Supplementary Materials for "Joint and Individual Component Regression"

A Proofs

Proof of Lemma 1. The proof of the first part of this lemma follows similar arguments as in Feng et al. (2018). Define $\text{row}(\mathbf{J}) = \bigcap_{g=1}^{G} \text{row}(\tilde{\mathbf{X}}_g)$. We assume that $\text{row}(\mathbf{J}) \neq \{\mathbf{0}\}$ for non-trivial cases. For each g, $\mathbf{J}_g$ and $\mathbf{A}_g$ can be obtained by projecting $\tilde{\mathbf{X}}_g$ to $\text{row}(\mathbf{J})$ and its orthogonal complement in $\text{row}(\tilde{\mathbf{X}}_g)$. Thus, by construction, we have $\tilde{\mathbf{X}}_g = \mathbf{J}_g + \mathbf{A}_g$ and $\text{row}(\mathbf{J}_g) \perp \text{row}(\mathbf{A}_g)$. Since for all g, we have $\text{row}(\mathbf{A}_g) \perp \text{row}(\mathbf{J})$, then $\bigcap_{g=1}^{G} \text{row}(\mathbf{A}_g)$ has a zero projection matrix. Therefore, we have $\bigcap_{g=1}^{G} \text{row}(\mathbf{A}_g) = \{\mathbf{0}\}$. On the other hand, since $\text{row}(\mathbf{J}_g) \subset \text{row}(\tilde{\mathbf{X}}_g)$, the orthogonal projection of $\tilde{\mathbf{X}}_g$ onto $\text{row}(\mathbf{J}_g)$ is unique. Therefore, the matrices $\mathbf{J}_g$ and $\mathbf{A}_g$ are uniquely defined.

For the second part of Lemma 1, note that $\tilde{\boldsymbol{Y}}_g \in \text{col}(\tilde{\mathbf{X}}_g) \subset \text{col}(\mathbf{J}_g) + \text{col}(\mathbf{A}_g)$. Then, $\boldsymbol{J}_g^Y$ and $\boldsymbol{A}_g^Y$ can be obtained by projecting $\tilde{\boldsymbol{Y}}_g$ onto $\text{col}(\mathbf{J}_g)$ and $\text{col}(\mathbf{A}_g)$ respectively. Because $\text{col}(\mathbf{J}_g) \perp \text{col}(\mathbf{A}_g)$, $\boldsymbol{J}_g^Y$ and $\boldsymbol{A}_g^Y$ are uniquely defined.

Proof of Lemma 2. We explicitly describe how to find $\mathbf{J}_g$, $\mathbf{A}_g$, $\mathbf{U}$, $\mathbf{U}_g$, $\mathbf{S}_g$, $\mathbf{T}_g$, $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}_g$ to satisfy the requirements. First, $\mathbf{J}_g$ can be obtained by finding an arbitrary set of bases that span $\bigcap_{g=1}^{G} \text{row}(\mathbf{X}_g)$. Then, $\mathbf{A}_g$ can be obtained by solving a system of linear equations. We prove that such $\mathbf{J}_g$ and $\mathbf{A}_g$ satisfy conditions (ii)–(iv) in Lemma 1 and $\text{col}(\mathbf{J}_g) \perp \text{col}(\mathbf{A}_g)$. Given $\mathbf{J}_g$ and $\mathbf{A}_g$, we construct $\mathbf{U}, \mathbf{U}_g, \mathbf{S}_g, \mathbf{T}_g$, so that they satisfy $\mathbf{J}_g = \mathbf{S}_g\mathbf{U}$, $\mathbf{A}_g = \mathbf{T}_g\mathbf{U}_g$ and $\mathbf{S}_g'\mathbf{T}_g = \mathbf{0}_{K \times K_g}$.

Let $\mathbf{w}_1, \ldots, \mathbf{w}_K \in \mathbb{R}^p$ be an arbitrary set of bases that span $\bigcap_{g=1}^{G} \text{row}(\mathbf{X}_g)$. Denote $\mathbf{W}_{p \times K} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$. Let $\mathbf{J}_g = \mathbf{X}_g\mathbf{W}(\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}'$. Next, we show that $\mathbf{J}_g$ satisfies condition (i) in Lemma 1 for any $0 \leq g_1 \neq g_2 \leq G$. It suffices to show that $\text{row}(\mathbf{J}_{g_1}) \subset \text{row}(\mathbf{J}_{g_2})$, i.e. there exists $\mathbf{C}^{g_2}_{g_1, n_{g_1} \times n_{g_2}}$, such that $\mathbf{C}^{g_2}_{g_1}\mathbf{J}_{g_2} = \mathbf{J}_{g_1}$. Since the columns of $\mathbf{W}$ form the bases of $\bigcap_{g=1}^{G} \text{row}(\mathbf{X}_g)$, there exists $\mathbf{Q}^{g_2}_{K \times n_{g_2}}$, such that $\mathbf{Q}^{g_2}\mathbf{X}_{g_2} = \mathbf{W}'$. Then, we have $\mathbf{Q}^{g_2}\mathbf{J}_{g_2} = \mathbf{Q}^{g_2}\mathbf{X}_{g_2}\mathbf{W}(\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}' = (\mathbf{W}'\mathbf{W})(\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}' = \mathbf{W}'$. Let $\mathbf{R}_{g_1} =$

$\mathbf{X}_{g_1}\mathbf{W}(\mathbf{W}'\mathbf{W})^{-1}$ and $\mathbf{C}_{g_1}^{g_2} = \mathbf{R}_{g_1}\mathbf{Q}^{g_2}$, then we have

$$\mathbf{C}_{g_1}^{g_2}\mathbf{J}_{g_2} = \mathbf{R}_{g_1}\mathbf{Q}^{g_2}\mathbf{J}_{g_2} = \mathbf{X}_{g_1}\mathbf{W}(\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}' = \mathbf{J}_{g_1}.$$

Given $\mathbf{W}$, we propose to solve

$$\begin{bmatrix} \mathbf{W}' \\ \mathbf{W}'\mathbf{X}_g'\mathbf{X}_g \end{bmatrix} \mathbf{x} = \mathbf{0}, \tag{A.1}$$

where $\mathbf{x} \in \mathbb{R}^p$ is unknown. We first show that (A.1) has non-zero solutions. Since $\mathbf{Q}^g\mathbf{X}_g = \mathbf{W}'$, (A.1) can be rewritten as $\tilde{\mathbf{Q}}^g\mathbf{B}_g\mathbf{X}_g\mathbf{x} = \mathbf{0}$, where

$$\tilde{\mathbf{Q}}^g = \begin{pmatrix} \mathbf{Q}^g & \\ & \mathbf{Q}^g \end{pmatrix} \text{ and } \mathbf{B}_g = \begin{pmatrix} \mathbf{I}_{n_g \times n_g} \\ \mathbf{X}_g\mathbf{X}_g' \end{pmatrix}.$$

Since

$$\text{rank}(\tilde{\mathbf{Q}}^g\mathbf{B}_g\mathbf{X}_g) \leq \min\{\text{rank}(\tilde{\mathbf{Q}}^g), \text{rank}(\mathbf{B}_g), \text{rank}(\mathbf{X}_g)\} \leq \min\{2K, n_g, \text{rank}(\mathbf{X}_g)\} \leq \text{rank}(\mathbf{X}_g) < p,$$

(A.1) has non-zero solutions.

