{\Omega}}_{0m} = \boldsymbol{\Gamma}_{0m}^T\mathbf{S}_m^{(s)}\boldsymbol{\Gamma}_{0m}$. Note that

$$\begin{aligned}
\boldsymbol{\Gamma}_m^T(\boldsymbol{e}_{ik}^{(s)})_{(m)} &= \boldsymbol{\Gamma}_m^T(\mathbf{X}_i - \overline{\mathbf{X}} - \llbracket\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}; \mathbf{P}_1, \ldots, \mathbf{P}_M\rrbracket)_{(m)} \\
&= \boldsymbol{\Gamma}_m^T(\mathbf{X}_i - \overline{\mathbf{X}} - \llbracket\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}; \mathbf{P}_1, \ldots, \mathbf{P}_{m-1}, \mathbf{I}_{p_m}, \mathbf{P}_{m+1}, \ldots, \mathbf{P}_M\rrbracket)_{(m)}, \\
\boldsymbol{\Gamma}_{0m}^T(\boldsymbol{e}_{ik}^{(s)})_{(m)} &= \boldsymbol{\Gamma}_{0m}^T(\mathbf{X}_i - \overline{\mathbf{X}})_{(m)}.
\end{aligned}$$

Therefore, we get

$$\widehat{\boldsymbol{\Omega}}_m = \boldsymbol{\Gamma}_m^T \mathbf{M}_m^{(s)} \boldsymbol{\Gamma}_m, \quad \widehat{\boldsymbol{\Omega}}_{0m} = \boldsymbol{\Gamma}_{0m}^T \mathbf{N}_m^{(s)} \boldsymbol{\Gamma}_{0m},$$

where

$$\begin{aligned}
\mathbf{M}_m^{(s)} &= \frac{1}{np_{-m}} \sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(s)} (\boldsymbol{\epsilon}_{ik}^{(s)})_{(m)} \boldsymbol{\Sigma}_{-m}^{-1} (\boldsymbol{\epsilon}_{ik}^{(s)})_{(m)}^T, \\
\mathbf{N}_m^{(s)} &= \frac{1}{np_{-m}} \sum_{i=1}^{n} (\mathbf{X}_i - \overline{\mathbf{X}})_{(m)} \boldsymbol{\Sigma}_{-m}^{-1} (\mathbf{X}_i - \overline{\mathbf{X}})_{(m)}^T, \\
\boldsymbol{\epsilon}_{ik}^{(s)} &= \mathbf{X}_i - \overline{\mathbf{X}} - [\![\widetilde{\boldsymbol{\mu}}_k^{(s)} - \overline{\mathbf{X}}; \mathbf{P}_1, \ldots, \mathbf{P}_{m-1}, \mathbf{I}_{p_m}, \mathbf{P}_{m+1}, \ldots, \mathbf{P}_M]\!].
\end{aligned}$$

D.5 Proof for Theorem 1(i)

First, for each m, we plug the estimators obtained in Theorem 1(i) back into Q-function and rewrite it as following:

$$Q(\boldsymbol{\theta}(\boldsymbol{\phi}) \mid \widehat{\boldsymbol{\theta}}^{(s-1)}) \simeq -\frac{np_{-m}}{2} (\log |\boldsymbol{\Gamma}_m^T \mathbf{M}_m^{(s)} \boldsymbol{\Gamma}_m| + \log |\boldsymbol{\Gamma}_{0m}^T \mathbf{N}_m^{(s)} \boldsymbol{\Gamma}_{0m}|) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(s)} H_{ik}(\widehat{\boldsymbol{\Omega}}_m, \widehat{\boldsymbol{\Omega}}_{0m}).$$

Then note that

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(s)} H_{ik}(\widehat{\boldsymbol{\Omega}}_m, \widehat{\boldsymbol{\Omega}}_{0m}) &= \operatorname{tr}\big\{ \widehat{\boldsymbol{\Omega}}_m^{-1} \boldsymbol{\Gamma}_m^T \Big(\sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(s)} (\boldsymbol{e}_{ik}^{(s)})_{(m)} \boldsymbol{\Sigma}_{-m}^{-1} (\boldsymbol{e}_{ik}^{(s)})_{(m)}^T \Big) \boldsymbol{\Gamma}_m \big\} \\
&\quad + \operatorname{tr}\big\{ \widehat{\boldsymbol{\Omega}}_{0m}^{-1} \boldsymbol{\Gamma}_{0m}^T \Big(\sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(s)} (\boldsymbol{e}_{ik}^{(s)})_{(m)} \boldsymbol{\Sigma}_{-m}^{-1} (\boldsymbol{e}_{ik}^{(s)})_{(m)}^T \Big) \boldsymbol{\Gamma}_{0m} \big\} \\
&= np_{-m} (\operatorname{tr}\big\{ \widehat{\boldsymbol{\Omega}}_m^{-1} \widehat{\boldsymbol{\Omega}}_m \big\} + \operatorname{tr}\big\{ \widehat{\boldsymbol{\Omega}}_{0m}^{-1} \widehat{\boldsymbol{\Omega}}_{0m} \big\}) \\
&= np_{-m} (\operatorname{tr}\big\{ \mathbf{I}_{u_m} \big\} + \operatorname{tr}\big\{ \mathbf{I}_{p_m - u_m} \big\}) \\
&= np_{-m} p_m = np.
\end{aligned}$$

