R2 & R3 - Why use ReLU: We use ReLU ($f(x) = \max(0, x)$) to enforce $\mathbf{P}$ positive. As the solution to integer linear programming (ILP) problem in Eq.5 is non-differentiable and NP-complete, we relax this constraint to Eq.6 through ReLU. After that, we adopt the Dykstra's projection algorithm to compute the intersection of convex sets by iteratively projecting $\mathbf{P}$ onto each of the convex sets, i.e., 1) $P_{ij} \geq 0$; 2) $\mathbf{P1} = \mathbf{1}$; 3) $\mathbf{P}^\top \mathbf{1} = \mathbf{1}$. In summary, we adopt ReLU to meet the first constraint, and the other functions will be alternative as long as it could make $\mathbf{P}$ positive.

R1 & R2 & R3 - Alignment module: The proposed module aligns the data with the computation complexity $\mathcal{O}(\tau_1 \tau_2 n^2)$ (n is the batch size for training), and allows the network to utilize the available correspondence information from partially aligned data in an end-to-end manner, as shown in Eq.4. We use Adam optimizer for network training. The alignment module recurrently computes the permutation matrix, but it is not the RNN architecture. The proposed module is pluggable for any neural network such as DCCA/DCCAE by computing the pairwise distance on the learned representations and achieving the alignment by the proposed module.

R1 & R2 & R3 & R4 - Convergence analysis: The convergence of the whole model could refer to Fig3.c. One could observe that the loss decreases a lot in the first 600 epochs, then continuously and smoothly decreases until convergence. As for the alignment module, we experimentally find that it converges fast with ($\tau_1 = 30, \tau_2 = 10$). The influence of the alignment module to the whole network is predictable, since PVC jointly learns the common representation and aligns the data to address the challenging PVP, it inevitably converges slower than representation learning only.

R2 & R4 - Results compared to fully aligned: The possible reason is two-fold. First, Reuters is a document database that consists of English documents and its machine-translated versions. Thus there may be some noise/incorrect pairs in the fully-aligned dataset which could be addressed by our alignment module. Second, the re-aligned multi-view data may improve data consistency and meet the intrinsic data distribution, thus boosting the clustering performance.

R1 - Literature review: 1) Multi-modality also faces the partially view-aligned problem (PVP) as different modality may be collected in the wrong order due to temporal and spatial complexity. 2) In the paper, L85-95 have indicated that only a few works try to alleviate the effect caused by PVP. The major reasons for hindering studies on PVP have been stated in L107-117. In short, the traditional shallow methods usually pre-align the data in the preprocessing phrase and then perform clustering on the re-aligned data with a two-stage paradigm, which is to avoid directly solving PVP. In other words, PVP is ignored in the traditional shallow setting which does not benefit from end-to-end optimization. Moreover, the non-differentiable alignment algorithms adopted by these methods hinder them extend to deep models, while the proposed differentiable alignment module is pluggable to multi-view models to address PVP and embrace the attributes of deep models. 3) The difference between PVC and the existing works is two-fold. First, the methods are shallow models and there are no efforts devoted to developing effective deep solutions so far as we knew, while PVC proposes the differentiable alignment module to facilitate the deep approach. Second, these works establish the correspondence of views in a separate step, while PVC jointly learns the common representations and aligns the data.

R1 - Network setting, memory cost, and parameter complexity: We provide the configuration and implementation details of PVC in the supplementary material. For the memory cost and parameter complexity, we conduct experiments on Caltech101-20 compared to AE2-Nets. For training, PVC occupies 1126 MiB GPU memory and needs about 1.02 hours to convergence, while AE2-Nets occupies 362 MiB and needs 0.62 hours to convergence. The reason why PVC needs more memory and computation cost is the additional memory and computation cost caused by the alignment module, which is to address the challenging PVP. For testing, Table 5 (Supplementary Materials) gives a comparison of the Hungarian and the alignment module. It shows that the proposed alignment module (0.09s) is much faster than the Hungarian (17.78s), which means PVC is more capable of practical applications when the model is well-trained.

R1 - Clustering characteristics: As presented at L20-L22, most existing multi-view clustering approaches jointly learn a common representation to bridge the gap among different views and then achieve clustering on the common representation. In other words, learning the common representation is the key problem for multi-view clustering. Similarly, PVC jointly learns the common representation while enforcing the cross-view consistency with the help of the re-aligned data by the differentiable alignment module.

R4 - When U larger than A: Fig.5 may be helpful to address this concern. The figure shows that our method achieves a promising result (ACC: 0.4517, NMI: 0.2231) when U (=0.8) is remarkably larger than A (=0.2), showing the superiority of PVC even there are more unaligned data than aligned ones.

R4 - Multiple views: Our model could easily extend to multiple views by selecting one view as the anchor, and align the other views to establish the correspondence with the corresponding permutation matrix.

R4 - Fluctuant curve: From Fig.3.c, one could see that PVC loss decreases a lot in the first 600 epochs, then continuously and smoothly decreases until convergence. As for ACC and NMI, they both increase roughly as the epoch increase with the fluctuant curve. The possible reason for the fluctuant curve is that the re-aligned data may contain noise/incorrect correspondence, thus leading to unstable ACC and NMI.