Let $\mathbf{W}_g = [\mathbf{w}_{g,1}, \ldots, \mathbf{w}_{g,K_g}]$ be an arbitrary set of bases that spans the space of solutions to (A.1). Let $\mathbf{A}_g = \mathbf{X}_g\mathbf{W}_g(\mathbf{W}_g'\mathbf{W}_g)^{-1}\mathbf{W}_g'$. Next we show that $\mathbf{A}_g$ satisfies $(ii) - (iii)$ in Lemma 1 and $\text{col}(\mathbf{A}_g) \perp \text{col}(\mathbf{J}_g)$. Since columns of $\mathbf{W}_g$ satisfy (A.1), we have $\mathbf{W}'\mathbf{W}_g = \mathbf{0}$ and $\mathbf{W}'\mathbf{X}_g'\mathbf{X}_g\mathbf{W}_g = \mathbf{0}$. Then, by definition, $\mathbf{A}_g'\mathbf{J}_g = \mathbf{0}$ and $\mathbf{J}_g\mathbf{A}_g' = \mathbf{0}$, which imply that $\text{row}(\mathbf{A}_g) \perp \text{row}(\mathbf{J}_g)$ and $\text{col}(\mathbf{A}_g) \perp \text{col}(\mathbf{J}_g)$. Since for all g, $\text{row}(\mathbf{A}_g)$ is perpendicular to the columns of $\mathbf{W}$, then we have $\bigcap_{g=1}^G \text{row}(\mathbf{A}_g) = \{\mathbf{0}\}$.

Finally, letting $\mathbf{S}_g = \mathbf{X}_g\mathbf{W}$, $\mathbf{T}_g = \mathbf{X}_g\mathbf{W}_g$, $\mathbf{U} = (\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}'$ and $\mathbf{U}_g = (\mathbf{W}_g'\mathbf{W}_g)^{-1}\mathbf{W}_g'$, we have $\mathbf{S}_g'\mathbf{T}_g = \mathbf{0}$. Let $\boldsymbol{\alpha} = (\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\boldsymbol{Y}$ and $\boldsymbol{\alpha}_g = (\mathbf{T}_g'\mathbf{T}_g)^{-1}\mathbf{T}_g'\boldsymbol{Y}_g$, where $\mathbf{S} = \mathbf{X}\mathbf{W}$. Then we obtain $\boldsymbol{J}_g^Y$ and $\boldsymbol{A}_g^Y$ by letting $\boldsymbol{J}_g^Y = \mathbf{S}_g\boldsymbol{\alpha} \in \text{col}(\mathbf{J}_g)$ and $\boldsymbol{A}_g^Y = \mathbf{T}_g\boldsymbol{\alpha}_g \in \text{col}(\mathbf{A}_g)$.

B Derivation of the CR algorithm for solving (3.5) and (3.6)

Consider the following CR problem that covers (3.5) and (3.6) as special cases:

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathrm{cov}(\mathbf{X}\mathbf{w}, \boldsymbol{Y})^2 \mathrm{var}(\mathbf{X}\mathbf{w})^{\gamma-1} \\ \text{s.t.} \quad & \mathbf{w}'\mathbf{w} = 1, \\ & \mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w}_j = 0; \quad j = 1, \ldots, k-1 \text{ if } k \geq 2, \\ & \hat{\mathbf{W}}'\mathbf{w} = \mathbf{0}; \\ & \hat{\mathbf{S}}'\mathbf{X}\mathbf{w} = \mathbf{0}, \end{aligned} \tag{B.1}$$

where $\hat{\mathbf{W}}$, $\hat{\mathbf{S}}$, $\mathbf{X}$ and $\boldsymbol{Y}$ are given a priori.

The solution to (S2) resides in $\mathrm{col}(\hat{\mathbf{W}})^{\perp}$, i.e. the space that is orthogonal to the columns of $\hat{\mathbf{W}}$. Hence, we let $\hat{\mathbf{P}} = \hat{\mathbf{W}}(\hat{\mathbf{W}}'\hat{\mathbf{W}})^{-1}\hat{\mathbf{W}}'$ and $\hat{\mathbf{X}} = \mathbf{X}(\mathbf{I} - \hat{\mathbf{P}})$, the latter being the projection of $\mathbf{X}$ into the space that is orthogonal to the columns of $\hat{\mathbf{W}}$. Let m be the rank of the matrix $\hat{\mathbf{X}}$, and $\hat{\mathbf{X}} = \mathbf{U}\mathbf{D}\mathbf{V}'$ be the corresponding rank-m SVD decomposition, with $\mathbf{U} \in \mathbb{R}^{n \times m}$, $\mathbf{V} \in \mathbb{R}^{p \times m}$, $\mathbf{D}$ the $m \times m$ diagonal matrix. Since the representation of the solution to (S2) might not be unique, to avoid ambiguity, we write the solution as the linear combination of the column vectors of $\mathbf{V}$, i.e. $\mathbf{w} = \mathbf{V}\mathbf{z}$, for some $\mathbf{z} \in \mathbb{R}^m$. Note that all $\mathbf{w} = \mathbf{V}\mathbf{z}$ satisfies $\hat{\mathbf{W}}'\mathbf{w} = \mathbf{0}$. Hence, the constraint $\hat{\mathbf{W}}'\mathbf{w} = \mathbf{0}$ from (S2) can be satisfied under this representation.

