Multimodal Learning with Incomplete Modalities by Knowledge Distillation

Qi Wang[1], Liang Zhan[2,3], Paul Thompson[4], Jiayu Zhou[1]

1. Computer Science and Engineering, Michigan State University, East Lansing, MI
2. Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA
3. Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA
4. Imaging Genetics Center, University of Southern California, Marina del Rey, CA

{wangqi19,jiayuz}@msu.edu; liang.zhan@pitt.edu; pthomp@usc.edu

ABSTRACT

Multimodal learning aims at utilizing information from a variety of data modalities to improve the generalization performance. One common approach is to seek the common information that is shared among different modalities for learning, whereas we can also fuse the supplementary information to leverage modality-specific information. Though the supplementary information is often desired, most existing multimodal approaches can only learn from samples with complete modalities, which wastes a considerable amount of data collected. Otherwise, model-based imputation needs to be used to complete the missing values and yet may introduce undesired noise, especially when the sample size is limited. In this paper, we proposed a framework based on knowledge distillation, utilizing the supplementary information from all modalities, and avoiding imputation and noise associated with it. Specifically, we first train models on each modality independently using all the available data. Then the trained models are used as teachers to teach the student model, which is trained with the samples having complete modalities. We demonstrate the effectiveness of the proposed method in extensive empirical studies on both synthetic datasets and real-world datasets.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Computing methodologies** → *Machine learning*;

KEYWORDS

Multimodal Learning, Knowledge Distillation, Incomplete Modalities

ACM Reference Format:

Qi Wang[1], Liang Zhan[2,3], Paul Thompson[4], Jiayu Zhou[1]. 2020. Multimodal Learning with Incomplete Modalities by Knowledge Distillation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403234

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403234

1 INTRODUCTION

Multimodal learning [1, 12, 30–32] has gained increasing attention in recent years as the heterogeneous data become ubiquitous. Modalities are defined as sets of heterogeneous features that are collected from diverse domains or extracted from various feature extractors [34]. The sets of features describe the same subjects and provide both shared and supplementary information of the subjects. Multimodal learning is to integrate predictive information from different modalities to enhance the performance of the learned models. Since different modalities are extracted from different domains or feature extractors, the representations of the modalities may be very distinct from each other. For example, when analyzing posts from social media, the images are typically associated with some text descriptions. During the learning, the text descriptions can be represented by one-hot vectors while the images are naturally represented as continuous-valued matrices. Thus, directly concatenating the modalities and using the typical learning algorithm to build models on the concatenated modalities may fail.

There are many multimodal algorithms developed in the past decade. CCA-based methods [1, 10] maximize the canonical correlation between two modalities to find the common structure of them, i.e., the shared information. Matrix factorization based subspace learning methods [35, 36] project all the modalities to the modality-invariant latent space and the learned modality-invariant representation is used in various prediction tasks. The success of such methods relies on the assumption that each modality provides enough information of the subjects but with different noise or irrelevant information. Therefore, the methods remove the noise of each modality by learning the shared part of different modalities.

Despite utilizing the shared information of multiple modalities, another motivation to use multimodal learning is that multiple modalities provide supplementary descriptions of the same subjects and correlating them in the learning can be more informative. When utilizing all the information from different modalities, the performance is expected to be improved compared to learning with the information from only one modality. During the past years, multiple methods have been proposed to combine the supplementary information. For example, kernel-based algorithms use the multiple kernel methods to combine the kernels of different modalities from linear combination methods such as linear convex combination [26] to nonlinear combination methods [27]. With the development of deep learning, multiple neural networks [11, 19] are used to extracted abstract feature representations for each modality. Then, the extracted representations from all modalities

are fused in different ways such as concatenation to combine the supplementary information.

Although the aforementioned methods work very well with utilizing supplementary information of multiple modalities, one common drawback of the methods is that they usually can only use samples that have complete modalities, and in practice there are very few samples of such kind, especially when considering a large number of modalities. For example, when studying neurodegenerative disease, only partial subjects have the diffusion-weighted MRI while only another part of subjects has genetic data available. The aforementioned methods may have to discard a large portion of data collected through huge efforts. There are two types of approaches one can use when dealing with the data with incomplete modalities. The first one is the subspace learning. However, as mentioned before, subspace learning methods only take the shared information of modalities into account. The second one is to impute the missing modalities. After imputation, standard multimodal learning methods can be used to combine the supplementary information. The incompleteness of modalities leads to block missing of features. Therefore, classical matrix completion methods such as matrix factorization [37] and etc. can not be used to impute the missing modalities. Some advanced imputation methods such as cascaded residual autoencoder [25] and adversarial training [2, 15, 23, 28], which have similar structure as GAN, have been proposed to deal with the modality missing problem. These solutions, however, may introduce unwanted imputation noise when imputing the missing modalities [4]. Especially when the size of samples having complete modalities is small, the modalities imputed by such methods may have a negative effect on the performance of the following tasks [4].