Therefore

$$\begin{aligned}
Q(\boldsymbol{\theta}(\boldsymbol{\phi}) \mid \widehat{\boldsymbol{\theta}}^{(s-1)}) &\simeq -\frac{np_{-m}}{2} (\log |\boldsymbol{\Gamma}_m^T \mathbf{M}_m^{(s)} \boldsymbol{\Gamma}_m| + \log |\boldsymbol{\Gamma}_{0m}^T \mathbf{N}_m^{(s)} \boldsymbol{\Gamma}_{0m}|) \\
&= -\frac{np_{-m}}{2} (\log |\boldsymbol{\Gamma}_m^T \mathbf{M}_m^{(s)} \boldsymbol{\Gamma}_m| + \log |\boldsymbol{\Gamma}_m^T (\mathbf{N}_m^{(s)})^{-1} \boldsymbol{\Gamma}_m| + \log |\mathbf{N}_m^{(s)}|).
\end{aligned}$$

Finally, for $m = 1, \ldots, M$, we get the conclusion that the envelope basis $\widehat{\boldsymbol{\Gamma}}_m(\widehat{\boldsymbol{\theta}}^{(s)})$ can be obtained as:

$$\widehat{\boldsymbol{\Gamma}}_m = \operatorname*{argmin}_{\boldsymbol{\Gamma}_m} G_m^{(s)}(\boldsymbol{\Gamma}_m) = \operatorname*{argmin}_{\boldsymbol{\Gamma}_m} \{ \log |\boldsymbol{\Gamma}_m^T \mathbf{M}_m^{(s)} \boldsymbol{\Gamma}_m| + \log |\boldsymbol{\Gamma}_m^T (\mathbf{N}_m^{(s)})^{-1} \boldsymbol{\Gamma}_m| \}.$$

E Review of ECD algorithm

The envelope coordinate descent (ECD) algorithm was proposed by Cook and Zhang [2018] that is fast and stable for envelope estimation problems. ECD algorithm was developed under the sequential 1D envelope algorithm framework [Cook and Zhang, 2016]. Therefore, to review the ECD algorithm, we first review the 1D algorithm.

E.1 The 1D algorithm

Suppose we want to estimate a generic envelope $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ where $\mathbf{M} > 0$ and $\mathbf{U} \geq 0$ are both in $\mathbb{S}^{p\times p}$. Then $\text{span}(\mathbf{U}) \subset \text{span}(M) = \mathbb{R}^p$ and the envelope is well-defined. In our problem, $\mathbf{M} = \mathbf{M}_m^{(s)}$, $\mathbf{U} = \mathbf{N}_m^{(s)} - \mathbf{M}_m^{(s)}$ for each m. Cook and Zhang [2016] proposed a generic objective function F for estimating $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$:

$$F(\mathbf{G}) = \log|\mathbf{G}^T\mathbf{M}\mathbf{G}| + \log|\mathbf{G}^T(\mathbf{M}+\mathbf{U})^{-1}\mathbf{G}|, \tag{C1}$$

where $\mathbf{G} \in \mathbb{R}^{p\times u}$ is semi-orthogonal with given envelope dimension $0 \leq u \leq p$. The 1D algorithm breaks down the u-dimensional Grassmannian optimization to a sequence of u one-dimensional optimization and is summarized in Algorithm 2.

Algorithm 2 The 1D algorithm [Cook and Zhang, 2016].

Let $g_k \in \mathbb{R}^p$, $k = 1, \ldots, u$, be the sequential directions obtained. Let $\mathbf{G}_k = (\mathbf{g}_1, \ldots, \mathbf{g}_k)$, let $(\mathbf{G}_k, \mathbf{G}_{0k})$ be an orthogonal basis for $\mathbb{R}^p$ and set initial value $\mathbf{g}_0 = \mathbf{G}_0 = 0$.
For $k = 0, \ldots, u-1$, repeat Step 1 and 2 in the following.