At step $k+1$, the original optimization problem can be reformulated as follows:

$$\max_{\mathbf{z}} (\mathbf{z}'\mathbf{d})^2 (\mathbf{z}'\tilde{\mathbf{E}}\mathbf{z})^{\gamma-1} \quad \text{s.t. } \mathbf{z}'\mathbf{z} = 1, \mathbf{z}'\tilde{\mathbf{E}}\mathbf{Z}_k = \mathbf{0}, \text{ and } \mathbf{z}'\mathbf{D}\mathbf{U}'\hat{\mathbf{S}} = \mathbf{0}, \tag{B.2}$$

where $\mathbf{d} = \mathbf{V}'\hat{\mathbf{X}}'\boldsymbol{Y}$, $\mathbf{Z}_k = [\mathbf{z}_1, \ldots, \mathbf{z}_k]$ and $\tilde{\mathbf{E}} = \mathbf{D}^2$. To solve (B.2), we can expand the objective to its Lagrangian form:

$$T^*(\mathbf{z}) = (\mathbf{z}'\mathbf{d})^2 (\mathbf{z}'\tilde{\mathbf{E}}\mathbf{z})^{\gamma-1} - \lambda_0(\mathbf{z}'\mathbf{z} - 1) - 2\mathbf{z}'[\tilde{\mathbf{E}}\mathbf{Z}_k \ \mathbf{D}\mathbf{U}'\hat{\mathbf{S}}]\boldsymbol{\Lambda}_k,$$

where $\mathbf{\Lambda}_k = [\lambda_1, \ldots, \lambda_{k+K}]'$ and $\lambda_0, \ldots, \lambda_{k+K}$ are Lagrange multipliers. To solve (B.2), we take the derivative of T^* with respect to $\mathbf{z}$, then the optimizer should be the solution to the following:

$$\frac{\partial T^*}{\partial \mathbf{z}} = 2(\mathbf{z}'\mathbf{d})(\mathbf{z}'\tilde{\mathbf{E}}\mathbf{z})^{\gamma-1}\mathbf{d} + 2(\gamma-1)(\mathbf{z}'\mathbf{d})^2(\mathbf{z}'\tilde{\mathbf{E}}\mathbf{z})^{\gamma-2}\tilde{\mathbf{E}}\mathbf{z} - 2\lambda_0\mathbf{z} - 2[\tilde{\mathbf{E}}\mathbf{Z}_k \ \mathbf{D}\mathbf{U}'\hat{\mathbf{S}}]\mathbf{\Lambda}_k = \mathbf{0}. \quad \text{(B.3)}$$

Left multiply $\mathbf{z}'$ to (B.3) and apply the constraints then we can conclude that $\lambda_0 = \gamma(\mathbf{z}'\mathbf{d})^2(\mathbf{z}'\tilde{\mathbf{E}}\mathbf{z})^{\gamma-1}$. Plug this back to (B.3), and let $\tau = \mathbf{z}'\mathbf{d}$ and $\rho = \mathbf{z}'\tilde{\mathbf{E}}\mathbf{z}$, then we have

$$\gamma\tau^2\rho^{\gamma-1}\mathbf{z} + (1-\gamma)\tau^2\rho^{\gamma-2}\tilde{\mathbf{E}}\mathbf{z} + [\tilde{\mathbf{E}}\mathbf{Z}_k \ \mathbf{D}\mathbf{U}'\hat{\mathbf{S}}]\mathbf{\Lambda}_k = \tau\rho^{\gamma-1}\mathbf{d}.$$

A simple reformulation of the above plus the constraints $\mathbf{z}'[\tilde{\mathbf{E}}\mathbf{Z}_k \ \mathbf{D}\mathbf{U}'\hat{\mathbf{S}}] = \mathbf{0}$ yields the following matrix form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}' & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{\Lambda}_k \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix}, \quad \text{(B.4)}$$

where $\mathbf{A} = \tau^2[\gamma\rho^{\gamma-1}\mathbf{I} + (1-\gamma)\rho^{\gamma-2}\tilde{\mathbf{E}}]$, $\mathbf{B} = [\tilde{\mathbf{E}}\mathbf{Z}_k \ \mathbf{D}\mathbf{U}'\hat{\mathbf{S}}]$, and $\mathbf{q} = \tau\rho^{\gamma-1}\mathbf{d}$. By the standard formula for inverse of a partitioned matrix and the constraint that $\mathbf{z}'\mathbf{z} = 1$, we obtain

$$\mathbf{z} = \frac{\mathbf{M}\mathbf{q}}{\|\mathbf{M}\mathbf{q}\|}, \quad \text{(B.5)}$$

where $\mathbf{M} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{B}'\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}^{-1}$. Note that $\mathbf{M}$ has a factor of τ^{-2} and $\mathbf{q}$ has a factor of τ, hence $\mathbf{M}\mathbf{q}$ has a factor of τ^{-1} that gets canceled out during the normalization in (B.5). Therefore, $\mathbf{z}$ does not rely on the quantity of τ. Without loss of generality, we can choose $\tau = 1$. Then the only unknown parameter is ρ. Hence, we can formulate (B.4) and (B.5) as a fixed point problem of $\mathbf{z}(\rho)'\tilde{\mathbf{E}}\mathbf{z}(\rho)$ as a function of ρ. More specifically, we seek for ρ^* that satisfies $\mathbf{z}(\rho^*)'\tilde{\mathbf{E}}\mathbf{z}(\rho^*) = \rho^*$. Afterwards, we obtain $\mathbf{z}^* = \mathbf{M}^*\mathbf{q}^*/\|\mathbf{M}^*\mathbf{q}^*\|$, where $\mathbf{M}^*$ and $\mathbf{q}^*$ are computed from ρ^*. We summarize the procedure to solve (B.2) as in the following two steps:

Step 1: Solve the fixed point ρ^* for $\rho = \mathbf{z}(\rho)'\tilde{\mathbf{E}}\mathbf{z}(\rho)$ with $\mathbf{z}(\rho) = \mathbf{Mq}/\|\mathbf{Mq}\|$, where

$$\begin{aligned}
\mathbf{M} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{B}'\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}^{-1}, \\
\mathbf{A} &= \gamma\rho^{\gamma-1}\mathbf{I} + (1-\gamma)\rho^{\gamma-2}\tilde{\mathbf{E}}, \\
\mathbf{B} &= [\tilde{\mathbf{E}}\mathbf{Z}_k \ \mathbf{D}\mathbf{U}'\hat{\mathbf{S}}], \\
\mathbf{q} &= \rho^{\gamma-1}\mathbf{d};
\end{aligned}$$

Step 2: Compute $\mathbf{M}^*$ and $\mathbf{q}^*$ from the fixed point ρ^* in Step 2. The solution to (B.2) is then given by $\mathbf{z}^* = \mathbf{M}^*\mathbf{q}^*/\|\mathbf{M}^*\mathbf{q}^*\|$.

The most challenging step is Step 2, where a nonlinear equation needs to be solved. This can be done numerically by several existing algorithms. For example, we can use Newton's method (Kelley 2003), which is implemented by many optimization packages and gives fast convergence in practice.