In this paper, we proposed a new multimodal learning framework to integrate the supplementary information of multiple modalities. Our method utilizes all the samples include the ones with incomplete modalities. The proposed method is based on knowledge distillation [8]. We first train models for each modality separately with all the data available. Then, we treat the trained models as teachers to teach a student model. The student model is a multimodal learning model which fuses the supplementary information from multiple modalities. It is trained with the soft labels labeled by the teacher models and the true one-hot label. Since the teacher models are trained with each modality separately, the sample size is much larger than the samples used to train the student model. With enough data, the well-trained teachers act as experts on each modality. The student then learns from these experts and combine the knowledge from all the experts. Compared with existing methods, our method does not discard the samples with incomplete modalities nor impute them. Instead, we use these samples to train the teacher models to make sure the teacher models are experts. To verify the effectiveness of our method, we demonstrate experiments on synthetic data and real-world data include three benchmark datasets.

2 RELATED WORK

Multimodal learning. There are many multimodal learning methods that have been developed in the past years. Canonical correlation analysis (CCA) [10] maximizes the canonical correlations between two modalities to find their common structure, i.e., the shared information across the modalities. Later, DCCA [1] is proposed to deal with nonlinear relationships between modalities.

Subspace learning [35] methods are also widely used in learning multiple modalities. Subspace learning algorithms utilize matrix factorization to factorize the modalities into a modality-invariant part and modal-specific parts. The modality-invariant part is then used to build predictive models. To deal with the nonlinear parts, deep version of subspace learning [30] was also proposed.

To fuse supplementary information, a popular method is to use deep neural networks to extract abstract representations from each modality and then fuse the representations in various ways. For example, Mehrizi et al. [13] fuses different representations by simple concatenation. Song et al. [22] proposed to use element-wise and weighted element-wise multiplication to fuse modalities.

To solve the missing modality problem, some imputation methods have been proposed. Shao et al. [20] proposed to impute the kernel matrix for the missing modality using the kernels of other modalities. In [25], the authors first concatenated all the modalities to form a large matrix. Then, they applied cascaded residual autoencoder to impute the missing elements. Cai et al. [2] uses adversarial learning to complete the missing modalities.

As mentioned in Sec. 1, most of the methods have limitations when fusing supplementary information of samples with missing modalities. Therefore, in this paper, we propose a new framework to tackle the limitations.

Knowledge distillation. Knowledge distillation transfers model information from one teacher model to a student model by "dark knowledge" from the teacher model. Hinton et al. [8] proposed to use the teacher to label samples with soft labels and then let the student mimic the soft label. Zhang et al. [39] trained two student networks simultaneously, during which the students teach each other. The performance of both the two student networks improves.

In this paper, we use knowledge distillation to transfer knowledge from teacher models which are trained with all the samples including that have missing modalities, to the student model which is trained only with the complete modalities. The student model's performance is greatly enhanced by the teaching process.

3 METHODOLOGY

In this section, we first give a brief introduction to knowledge distillation [8]. Then, we introduce our method which leverages knowledge distillation to conduct multimodal learning with supplementary information.

3.1 Knowledge Distillation

Knowledge distillation is used to transfer "dark knowledge" from a teacher to a student. To transfer knowledge, the teacher is first trained on a dataset. Denote the trained teacher model as $Te(\phi)$ with ϕ denotes the parameters of the teacher model. Then, the student is trained to mimic the output of the teacher on the training dataset. Given a dataset $D = \{\{X_1, y_1\}, \{X_2, y_2\} \dots \{X_N, y_N\}\}$ used to train the student, the teacher is first applied on the data and label the data with the logits. We assume there are in total C classes, and the labels are thus given by:

$$z_i = Te(X_i; \phi), \tag{1}$$

where $z_i \in \mathbb{R}^{C\times 1}$ is the logits labeled by the teacher model for sample X_i. The student model is then trained with both the true one-hot label $\{y_i, y_2, \ldots, y_N\}$ and the logits $\{z_1, z_2, \ldots, z_N\}$. Suppose the student model is a deep neural network $f(\theta)$ parameterized by θ. It takes X_i as input and outputs a $C \times 1$ vector which is the logit vector. Then, a SoftMax function is added to the logit vector to output the probability of X_i to be classified as C classes. The loss function of training the student network is:

$$\min_{\theta} l = \sum_{i}^{N} l_c(X_i, y_i; \theta) + l_d(X_i, z_i; \theta). \tag{2}$$

where l_c is a classification loss with the true one-hot label with the form:

$$l_c(X_i, y_i; \theta) = H(\sigma(f(X_i; \theta)), y_i), \tag{3}$$

where H is the negative cross-entropy loss, and $\sigma(x) : \mathbb{R}^C \to \mathbb{R}^C$ is the SoftMax function:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{C} e^{x_k}} \quad \text{for } i = 1, 2, \ldots, C. \tag{4}$$

$l_d(X_i, z_i; \theta)$ is the distillation loss. Examples of the distillation loss include negative cross-entropy loss or KL-divergence. Without loss of generality, we adopt KL-divergence as the distillation loss:

$$l_d(X_i, z_i; \theta) = D_{KL}(\sigma_T(f(X_i; \theta); T), \sigma_T(z_i; T)). \tag{5}$$

where $\sigma_T(x; T)$ denotes the SoftMax with temperature T:

$$\sigma_T(x; T)_j = \frac{e^{\frac{x_j}{T}}}{\sum_{k=1}^{C} e^{\frac{x_k}{T}}}. \tag{6}$$

With temperature T, the output probability is rescaled and smoothed. If temperature T is large, the probability will be more smooth compared with a small temperate T. The output of $\sigma_T(z_i; T)$ is called the "soft label", which is labeled by the teacher model on the sample X_i. It is believe that the "soft labels" contain more information than the one-hot label [8].