1. Let $\mathbf{G}_k = (\mathbf{g}_1, \ldots, \mathbf{g}_k)$, and let $(\mathbf{G}_k, \mathbf{G}_{0k})$ be an orthogonal basis for $\mathbb{R}^p$. Set $\mathbf{N}_k = [\mathbf{G}_{0k}^T(\mathbf{M}+\mathbf{U})\mathbf{G}_{0k}]^{-1}$, $\mathbf{M}_k = \mathbf{G}_{0k}^T\mathbf{M}\mathbf{G}_{0k}$ and the unconstrained objective function

$$\phi_k(\mathbf{w}) = \log(\mathbf{w}^T\mathbf{M}_k\mathbf{w}) + \log(\mathbf{w}^T\mathbf{N}_k\mathbf{w}) - 2\log(\mathbf{w}^T\mathbf{w}). \tag{C2}$$

2. Solve $\mathbf{w}_{k+1} = \text{argmin}\,\phi_k(\mathbf{w})$, then the $(k+1)$-th envelope direction is $\mathbf{g}_k = \mathbf{G}_{0k}\mathbf{w}/\|\mathbf{w}_{k+1}\|$.

The 1D algorithm requires no initial guess, yields $\sqrt{n}$-consistent estimators under mild conditions and was demonstrated to be much faster than a commonly used algorithm based on direct optimization over the appropriate Grassmannian.

E.2 The ECD algorithm

The 1D algorithm minimizes $\phi_k(\mathbf{w})$ iteratively for each direction $\mathbf{w}_{k+1}$, where the objective function $\phi_k(\mathbf{w})$ is non-convex and has local minima. The ECD algorithm was proposed for solving $\phi_k(\mathbf{w})$

and is much faster and stabler than any standard nonlinear optimization method and is guaranteed to not increase the value of the objective function at each iteration. Under the 1D algorithm framework, the part of ECD algorithm for solving $\phi_k(\mathbf{w})$ in (C2) of Algorithm 2 is summarized in Algorithm 3.

Algorithm 3 The ECD algorithm [Cook and Zhang, 2018] for solving $\phi_k(\mathbf{w})$.

1. Eigenvalue decomposition of $\mathbf{M}_k$ as $\mathbf{M}_k = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ where $\mathbf{V}$ is an orthogonal matrix and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{p-k})$ is a diagonal matrix.

2. Transform the original objective function into canonical coordinates: $\mathbf{V}^T\mathbf{w} \mapsto \mathbf{v}$, $\mathbf{V}^T\mathbf{N}_k\mathbf{V} \mapsto \widetilde{\mathbf{N}}$ and
$$\phi_k(\mathbf{w}) = \varphi_k(\mathbf{v}) = \log(\mathbf{v}^T\mathbf{\Lambda}\mathbf{v}) + \log(\mathbf{v}^T\widetilde{\mathbf{N}}) - 2\log(\mathbf{v}^T\mathbf{v}). \tag{C3}$$

3. For $t = 1, \ldots, T_{max}$, where T_{max} is the maximum number of iterations, update $\mathbf{v}^{(t)}$ following Step 4-7 and terminate iteration if $\varphi_k(\mathbf{v}^{(t)}) - \varphi_k(\mathbf{v}^{(t-1)}) \leq \epsilon$ for some tolerance value $\epsilon > 0$. At the termination, transform back to $\mathbf{w}_{k+1} = \mathrm{argmin}\, \phi_k(\mathbf{w}) = \mathbf{V}\mathbf{v}$.

4. Update $a^{(t)} \leftarrow (\mathbf{v}^T\mathbf{\Lambda}\mathbf{v})^{-1}$, $b^{(t)} \leftarrow (\mathbf{v}^T\widetilde{\mathbf{N}}\mathbf{v})^{-1}$, $c^{(t)} \leftarrow (\mathbf{v}\mathbf{v}^T)^{-1}$ according to the current stage $\mathbf{v}^{(t)}$.

5. For $j = 1, \ldots, p-k$, if $a^{(t)}\lambda_j + b^{(t)}\widetilde{\mathbf{N}}_{jj} - 2c^{(t)} \neq 0$ then consider moving each coordinate of $\mathbf{v}$ as
$$v_j^{(t+1)} \leftarrow \left(\sum_{i \neq j}^{p-k} b^{(t)}\widetilde{N}_{ij}v_i^{(t)}\right) \Big/ \left(a^{(t)}\lambda_j + b^{(t)}\widetilde{\mathbf{N}}_{jj} - 2c^{(t)}\right). \tag{C4}$$

6. If the objective function is not decreased by moving $v_j^{(t)}$ to $v_j^{(t+1)}$ then back up $v_j^{(t+1)}$ to $v_j^{(t)}$.