C JICO iterative algorithm

After each iteration, we obtain K joint objective values as a $K \times 1$ vector:

$$\mathcal{L} = \text{diag}(\mathbf{W}'\mathbf{X}^{\text{Joint}\prime}\mathbf{X}^{\text{Joint}}\mathbf{W})^{\gamma-1}(\mathbf{W}'\mathbf{X}^{\text{Joint}\prime}\boldsymbol{Y}^{\text{Joint}})^2, \tag{C.1}$$

and K_g individual objective values as a $K_g \times 1$ vector for $g = 1, \ldots, G$:

$$\mathcal{L}_g = \text{diag}(\mathbf{W}_g'\mathbf{X}_g^{\text{Indiv}\prime}\mathbf{X}_g^{\text{Indiv}}\mathbf{W}_g)^{\gamma-1}(\mathbf{W}_g'\mathbf{X}_g^{\text{Indiv}\prime}\boldsymbol{Y}_g^{\text{Indiv}})^2, \tag{C.2}$$

and compare them with the corresponding vectors obtained from the previous iteration step. Our iterative procedure stops until the differences between two consecutive iteration steps are under a certain tolerance level. Empirically, the algorithm have always met convergence criteria, albeit there are no theoretical guarantees for it.

Data: $\{\mathbf{X}_g, \boldsymbol{Y}_g\}_{g=1}^G$;
Parameters: tolerance level τ; joint rank K and individual rank K_g;
Initialize $\mathbf{T}_g = \mathbf{0}_{n_g \times K_g}$; $\mathbf{U}_g = \mathbf{0}_{K_g \times p}$; $\boldsymbol{\alpha}_g = \mathbf{0}_{K_g \times 1}$; ;
while *Euclidian distance* $\|\nabla \mathcal{L}\|, \|\nabla \mathcal{L}_g\| > \tau$ **do**
  $\mathbf{X}_g^{\text{Joint}} = \mathbf{X}_g - \mathbf{T}_g \mathbf{U}_g$; $\boldsymbol{Y}_g^{\text{Joint}} = \boldsymbol{Y}_g - \mathbf{T}_g \boldsymbol{\alpha}_g$;
  Estimate $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_K)$ from (3.2);
  $\mathbf{S} = \mathbf{X}^{\text{Joint}} \mathbf{W}$; $\mathbf{U} = (\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}'$; $\boldsymbol{\alpha} = (\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\boldsymbol{Y}^{\text{Joint}}$;
  for $g = 1, \ldots, G$ **do**
    $\mathbf{X}_g^{\text{Indiv}} = \mathbf{X}_g - \mathbf{S}_g \mathbf{U}$; $\boldsymbol{Y}_g^{\text{Indiv}} = \boldsymbol{Y}_g - \mathbf{S}_g \boldsymbol{\alpha}$;
    Estimate $\mathbf{W}_g = (\mathbf{w}_{g1}, \ldots, \mathbf{w}_{gK_g})$ from (3.4);
    $\mathbf{T}_g = \mathbf{X}_g^{\text{Indiv}} \mathbf{W}_g$; $\mathbf{U}_g = (\mathbf{W}_g'\mathbf{W}_g)^{-1}\mathbf{W}_g'$; $\boldsymbol{\alpha}_g = (\mathbf{T}_g'\mathbf{T}_g)^{-1}\mathbf{T}_g'\boldsymbol{Y}_g^{\text{Indiv}}$;
  end
  $\tilde{\mathbf{W}} = [\mathbf{W}_1, \ldots, \mathbf{W}_G]$;
end

Algorithm 1: JICO Algorithm

D Selection of tuning parameters

Based on our numerical experience, we find that JICO's performance is more susceptible to the choice of γ than the ranks. Thus, we propose to select the optimal γ by fine-tuning it in a wide range with a coarse grid search of ranks K and K_g. For rank selection, we find that JICO's performance depends more on the choice of K but less on K_g, especially when we slightly over-estimate the latter. Therefore, we propose to tune ranks by performing an exhaustive search on $K \in \{0, 1, \ldots, D_1\}$ and $K_1 = \cdots = K_G \in \{0, 1, \ldots, D_2\}$, where D_1 and D_2 are two user-defined integers. To save computational costs, K_g's from all groups are kept at the same value. In this way, we need to try $(D_1 + 1) \times (D_2 + 1)$ combinations in total. Since we assume $\mathbf{J}_g$ and $\mathbf{A}_g$ are both low-rank, we can set D_1 and D_2 to be relatively small.

We perform simulation studies to support our previous claims on the parameter tuning. We consider both PCR and PLS settings. For the PCR setting, we generate samples of two groups, using the data model described in Section 4.1 with true ranks $K = 2, K_1 = 3$ and $K_2 = 5$. We implement the JICO method with $\hat{K} = 1, 2, 3$ and three combinations of $(\hat{K}_1, \hat{K}_2) = (3, 3), (5, 5), (3, 5)$, using $\gamma = 1$ and $1e10$ respectively. For the PLS setting, we generate samples of two groups, using the data model described in Section 4.2 with true ranks $K = 1, K_1 = 1$ and $K_2 = 2$. Under this setting, we implement the JICO method

with $\hat{K} = 0, 1, 2$ and three combinations of $(\hat{K}_1, \hat{K}_2) = (1, 1), (1, 2), (2, 2)$, using $\gamma = 1$ and $1e10$ respectively. For both settings, we run the simulations for 50 times and report MSEs of different JICO models in Tables D.1 and D.2 respectively.

$\hat{K}$	γ	$\hat{K}_1 = 3, \hat{K}_2 = 3$	$\hat{K}_1 = 3, \hat{K}_2 = 5$	$\hat{K}_1 = 5, \hat{K}_2 = 5$
1	1	5.1419 (0.7271)	5.1597 (0.7296)	5.1954 (0.7347)
	$1e10$	4.5322 (0.6409)	2.7167 (0.3842)	1.9906 (0.2815)
2	1	8.4345 (1.1928)	8.4890 (1.1961)	8.4890 (1.2005)
	$1e10$	2.1063 (0.2979)	**0.0818** (0.0116)	0.0821 (0.0116)
3	1	10.6540 (1.5067)	10.6703 (1.5090)	10.6929 (1.5122)
	$1e10$	2.7541 (0.3894)	1.5409 (0.2179)	1.4494 (0.2049)

Table D.1: MSEs of JICO model fit with different tuning parameters under the PCR setting. Numbers in brackets are standard errors. The MSE with the correct choice of tuning parameters is in boldface.