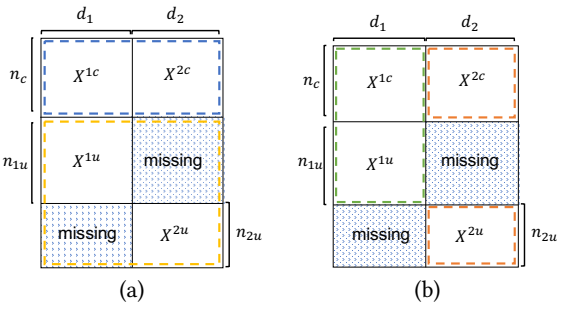


Figure 1: (a) shows the structure of a dataset with two modalities. Samples in the blue dashed-line box have complete modalities and samples in the yellow dashed-line box only have one modality available. (b) Illustration of samples used to train teacher models for two-modal learning. Samples in the green dashed-line box are used to train the first teacher and samples in the orange dashed-line box are used to train the second teacher.

3.2 Multimodal learning with missing modalities

For multimodal learning, it is rather common that some samples do not have complete modalities. Below we first start our discussions on two modalities and then generalize our method to multiple modalities.

Given two modalities $\{X^1 \in \mathbb{R}^{n_1\times d_1}, X^2 \in \mathbb{R}^{n_2\times d_2}\}$ with labels, we denote the samples have complete modalities as $\{X^{1c} \in \mathbb{R}^{n_c\times d_1}, X^{2c} \in \mathbb{R}^{n_c\times d_2}, y^c \in \mathbb{R}^{n_c}\}$. Samples only have the first modality are denoted as $\{X^{1u} \in \mathbb{R}^{n_{1u}\times d_1}, y^{1u} \in \mathbb{R}^{n_{1u}}\}$ and samples only have the second modality are denoted as $\{X^{2u} \in \mathbb{R}^{n_{2u}\times d_2}, y^{2u} \in \mathbb{R}^{n_{2u}}\}$ with $n_1 = n_c + n_{1u}$ and $n_2 = n_c + n_{2u}$. In Figure 1, (a) shows the structure of the data. Samples in the blue dashed-line box are these with complete modalities and samples in yellow dashed-line box only have one modality available. To utilize all the samples, we first train two single modal models with all the available data including the samples with missing modalities. These two models are then acting as teacher models in our framework. We assume that the two teachers are two neural networks $g_1(\phi_1)$ and $g_2(\phi_2)$ with parameters ϕ_1 and ϕ_2. $g_1(\phi_1)$ takes the samples from $[X^{1c}, X^{1u}]$ as input and outputs the logits and $g_1(\phi_1)$ takes the samples from $[X^{2c}, X^{2u}]$ as input and output the logits. The two teachers are trained by minimizing the following loss functions:

$$\begin{aligned} Te_1(\phi_1) &= \min_{\phi_1} \sum\nolimits_{i}^{n_1} H(\sigma(g_1(X_i^1; \phi_1)), y_i), \\ Te_2(\phi_2) &= \min_{\phi_2} \sum\nolimits_{i}^{n_2} H(\sigma(g_2(X_i^2; \phi_2)), y_i) \end{aligned} \tag{7}$$

Then, we use the two teachers to label the samples in $\{X^{1c}, X^{2c}\}$. The logits for the i-th sample are:

$$z_i^1 = Te_1(X_i^{1c}; \phi_1), \quad z_i^2 = Te_2(X_i^{2c}; \phi_2), \tag{8}$$

where z_i^j denotes the logit labeled by teacher j for the i-th sample.

In order to fuse the supplementary information from different modalities, we train a student model with multimodal DNN (M-DNN) [18]. The M-DNN for two modalities contains two branches. Each branch takes one modality as input and is followed with several nonlinear fully-connected layers. The outputs of all the branches are concatenated to form a joint representation. Then, the joint representation is connected to a linear layer to output the logits z. The reason we use such a model as the student model is that the joint representation learned with this model contains the supplementary information of the two modalities. If we train the M-DNN as the methods in [33], i.e., only use the samples with complete modalities $\{X^{1c}, X^{2c}, y^c\}$ to train the model, the sample size is limited to be n_c. If n_c is very small compared with n_1 and n_2, a large amount of useful information is discarded and the samples for training the model is not enough. Thus, we propose to train the M-DNN with the information from the two teachers $Te_1(\phi_1)$ and $Te_2(\phi_2)$ to improve the performance as the two teachers are trained on much larger datasets. The final classification performance for each teacher might be not good enough since each teacher only has access to one modality. But the teachers can do the best to learn classifier with these modalities, provide the expertise for these modalities and teach the student with this knowledge. Denote the student

Algorithm 1: ALgorithm of the proposed method for two modalities.