7. If none of the coordinates is updated, run one iteration of any standard nonlinear optimization method to update $\mathbf{v}$.

The ECD algorithm transforms the basis to canonical coordinate $\mathbf{w} \mapsto \mathbf{v}$ so that the first term in the objective function become more separable: $\log(\mathbf{w}^T\mathbf{M}_k\mathbf{w}) \mapsto \log(\mathbf{v}^T\mathbf{\Lambda}\mathbf{v}) = \log(\sum_{i=1}\lambda_i v_i^2)$, which in fact speeds up the algorithm and make the optimization more accurate. Step 5 in Algorithm 3 approximates the solution to $\partial\varphi_k(\mathbf{v})/\partial v_j = 0$, which can be written as

$$\frac{2\lambda_j v_j}{\mathbf{v}^T\mathbf{\Lambda}\mathbf{v}} + \frac{2\sum_{i=1}^{p-k}\widetilde{N}_{ij}v_i}{\mathbf{v}^T\widetilde{\mathbf{N}}\mathbf{v}} - \frac{4v_j}{\mathbf{v}^T\mathbf{V}} = 0.$$

The approximate solution is obtained by treating the denominators $\mathbf{v}^T\mathbf{\Lambda}\mathbf{v}$, $\mathbf{v}^T\widetilde{\mathbf{N}}\mathbf{v}$ and $\mathbf{v}^T\mathbf{v}$ as constants at the current step, and solving the resulting linear equation in v_j from the numerators. Step 6 is a back-tracking step that ensures the objective function to be monotonically non-decreasing.

Step 7 guarantees the algorithm will converge results from the basic properties of the standard nonlinear optimization method chosen in Step 7. In practice, the solution in Step 5 approximates the true minimizer for the coordinates very well and in rare situations Step 7 is needed.

The $\sqrt{n}$-consistency of the ECD algorithm follows as a result of the 1D algorithm consistency and also because the ECD algorithm is guaranteed to solve $\phi_k(\mathbf{w})$ from Step 6 and 7 of Algorithm 3.

F Theoretical justification for envelope dimension selection

In this section, we review the theoretical justification of the BIC type selection for envelope methods [Zhang and Mai, 2018]. We emphasize the application of envelopes do not rely on any specific model, in a general statistical estimation problem of some parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, Cook and Zhang [2015] generalized the notion of envelopes as a way to improve some "standard" existing $\sqrt{n}$-consistent estimator $\widehat{\boldsymbol{\theta}}$. To estimate a semi-orthogonal envelope basis matrix, we solve for $\widehat{\boldsymbol{\Gamma}} \in \mathbb{R}^{p\times u}$ that minimizes the generic moment-based objective function:

$$\mathrm{J}_n(\boldsymbol{\Gamma}) = \log|\boldsymbol{\Gamma}^T\widehat{\mathbf{M}}\boldsymbol{\Gamma}| + \log|\boldsymbol{\Gamma}^T(\widehat{\mathbf{M}}+\widehat{\mathbf{U}})^{-1}\boldsymbol{\Gamma}|. \tag{C1}$$

Given the true envelope dimension u, the $\sqrt{n}$-consistency of the estimated envelope is established by Cook and Zhang [2016]. Zhang and Mai [2018] showed that $\mathrm{J}_n(\boldsymbol{\Gamma})$ can be viewed as a quasi-likelihood function that directly leads to the envelope dimension selection theory we review in the following.

F.1 Envelope dimension selection

It can be shown that $\mathrm{J}_n(\boldsymbol{\Gamma})$ defined in (C1) converges uniformly in probability to its population counterpart $\mathrm{J}(\boldsymbol{\Gamma}) = \log|\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma}| + \log|\boldsymbol{\Gamma}^T(\mathbf{M}+\mathbf{U})^{-1}\boldsymbol{\Gamma}|$. To distinguish estimators at different envelope working dimensions, let $\boldsymbol{\Gamma}_k$ and $\widehat{\boldsymbol{\Gamma}}_l \in \mathbb{R}^{p\times k}$ denote the minimizers of the population objective function $\mathrm{J}(\boldsymbol{\Gamma})$ and the sample objective function $\mathrm{J}_n(\boldsymbol{\Gamma})$ at dimension k, define $\mathrm{J}_n(\boldsymbol{\Gamma}_k) = \mathrm{J}(\boldsymbol{\Gamma}) = 0$ for $k = 0$, Zhang and Mai [2018] proposed the following results.