$\hat{K}$	γ	$\hat{K}_1 = 1, \hat{K}_2 = 1$	$\hat{K}_1 = 1, \hat{K}_2 = 2$	$\hat{K}_1 = 2, \hat{K}_2 = 2$
0	1	1.0322 (0.0241)	0.9226 (0.0191)	0.8396 (0.0170)
	$1e10$	2.9912 (0.0676)	2.970 (0.066)	2.926 (0.0627)
1	1	0.6594 (0.0185)	**0.6415** (0.0183)	0.6259 (0.0187)
	$1e10$	2.9410 (0.0650)	2.9215 (0.0647)	2.9002 (0.0640)
2	1	0.9770 (0.0280)	0.9753 (0.0275)	0.9761 (0.0275)
	$1e10$	2.9210 (0.0658)	2.8767 (0.0677)	2.8486 (0.0683)

Table D.2: MSEs of JICO model fit with different tuning parameters under the PLS setting. Numbers in brackets are standard errors. The MSE with the correct choice of tuning parameters is in boldface.

First, we observe that the choice of γ greatly impacts the JICO's performance. As can be seen in Table D.1, in the PCR setting, all models with $\gamma = 1$ are worse than those with $\gamma = 1e10$ by a great margin, even when the latter uses wrong choices of K and K_g. Similarly, in the PLS setting, all models with $\gamma = 1e10$ are worse than those with $\gamma = 1$ by a great margin as shown in Table D.2. Hence, we recommend users focusing more on tuning γ, by doing a fine grid search in a relatively large range.

Second, we observe that if we use the good γ (i.e., $\gamma = 1e10$ for the PCR setting and $\gamma = 1$ for the PLS setting), JICO's performance depends more on the choice of K but less on K_g, especially if we slightly over-estimate the latter. In Table D.1, with $\gamma = 1e10$, the MSEs with $\hat{K} = 2$ and different $\hat{K}_g$'s are all smaller than the counterparts with $\hat{K} = 1$

or 3. This suggests that tuning K is more crucial than tuning K_g. On the other hand, when we choose $\hat{K} = 2$, $\hat{K}_1 = 5$ and $\hat{K}_2 = 5$, its performance is not much worse than the choice of $\hat{K} = 2$, $\hat{K}_1 = 3$ and $\hat{K}_2 = 5$. However, if we choose $\hat{K} = 2$, $\hat{K}_1 = 3$ and $\hat{K}_2 = 3$, its performance becomes worse. Therefore, to save the tuning time, we suggest a two-dimensional grid search of K and K_g, by enforcing $K_1 = \cdots = K_G$. From Table D.2, we see that the same conclusions can also be drawn under the PLS setting.

Finally, we point out that the goal of our algorithm is to approximate the joint and individual structure of the data by some low-rank matrices, and use the resulting low-rank matrices for prediction. Therefore, when the individual and joint matrices are too complex and cannot be approximated by some low-rank matrices, we suggest to use alternative methods to solve the problem. In addition, our method relies on proper tuning of γ and ranks. If the differences between the ranks of individual matrices and the joint matrix are large, our algorithm may take a long time to compute.

E The impact of using different initial values of $\mathbf{I}_g$

The following simulation study illustrates the impact of using different matrices as the initial value for our algorithm.

We use the same data generating scheme as described in Section 4. We fix $G = 2, p = 200, n_1 = n_2 = 50$. In each replication, we generate 100 training samples to train the models and evaluate the corresponding Mean Squared Error (MSE) in an independent test set of 100 samples. We also record the objective functions as described below:

- When $\gamma = 0$, we record the individual objective function as defined in (C.2) for $g = 1, 2$. The joint rank is zero in this case.
- When $\gamma = 1$, we record the joint objective function $\text{cov}(\mathbf{X}^{\text{Joint}}\mathbf{w}, \boldsymbol{Y}^{\text{Joint}})^2$ and the individual objective function $\text{cov}(\mathbf{X}_g^{\text{Indiv}}\mathbf{w}, \boldsymbol{Y}_g^{\text{Indiv}})^2$ for $g = 1, 2$.
- When $\gamma = \infty$, we record the joint objective function $\text{var}(\mathbf{X}^{\text{Joint}}\mathbf{w})$ and the individual objective function $\text{var}(\mathbf{X}_g^{\text{Indiv}}\mathbf{w})$ for $g = 1, 2$.

We compare the performance of our algorithm by giving the boxplots of the MSEs and the converged objective values using different initial values. We repeat the simulation for 50 times, with three choices of initial values of the individual matrices $\mathbf{I}_g$ as follows:

- Zero matrix: In this setting, our algorithm starts with zero matrices as the initial value, where all entries in $\mathbf{I}_g$ are equal to 0.
- Large matrix: In this setting, our algorithm starts with matrices with large entries as the initial value, where all entries in $\mathbf{I}_g$ are equal to 3.
- Joint matrix: In this setting, our algorithm starts with the estimated joint matrices $\hat{\mathbf{J}}_g$ as the initial value, where $\hat{\mathbf{J}}_g$ are derived from our first setting, that is using zero matrices as the initial value.

Figures E.2 shows the converged value of the joint objective function under three choices of $\mathbf{I}_g$ mentioned above. Figures E.3 and E.4 show the converged value of the first and the second group's objective function under the three different initial values, respectively. The results show that our algorithm may not converge to the same objective function value using different initial values when $\gamma = 1$, but Figure E.1 shows that our algorithm tends to have similar performance in terms of prediction error, no matter how we choose the initial individual matrices. To ensure better performance, we recommend running our algorithm on the same dataset multiple times with different initial values, and then choose the result with the best performance.

F Convergence and computational time of JICO

In this section, we perform additional numerical studies using the same simulated model as in Appendix E, but with a focus on understanding the algorithm's convergence and computational time.

In the first numerical study, we investigate how fast the objective functions converge with different initial values. The definitions of objective functions can be found in Appendix E. We generate the data under PLS and PCR settings as described in Appendix E and

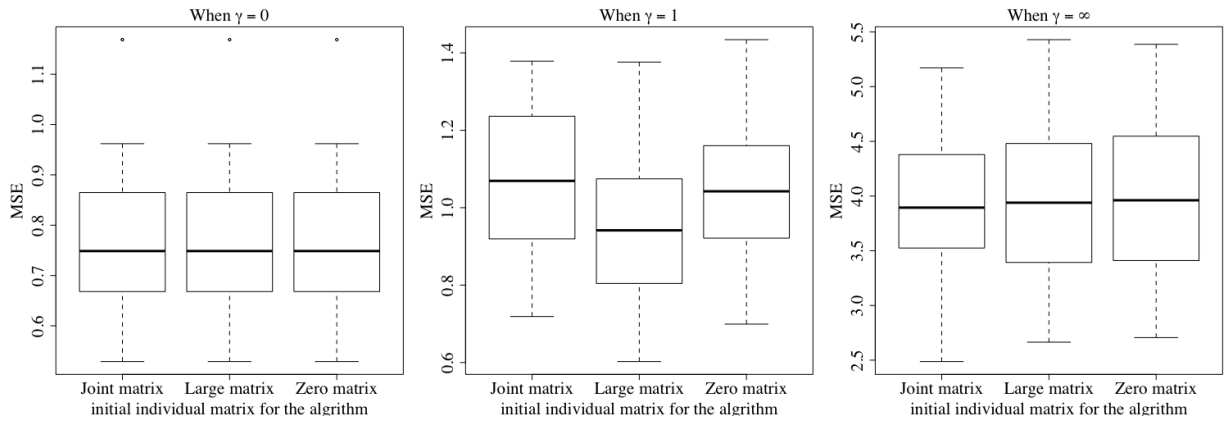


Figure E.1: The impact of using different initial individual matrix on MSE under different settings.