Inputs: X^{1c}. X^{2c}, y^c, X^{1u}, y^{1u}, X^{2u}, y^{2u}, α, β, T
/*Train teacher models*/
for number of training iterations **do**
  Train teacher model Te_1 with $\{[X^{1c}, X^{1u}], [y^c, y^{1u}]\}$
end for
for number of training iterations **do**
  Train teacher model Te_2 with $\{[X^{2c}, X^{2u}], [y^c, y^{2u}]\}$
end for
Label the samples to train student model with the teachers with Eq. (8).
for number of training iterations **do**
  Train student model with Eq. (9)
end for

network as $f(\theta)$ with θ representing the parameters. The loss function for the proposed method is:

$$\min_\theta l = \min_\theta \sum_i^{n_c} l_c(X_i^1, X_i^2, y_i; \theta) + \alpha l_{d1}(X_i^1, X_i^2, y_i; \theta, Te_1(\phi_1)) + \beta l_{d2}(X_i^1, X_i^2, y_i; \theta, Te_2(\phi_2)), \quad (9)$$

where $l_c(X_i^1, X_i^2, y_i; \theta)$ is the classification loss as

$$l_c(X_i^1, X_i^2, y_i; \theta) = H(\sigma(f(X_i^1, X_i^2; \theta)), y_i).$$

l_{d1}, l_{d2} are distillation loss, α and β are two tunable parameters to control how much knowledge the student model needs from the teacher models. If the parameter is large, it means the student model needs more knowledge from this teacher than a small regularization parameter. The formulations of l_{d1} and l_{d2} are:

$$l_{d1}(X_i^1, X_i^2, y_i; \theta, Te_1(\phi_1)) = D_{KL}(\sigma_T(f(X_i^1, X_i^2; \theta)), \sigma_T(z_i^1)), \quad (10)$$

$$l_{d2}(X_i^1, X_i^2, y_i; \theta, Te_2(\phi_2)) = D_{KL}(\sigma_T(f(X_i^1, X_i^2; \theta)), \sigma_T(z_i^2)). \quad (11)$$

Figure 2 overviews of the proposed framework, and we summarize the training procedure in Algorithm 1.

We would like to highlight the difference between the proposed method with two similar and intuitive methods. The first one is *late fusion*, i.e., fusion at the decision level, which directly combines the labels/logits labeled by the teacher models as the final prediction. Since the teachers only have partial knowledge of the data, the data labeled by the teachers may not be perfect. Researches have shown for most cases late fusion performs worse than *early fusion*, i.e., feature level fusion [6, 21]. In our proposed method, we not only utilize the labels from the teachers, but also perform early fusion with the M-DNN. So, the performance is expected better than late fusion. Another method is to use the teachers as feature extractors to extract abstract features and then use these abstract features as new sets of features to replace the original inputs to train a multimodal model. The performance of this method may perform well when different modalities only have common or shared information and modality-specific noises. However, when different modalities contain supplementary information, the abstract features extracted by each teacher models may have already lost some useful

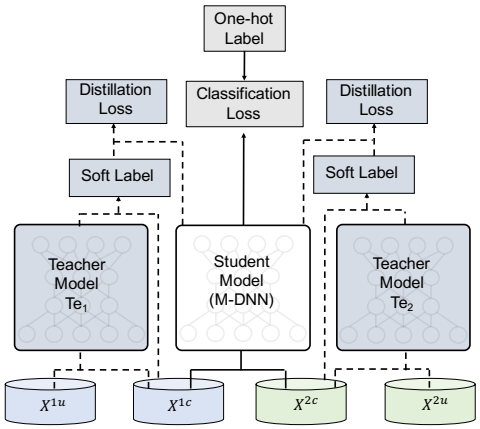


Figure 2: Overview of the proposed model. We first train teacher models with all the available data including the samples have missing modalities. Then, we use the soft labels labeled by the teacher models along with the one-hot label to train the student model.

information as the teacher models are trained on only one modality and are biased. Therefore, its performance is likely to be worse than the proposed method. We will show the performance of these methods in the experiment session.

Mechanism of the proposed method: The underlying mechanism of the proposed approach can be illustrated using gradient analysis. The gradient of the classification loss with respect to the output probability of the k-th class is:

$$\frac{\partial l_c}{\partial p_k} = \sum_i^N (p_{ik} - y_{ik}),$$

where y_{ik} denote the one-hot label of sample i for class k, p_{ik} denote the output probability of sample i for class k. Let L_d denote all the distillation losses, the gradient of the distillation losses with respect to the output probability p_k is:

$$\begin{aligned}\frac{\partial L_d}{\partial p_k} &= \frac{\partial}{\partial p_k}(\alpha \sum_i^N D_{KL}(\sigma_T(z_i), \sigma_T(z_i^1)) \\ &\quad + \beta \sum_i^N D_{KL}(\sigma_T(z_i), \sigma_T(z_i^2))) \\ &= \alpha \sum_i^N (\log p_{ik} - \log q_{ik}^1) + \beta \sum_i^N (\log p_{ik} - \log q_{ik}^2) \quad (12) \\ &\approx \alpha \sum_i^N (p_{ik} - q_{ik}^1) + \beta \sum_i^N (p_{ik} - q_{ik}^2), \quad (13)\end{aligned}$$

where q_{ik}^m is the soft label produced by teacher m for sample i at class k with $m = 1, 2$. We use $log(1 + x) \approx x$ to get (13) from (12). The gradient of the total loss with respect to p_k is:

$$\begin{aligned}\frac{\partial l}{\partial p_k} &= \sum_i^N ((p_{ik} - y_{ik}) + \alpha(p_{ik} - q_{ik}^1) + \beta(p_{ik} - q_{ik}^2)) \\ &= \sum_i^N (1 + \alpha \frac{p_{ik} - q_{ik}^1}{p_{ik} - y_{ik}} + \beta \frac{p_{ik} - q_{ik}^1}{p_{ik} - y_{ik}})(p_{ik} - y_{ik}) \quad (14) \\ &= \sum_i^N w_{ik}(p_{ik} - y_{ik}), \quad (15)\end{aligned}$$

where $w_{ik} = (1+\alpha(p_{ik}-q^1_{ik})/(p_{ik}-y_{ik})+\beta(p_{ik}-q^1_{ik})(p_{ik}-y_{ik}))$. Eq. (15) indicates the samples are reweighted by w_{ik}. w_{ik} is determined by the soft labels and the confidence of the soft labels. If both teachers labeled the sample correctly and the confidence p_{ik} for the correct label is high, the weight w_{ik} is around $(1+\alpha+\beta)$ for this sample. If only one teacher labeled the sample correctly and the confidence is high, the weight is $(1+\alpha)$ or $(1+\beta)$, which is smaller than the samples that are correctly labeled by both teachers with high confidence. If the teachers both make mistakes or if they labeled correctly but with very low confidence, the weight is lower than the aforementioned two cases. So, the proposed method reweights the samples with the teachers' labels and the confidence of the teachers and assign higher weights for the samples that are correctly labeled by the teachers with high confidence.

Generalize to multiple modalities: Given m modalities $X^1 \in \mathbb{R}^{n_1\times d_1}, X^2 \in \mathbb{R}^{n_2\times d_2}, ...X^m \in \mathbb{R}^{n_m\times d_m}$, the dataset could be divided into n parts: (1) samples with complete modalities $X^{ic} \in \mathbb{R}^{n_c\times d_i}$ with $i = \{1, 2, ...m\}$; (2) samples with one modality available $X^{iu} \in \mathbb{R}^{n_{ui}\times d_i}$ with $i = \{1, 2, ..., m\}$; (3) samples with two modalities available $X^{ku\{ij\}} \in \mathbb{R}^{n_{u\{ij\}}\times d_k}$ with $i, j = \{1, 2, ...n\}$ and $k = \{i, j\}$. $X^{ku\{ij\}}$ is the k-th modality for the subset that samples contains i-th and j-th modality; ... (n) samples with $n-1$ modalities available $X^{ku\{M\backslash i\}} \in \mathbb{R}^{n_{\{M\backslash i\}}\times d_k}$ with $i = \{1, 2, ..., m\}$. We use $\{M\}$ denote the set of the index for all m modalities, i.e., $\{M\} = \{1, 2, ..., m\}$. $\{M\backslash i\}$ denotes the set without index i. k is an index taken from the set $\{M\backslash i\}$. $X^{ku_{\{M\backslash i\}}}$ is the k-th modality for the subset in which samples contain $\{M\backslash i\}$ modalities. We train the teacher models in a hierarchical manner. First, we train teacher models on each modality separately and obtain Te_i with $i = \{1, 2, \ldots, m\}$. Then, we use these models to teach the teacher models trained with two modalities and obtained teacher model Te_{ij} with $i, j = \{1, 2, \ldots, m\}$. Next, we use all the Te_{ij} to teach the teacher models trained with three modality and so forth. Finally, we obtain all the teachers hierarchically. Denote the teachers trained with h modalities as the h-level teachers. $\{C_h\}$ is the set that composed by all the combination of h indexes sampled from set M. The size of $\{C_h\}$ is $\binom{m}{h}$. For example, if $\{M\} = \{1, 2, 3, 4\}$, $\{C_2\} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$ and $\{C_3\} = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$. H-level teacher models are trained on the modalities indexed by the elements in $\{C_h\}$. For the above example, there are four 3-level teachers, i.e., a teacher trained with the modalities 1,2,3, a teacher trained with the modality 1,2,4, a teacher trained with modalities 1,3,4 and a teacher trained with modalities 2,3,4. Denote the model of the t-th teacher from the h-level teachers by $Te_{C_{ht}}(\phi_{ht})$ with ϕ_{ht} denoting the network parameters and C_{ht} denoting the t-th element of set $\{C_h\}$. For the above example, $C_{23} = \{1, 4\}$. $Te_{C_{ht}}(\phi_{ht})$ is trained by minimizing the following loss function:

$$\min_{\phi_{ht}} l_{C_{ht}} = \min_{\phi_{ht}} \sum_i^{N_{C_{ht}}} l_c(\{X_i^{ku_{C_{ht}}}\}_{k=C_{ht}}, y_i^{u_{C_{ht}}}; \phi_{ht}) + \sum_i^{N_{C_{ht}}} \sum_j^{|C_{h-1}|} \alpha_j l_d(\{X_i^{ku_{C_{ht}}}\}_{k=C_{ht}}; Te_{C_{(h-1)j}}), \tag{16}$$

where $|C_{h-1}|$ is the size of set C_{h-1} and $N_{C_{ht}}$ is the size of sample having modalities indexed by C_{ht}. After obtaining all teachers, we train the final student model with all the teachers.

One potential issue is that if we have a lot of modalities, the number of teacher models can be very lager. For m modalities, the complete number of teacher models is $2^m - 1$. As such, we cannot build all the teachers to train the student model due to the computational cost. As a solution, we propose to prune the teachers to improve the scalability of the proposed framework. A simple pruning strategy is to select a subset of teachers to train the student model. Basically, after first-level teachers are trained, i.e., single-modal teachers. We only select the teachers that have high performance to train the second level teachers. The modalities that have bad performance are also discarded when building the second level teachers. We build teachers at all other levels in the same way. Finally, all the remaining teachers are used to teach a student model build with m modalities. This pruning method drastically reduces the number of teachers and make the proposed method scalable. For example, for a dataset with five modalities, if in the first level we eliminate two teachers and in the second level we eliminate one teacher, the total teacher number is reduced to five. We demonstrate experiment on synthetic data to show the process of pruning and verify its effectiveness.

4 EXPERIMENT

In this section, we validate the proposed method and baselines on both synthetic and real datasets. The baselines included are (1) Te_i: the i-th teacher model (we use DNN as the teacher model in all the experiments[1]), (2) M-DNN: multimodal DNN trained only with the complete samples, (3) T-DNN: first using teacher models to extract abstract features and then training a DNN with the concatenation of these abstract features as input, (4) CAS-AE [25]: first using cascade residue autoencoder to impute the missing modalities and then training multimodal DNN with the original data and imputed data (5) ADV [2]: first using adversarial learning to generate the missing modalities and then training multimodal DNN with the original data and imputed data, (6) Subspace: multimodal subspace learning [24], (7) CCA [10]: canonical correlation analysis, (8) DCCA [1]: deep canonical correlation analysis, (9) T-LATE: weighted adding the teachers' logits, (10) MCTN [2] [17]: multimodal cyclic translation network. Our method is denoted as TS.

4.1 Synthetic data experiments

Setting 1: We synthesize data with two modalities in the following steps: (1) We draw n samples from $\mathcal{N}(1, I)$ and $\mathcal{N}(-1, I)$ separately. Samples from each normal distribution form one modality. Denote these samples as X^1 and X^2. The feature dimension is fixed to 32. (2) We then generate random weight matrices $W_1^1 \in \mathbb{R}^{32\times 64}, W_1^2 \in \mathbb{R}^{64\times 64}, W_2^1 \in \mathbb{R}^{32\times 64}, W_2^2 \in \mathbb{R}^{64\times 64}$ and use these weight matrices with ReLU function to transform the X^1 and X^2 to abstract features, i.e., $ReLU(ReLU(X^1W_1^1)W_1^2)$ and $ReLU(ReLU(X^2W_2^1)W_2^2)$. (3) After obtaining the transformed features for the two modalities, we concatenate those features to form the joint features and use a linear layer to transform the joint features to logits z. The final class

[1] Other models could also be used as teacher models. The reason we use DNN as the teacher model in our work is that DNN model's performance is relatively high compared with other commonly used classifiers. Ensemble models also have high performance. But DNN model could generate soft labels more easily than ensemble models.
[2] We use fully connected neural networks instead of RNNs for the encoder, decoder and the prediction subnetwork since our data are not time series data.

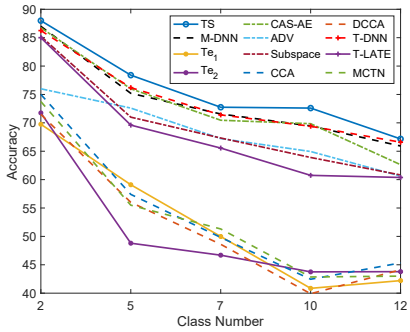


Figure 3: Classification accuracy for Setting 1. The proposed method (TS) outperforms all the other baselines.