Lemma 1. *If $u = 0$, then $\mathrm{J}(\boldsymbol{\Gamma}_k) = 0$ for all $k = 0, \ldots, p$. If $u > 0$, then $\mathrm{J}(\boldsymbol{\Gamma}_u) < \mathrm{J}(\boldsymbol{\Gamma}_k) < 0$, for $0 < k < u$, and $\mathrm{J}(\boldsymbol{\Gamma}_k) = \mathrm{J}(\boldsymbol{\Gamma}_u) < 0$, for $k \geq u$. Moreover, for $0 \leq u < k$, $\mathcal{E}_{\mathbf{M}}(\mathbf{U}) \subset \mathrm{span}(\boldsymbol{\Gamma}_k)$.*

Lemma 1 shows that, $\mathrm{J}(\boldsymbol{\Gamma}_k)$ is strictly greater than $\mathrm{J}(\boldsymbol{\Gamma}_u)$ when $k < u$, and remains constant once k exceeds u. Thus Zhang and Mai [2018] proposed to select envelope dimension selection via minimizing the following criterion,

$$\mathcal{I}_n(k) \equiv \mathrm{J}_n(\widehat{\boldsymbol{\Gamma}}_k) + \frac{C\cdot k\cdot\log(n)}{n}, \quad k = 0, 1, \ldots, p, \tag{C2}$$

where $C > 0$ is a constant and $\mathcal{I}_n(0) = 0$. The envelope dimension is selected as $\widehat{u}_{FG} = \text{argmin}_{0\leq k\leq p}\mathcal{I}_n(k)$, where subscript FG denote full Grassmannian optimization of $\text{J}_n(\mathbf{\Gamma})$. This criterion has a form similar to the Bayesian information criterion, but has the fundamental difference that $\text{J}_n(\widehat{\mathbf{\Gamma}}_k)$ is not a likelihood function. However, Zhang and Mai [2018] showed (C2) leads to consistent dimension selection without likelihood arguments.

Theorem 1. *For any constant $C > 0$ and $\sqrt{n}$-consistent $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{U}}$ in* (C2)*, we have* $\Pr(\widehat{u}_{FG} = u) \to 1$ *as* $n \to \infty$.

Some remarks about Theorem 1. First, it reveals that the choice of C does not affect the consistency of the dimension selection procedure. More discussion on the role of C see [Zhang and Mai, 2018]. Second, (C2) can be applied to any models with envelope structure. Third, in the heavily-studied case of multivariate linear regression model, $\text{J}_n(\mathbf{\Gamma})$ will reproduce the normal likelihood-based objective function if appropriate choices of $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{U}}$ are plugged in [Cook and Zhang, 2015]. In our paper, we chose C to be 1 as suggested by Zhang and Mai [2018].

References

Israel Almodóvar-Rivera and Ranjan Maitra. Fast adaptive smoothing and thresholding for improved activation detection in low-signal fmri. *IEEE Transactions on Medical Imaging*, 38(12):2821–2828, 2019.

R Dennis Cook and Xin Zhang. Foundations for envelope models and methods. *Journal of the American Statistical Association*, 110(510):599–611, 2015.

R Dennis Cook and Xin Zhang. Algorithms for envelope estimation. *Journal of Computational and Graphical Statistics*, 25(1):284–300, 2016.

R Dennis Cook and Xin Zhang. Fast envelope algorithms. *Statistica Sinica*, 28(3):1179–1197, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

Ranjan Maitra. Assessing certainty of activation or inactivation in test–retest fmri studies. *Neuroimage*, 47(1):88–97, 2009.

Ranjan Maitra. A re-defined and generalized percent-overlap-of-activation measure for studies of fmri reproducibility and its use in identifying outlier activation maps. *Neuroimage*, 50(1): 124–135, 2010.

Ranjan Maitra, Steven R Roys, and Rao P Gullapalli. Test-retest reliability estimation of functional mri data. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 48(1):62–70, 2002.

Aaron J Molstad and Adam J Rothman. A penalized likelihood method for classification with matrix-valued predictors. *Journal of Computational and Graphical Statistics*, 28(1):11–22, 2019.

Geoffrey Z Thompson, Ranjan Maitra, William Q Meeker, and Ashraf F Bastawros. Classification with the matrix-variate-t distribution. *Journal of Computational and Graphical Statistics*, pages 1–7, 2020.

Xin Zhang and Lexin Li. Tensor envelope partial least-squares regression. *Technometrics*, 59(4): 426–436, 2017.

Xin Zhang and Qing Mai. Model-free envelope dimension selection. *Electronic Journal of Statistics*, 12(2):2193–2216, 2018.