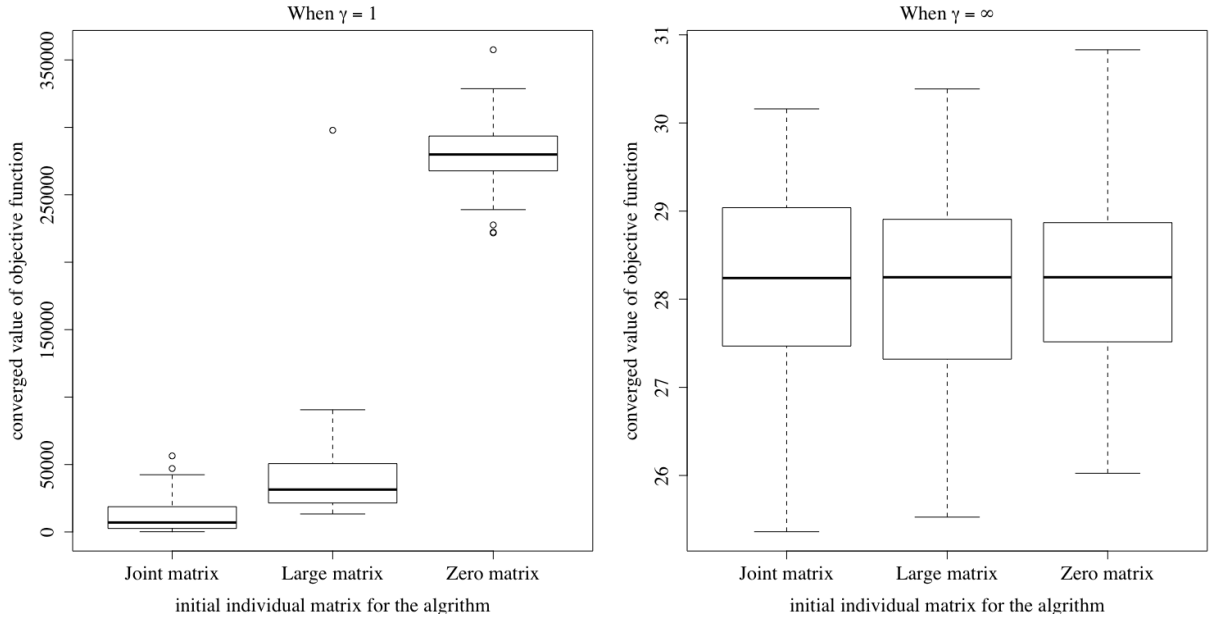


Figure E.2: The impact of using different initial individual matrix on converged value of the joint objective function under different settings.

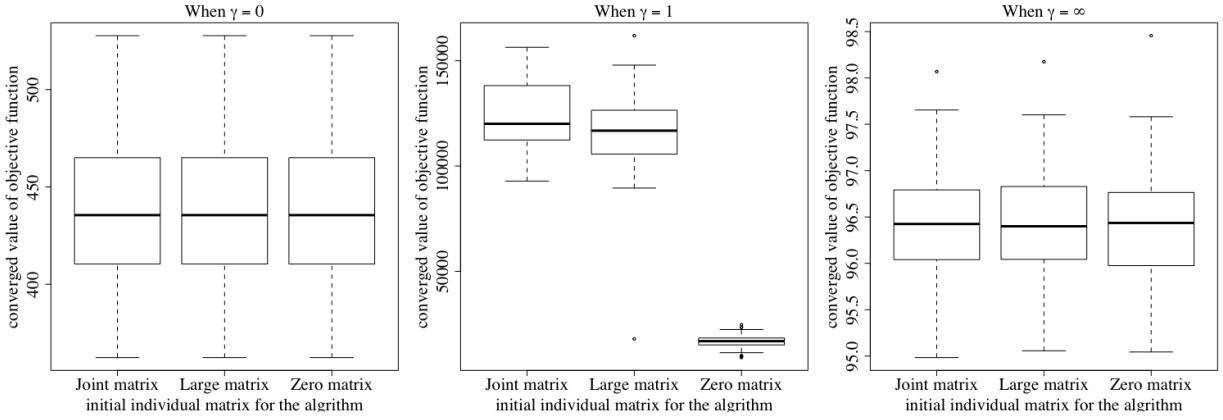


Figure E.3: The impact of using different initial individual matrix on converged value of the first group's individual objective function under different settings.

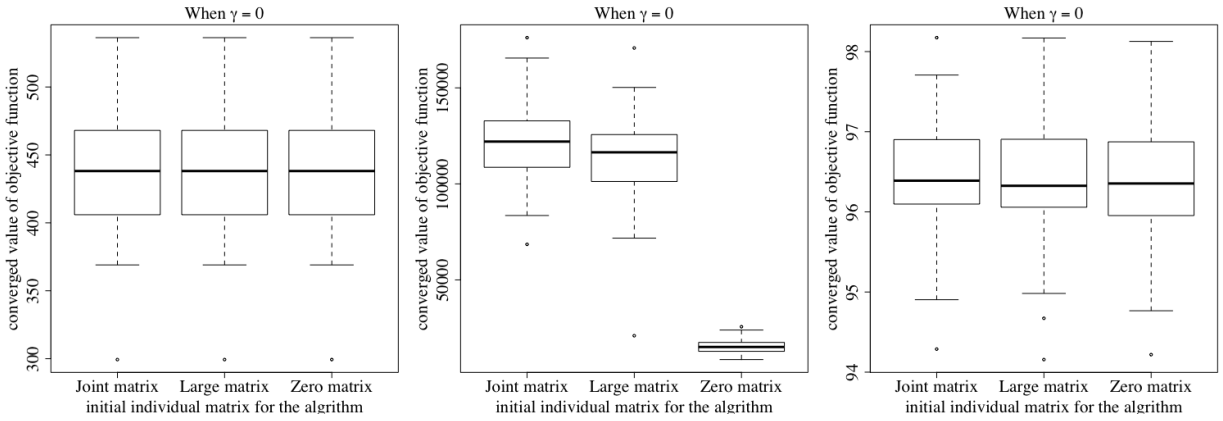


Figure E.4: The impact of using different initial individual matrix on converged value of the second group's individual objective function under different settings.