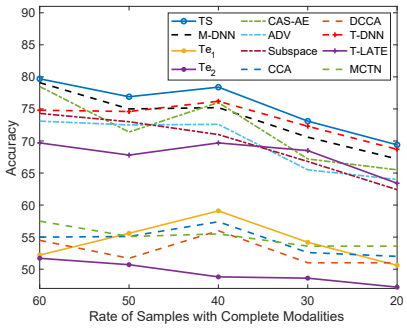


Figure 4: Classification accuracy for Setting 2. The proposed method (TS) outperforms all the other baselines.

label is $\sigma(z)$. When synthesizing the data, we make sure the number of samples for each class to be the same by generating more than n samples and downsampling. (4) We random select $a\%$ samples to be X^{1c} and X^{2c}. The remaining samples are divided into two equal parts. We remove one modality for each part to form X^{1u} and X^{2u}. So, X^{1u} and X^{2u} all have $n(1-a\%)/2$ samples. For each class, we randomly choose 80% of data as the training set, 10% as the validation set, and 10% as the testing set. We repeat the experiments for 5 times.

In this setting, we fix the number of samples per class to be 400 and change the class number in $\{2, 5, 7, 10, 12\}$. The samples with complete modalities are fixed to be 40%. The missing rate for each modality is 30%. The teacher model is a DNN model with 3 hidden layers and the hidden nodes are tuned in $\{32, 64, 128, 256\}$. For TS and M-DNN, we fix the network structure to be identical to the one used to generate the data but with unknown weight matrices. Since the two modalities have equal contribution to the output when we synthesize the data, we set α to be equal to β and is tuned in $\{0.1, 0.2, \ldots, 0.9\}$. Temperature T is tuned in $\{1, 5, 10, 15, 20\}$. Details of the settings of all other baselines are in Supplementary.

The results are shown in Figure 3. We see that TS outperforms all other models. The performance of Te_1 and Te_2 are much worse than M-DNN since each teacher only has access to the information of one modality. Although they are well-trained with all the available data, the information loss still makes the performance to be worse than M-DNN. The performance of the ADV and CAS-AE is lower than M-DNN because the imputed samples have low quality with limited samples having complete modalities. Although these two methods enlarge the sample size, they still cannot outperform M-DNN. Especially for ADV, the performance is much lower than M-DNN and CAS-AE since adversarial training is much more difficult than training an autoencoder. The difference between T-LATE and the TS model increases as the class number increase which implies that late fusion does not work well when the class number is large. The key difference between our model and T-DNN is that our model uses teachers to teach the student via labeling the samples, but T-DNN directly uses the features extracted by the teachers as the input features. The samples and model structures to train the teachers and the student models are all the same for the two methods. However, the performance of T-DNN is worse than the proposed method. One reason is that features extracted by teachers have lost some useful information.

Setting 2: In the second setting, the data are synthesized the same way as Setting 1. We change the rate of samples with complete modalities (complete rate) to be $\{60\%, 50\%, 40\%, 30\%, 20\%\}$. All the model structure and parameter settings are identical to Setting 1. The results are shown in Figure 4. We see similar patterns as Setting 1. When the complete rate is large, the performance of TS and M-DNN or CAS-AE is almost the same. But when the complete rate is small enough, TS is much better than M-DNN and CAS-AE since M-DNN and CAS-AE are trained well with a large complete rate. When the complete rate is small, there is no enough data to train them. T-DNN and T-LATE show the opposite pattern with M-DNN and CAS-AE, i.e., the difference between TS and these two models is smaller with a small complete rate than that with a large complete rate. T-DNN and T-LATE rely less on the complete samples. When the complete rate is small, the benefit of using large data to train the teachers makes them perform much better than the models only using complete samples. For our proposed model, we utilize this benefit to make sure the performance to be good when complete samples are scarce.

Setting 3: In this setting, we show the results of 5-modality synthetic data experiments. The challenge of 5-modality learning is from scalability since there are too many teachers available. We test the proposed pruning strategy in this section. The dataset is synthesized in the following way. (1) We draw n samples from $\mathcal{N}(1, I)$ and $\mathcal{N}(-1, I)$ separately. Samples from each normal distribution form one modality. Denote these samples as X^1 and X^2. The feature dimension is fixed to 32. (2) We use a random matrix $T \in \mathbb{R}^{32\times32}$ to linearly transform X^1 to form the third modality, i.e., $X^3 = X^1T$. (4) We take first half features from X^2 and then multiply a random matrix $M \in \mathbb{R}^{16\times32}$ to form modality 4. (5) We then draw n samples from $\mathcal{N}(0, I)$. The feature dimension is set to 32. These samples form the fifth modality. But when forming the joint representation, we only use the first half features of the fifth modality, denoted by $X^5_{1/2}$. (6) We then generate a random weight matrices $W^1_1, W^2_1, W^1_2, W^2_2$ and W^1_5, W^2_5. The size is 32 for W^1_1, W^1_2, 64×64 for W^2_1, W^2_2, 16×32 for W^1_5 and 32×32 for W^2_5. (7) We use ReLU as the nonlinear activation function. The joint representation is the concatenation of $ReLU(ReLU(X^1W^1_1)W^2_1)$, $ReLU(ReLU(X^2W^1_2)W^2_2)$ and $ReLU(ReLU(X^5_{1/2}W^1_5)W^2_5)$] We only use X^1, X^2 and X^5 to form joint representation because X^3 and X^4 are generated by X^1 and X^2. (8) A linear layer is added to the joint representation to generate the logits z. The final class label is $\sigma(z)$. (9) We randomly select

40% samples to be X^{1c}, X^{2c}, X^{3c}, X^{4c}, X^{5c}. We divide the remaining samples into three equal parts. We remove one modality for each part to form X^{1u}, X^{2u} and X^{5u}. X^{3u} has the same missing pattern with X^{1u} and X^{4u} has the same missing pattern with X^{2u}. For each class, we choose 80% of data as training, 10% as validation, and 10% as testing. Experiments are repeated 5 times.