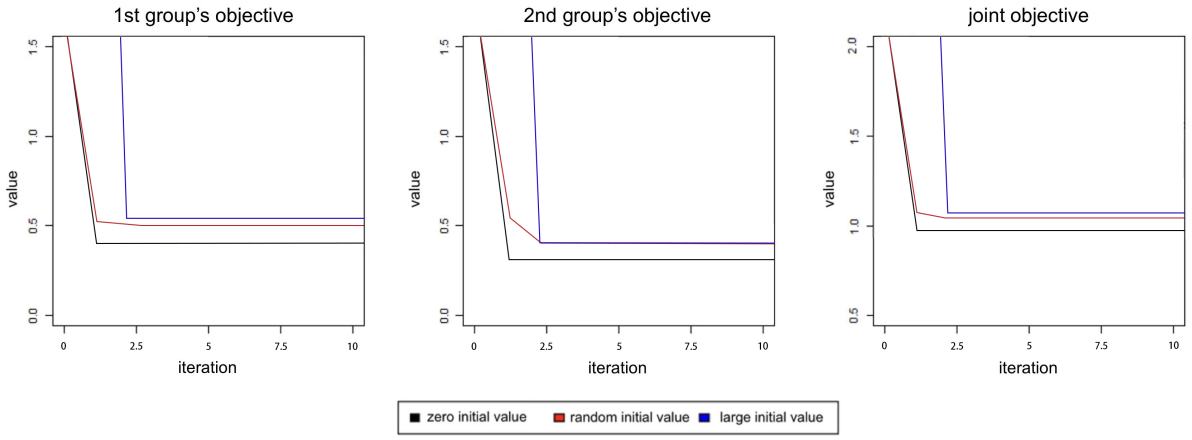


Figure F.1: Numerical convergence of JICO under PCR setting. Models are fit with $\gamma = 1e10$.

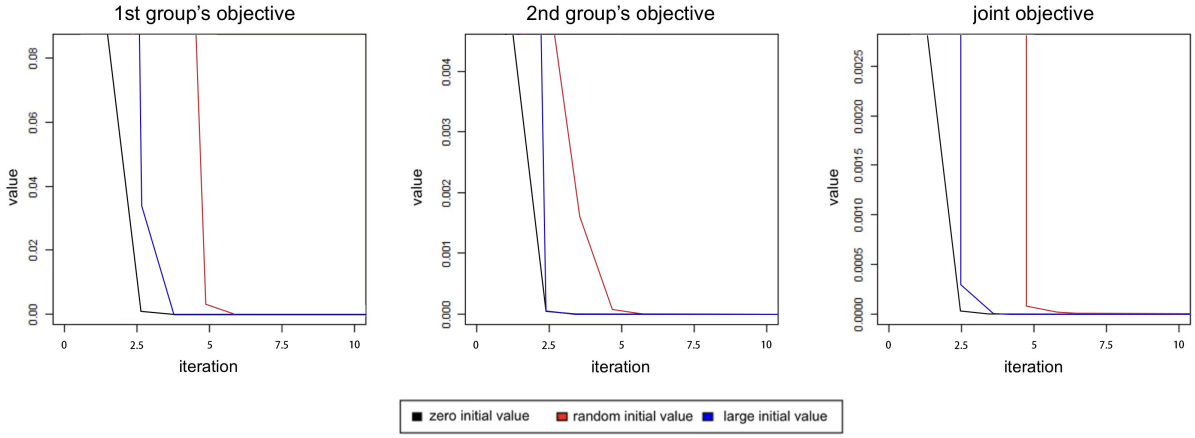
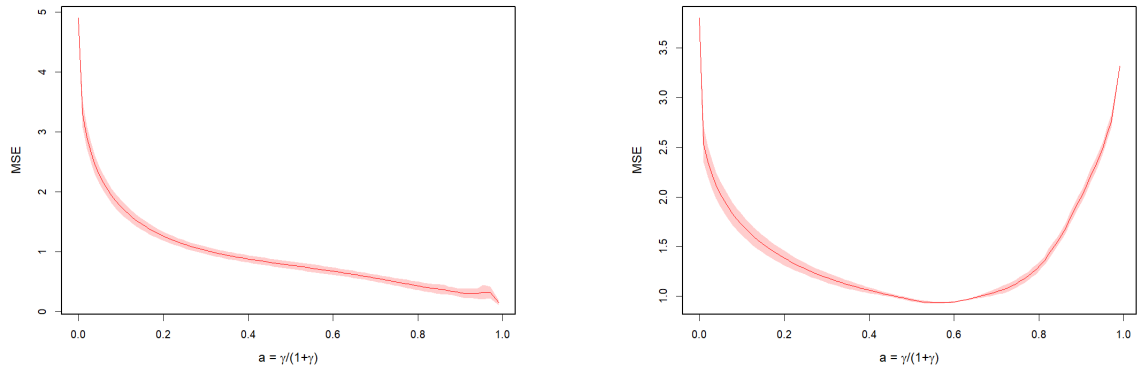


Figure F.2: Numerical convergence of JICO under PLS setting. Models are fit with $\gamma = 1$.

repeat the simulation for 20 times. We fit the JICO models with $\gamma = 1e10$ for the PCR setting and with $\gamma = 1$ for the PLS setting. We consider the same three different initial values of the individual matrix $\mathbf{I}_g$ as in Appendix E.

We plot the objective function values against iterations under different initial values in Figures F.1 (PCR setting) and F.2 (PLS setting). As can be seen in these plots, our algorithm converges quickly in less than 10 iterations in both settings, regardless of the initial choice of $\mathbf{I}_g$.

In the second numerical study, we investigate if our algorithm can give stable predictions when it starts with different initial values. To this end, we repeat our simulations 30 times



(a) PCR Example (b) PLS Example

Figure F.3: MSE curves of JICO with randomly initialized values on $\mathbf{I}_g$ under the PCR setting (a) and PLS setting (b). The red solid curve gives the mean MSE and the shadow gives the standard errors. Here $a = 0.5$ and 1 correspond to PLS and PCR respectively.

using the same data settings as described above. For each time, we run our algorithm with 10 randomly initialized values $\mathbf{I}_g \sim N(0, 1)$ and then choose the best model that gives the smallest MSE. Then, we evaluate the MSE given by the best model on an independent test set. Figures F.3(a) and F.3(b) demonstrate the MSEs evaluated on the test set over these 30 repetitions under (a) PCR setting and (b) PLS setting. We can see that the performance of our algorithm is quite stable, especially when γ approaches ∞ or equals 1 for the PCR and PLS settings, respectively. It shows that JICO can give stable predictions with different initial values of $\mathbf{I}_g$.

Finally, we perform a numerical study to compare the computational time of JICO with OLS, PLS and PCR, which includes the tuning time for finding the optimal γ, K and K_g, as well as the training time with multiple initial values, using the optimal tuning parameters.

In this numerical study, we generate 100 samples in each group and 200 variables using the same PCR setting as described above and we repeat the same process for 100 times. We fit the data with all mentioned methods. We tuned JICO with 10 different γ's and 3 combinations of K and K_g using 3-fold cross-validation. We tune PLS and PCR methods with the same choices of ranks using 3-fold cross-validation. The averaged training and tuning time and their standard errors are summarized in Table F.3. Note that the training

of JICO is not much slower than the three non-iterative methods. The tuning for JICO takes longer as it requires additional tuning of γ, whereas PCR and PLS only need to tune ranks.

	OLS	PLS	PCR	JICO		
				zero matrix	random matrix	large matrix
training time	0.01(0.0001)	0.05(0.0013)	0.04(0.0014)	0.12(0.0130)	0.23(0.0252)	0.13(0.0128)
tuning time	-	0.25(0.0044)	0.14(0.0022)	10.34(0.0214)	101.20(3.2217)	35.06(0.0266)

Table F.3: Average training and tuning time needed for each method. Numbers in parenthesis are standard errors. (unit: second)

In general, we find that the tuning and training time of JICO is still quite acceptable. Compared with OLS, PLS and PCR, the additional computation cost of JICO is not too expensive, yet it is more flexible and gives better numerical performance. We hope that the simulation results and Table F.3 can serve as a general reference for people to be aware of JICO's trade-offs between model accuracy versus computational cost.

G More simulation results

We conduct more simulations to demonstrate the performance of JICO. The simulation settings are the same as described in Section 4 except for the following changes. We now generate the error term $\mathbf{e}_g$ in (4.1) from $\mathcal{N}(0,1)$. We also change α and α_g in (4.1). In the PCR Example, we set $\alpha = 3$ and $\alpha_g = 3$. In the PLS Example, we set $\alpha = 2$ and $\alpha_g = 1$. In the OLS Example (a), we set $\alpha = 3$ and $\alpha_g = 0$. In the OLS Example (b), we set $\alpha = 0$, $\alpha_g = 3$. Table G.4 reports the MSEs of JICO along with other methods. We see that the same conclusion can be drawn as for Table 1. That is, the JICO model with the right choice of γ consistently outperforms other methods in all examples. Since the signal-to-noise ratios in the new settings are lower than the counterparts in the Section 4, the MSEs all enlarge, which is reasonable.

	Method		$g = 1$	$g = 2$	Overall
(a) PCR Example	JICO	$\gamma = 0$	3.226(0.504)	2.156(0.398)	5.382(0.642)
		$\gamma = 1$	3.702(0.650)	2.260(0.385)	5.962(0.755)
		$\gamma = \infty$	**1.358**(0.286)	**1.179**(0.271)	**2.537**(0.394)
	Global	Ridge	7.622(1.461)	8.047(2.160)	15.669(2.608)
		PLS	11.340(2.185)	12.146(2.713)	23.486(3.484)
		PCR	5.241(1.073)	9.831(1.979)	15.072(2.251)
	Group-specific	Ridge	5.520(1.048)	5.710(1.083)	11.229(1.507)
		PLS	3.310(0.502)	1.906(0.333)	5.216(0.602)
		PCR	6.708(1.163)	6.158(0.989)	12.866(1.527)
(b) PLS Example	JICO	$\gamma = 0$	3.127(0.595)	3.041(0.476)	6.167(0.762)
		$\gamma = 1$	**2.209**(0.350)	**2.174**(0.366)	**4.383**(0.506)
		$\gamma = \infty$	6.848(1.307)	7.043(1.252)	13.892(1.810)
	Global	Ridge	3.815(0.710)	3.412(0.549)	7.226(0.898)
		PLS	2.779(0.494)	2.592(0.467)	5.371(0.680)
		PCR	6.870(1.356)	6.770(1.254)	13.640(1.847)
	Group-specific	Ridge	3.915(0.765)	3.901(0.670)	7.816(1.017)
		PLS	2.854(0.551)	2.788(0.478)	5.641(0.730)
		PCR	6.848(1.307)	7.044(1.252)	13.892(1.810)
(c) OLS Example (a)	JICO	$\gamma = 0$	**3.413**(0.589)	**2.008**(0.360)	**5.421**(0.690)
		$\gamma = 1$	4.247(0.652)	3.033(0.590)	7.281(0.879)
		$\gamma = \infty$	13.586(2.676)	11.757(2.400)	25.344(3.595)
	Global	Ridge	8.072(1.500)	6.818(1.366)	14.890(2.029)
		PLS	6.459(0.652)	4.727(0.590)	11.186(0.879)
		PCR	13.550(2.718)	12.780(2.307)	26.330(3.565)
	Group-specific	Ridge	9.836(1.978)	9.990(1.956)	19.826(2.782)
		PLS	9.680(1.507)	9.274(1.757)	18.954(2.315)
		PCR	13.638(2.713)	13.102(2.545)	26.740(3.720)
(d) OLS Example (b)	JICO	$\gamma = 0$	**1.755**(0.310)	**1.542**(0.303)	**3.297**(0.433)
		$\gamma = 1$	2.026(0.444)	2.014(0.347)	4.040(0.564)
		$\gamma = \infty$	10.850(1.902)	8.231(1.598)	19.081(2.485)
	Global	Ridge	7.190(1.355)	6.478(1.046)	13.668(1.711)
		PLS	7.509(1.821)	7.424(2.004)	14.934(2.708)
		PCR	11.194(2.047)	8.117(1.528)	19.310(2.554)
	Group-specific	Ridge	4.870(0.945)	4.030(0.789)	8.901(1.232)
		PLS	3.518(0.444)	3.450(0.347)	6.968(0.564)
		PCR	11.071(1.903)	8.231(1.594)	19.302(2.482)

Table G.4: Groupwise and overall MSEs under the PCR, PLS and OLS settings. Numbers in brackets are standard errors.

References

Feng, Q., Jiang, M., Hannig, J. & Marron, J. (2018), 'Angle-based joint and individual variation explained', *Journal of Multivariate Analysis* **166**, 241–265.

Kelley, C. T. (2003), *Solving nonlinear equations with Newton's method*, Vol. 1, Siam.