We set the number of samples per class to be 1000 and the class number to be 5. We first train the teachers with every single modality. Then, we compare the performance of these teachers. The results are shown in Table 1. From Table 1, we see the performance of 4-th teacher and 5-teacher is relatively low compared with other teachers. Thus, we only use the first 3 teachers and modalities to form the two-modal teachers, which are Te_{12}, Te_{23} and Te_{13}. Then, we find the performance of Te_{13} is much worse than the performance of Te_{12} and Te_{23}. So, we do not need to train a 3-modality model with modality 1, 2, 3 as the teacher since it contains both the modality 1 and the modality 3. The final teacher we used are Te_1, Te_2, Te_3, Te_{12}, and Te_{23}. If we do not select teachers, the teacher number will be $2^5 - 1 = 31$. But now, we only need 5 teachers. As a comparison, we train models with modality 5 and 4 and then use them as teachers along with all the 5 teachers to teach the student model. The performance drops to 70.76 ± 0.01. So, when the performance of one teacher is too bad, we do not use this teacher to teach the student. We note that although modality 5 alone has bad performance, it still contributes to the joint representation as shown in the steps when we synthesize the data. We thus only use this method to select teachers but not the modalities used to train the student model.

4.2 Experiments on Alzheimer's diagnosis

In this subsection, we report the experiment performance on the union of two-stage of ADNI datasets [3], i.e., ADNI1 and ADNI2, and NACC dataset [4]. These datasets contain brain imaging data of subjects with different stages of Alzheimer's disease. Two modalities are used in this experiments. The first one is T1 MRI. 136 cortical volume and thickness features are extracted for 68 brain region of interests (ROIs) based on Desiken-Killiany atlas [3]. The second modality is dMRI-derived structural network. We use PICo [16] to construct brain networks for 113 ROIs based on the Harvard Oxford Cortical and subcortical Probabilistic Atlas [3, 5]. Since the network is undirected, we extract the upper triangle of the weighted adjacency matrix to form 6328 features. Finally, We use stability selection [14, 29] to select the top 172 features which have the top 30% stability scores as the final features for this modality. Our task is to classify if the subject is normal control (NC), mild cognitive impairment (MCI) or dementia (AD). The sample sizes on the three classes are (223,385,186) for ANDI1, (50, 112, 39) for ADNI2, and (329, 57, 53) for NACC respectively. ADNI2 and NACC have both dMRI and T1 MRI modalities while ADNI1 only has T1 MRI.

We train the teacher networks, the student network and M-DNN before the fusion layer with 4 hidden layers and the hidden node number is tuned in {256, 512, 1024}. After the fusion layer, a linear layer with SoftMax classifier is added to complete the classification. α and β are tuned in {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

[3] http://adni.loni.usc.edu
[4] https://www.alz.washington.edu

Model	Te_1	Te_2	Te_3	Te_4	Te_5
ACC	47.80 ± 0.09	47.04 ± 0.17	44.98 ± 0.26	34.48 ± 0.05	22.80 ± 0.04
Model	Te_{12}	Te_{23}	Te_{13}	M-DNN	TS
ACC	71.32 ± 0.03	68.84 ± 0.10	47.40 ± 0.3	71.44 ± 0.07	72.28 ± 0.03

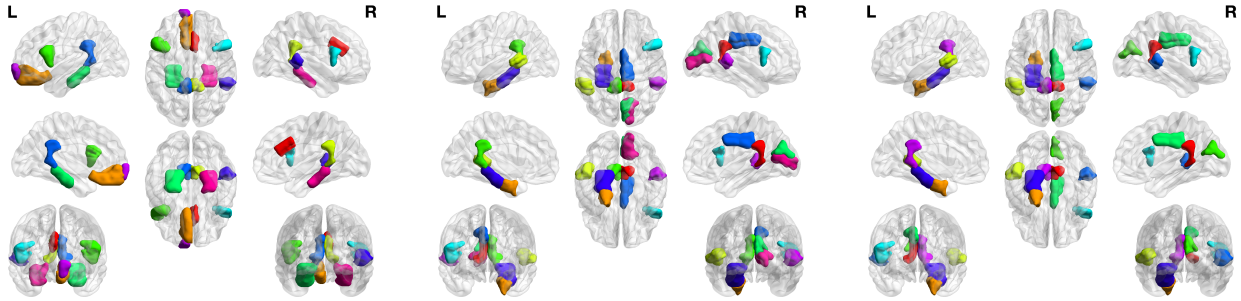
Table 1: Classification accuracy of Setting 3. We use the selected teachers to train the student model. As compassion, the accuracy drops to 70.76 ± 0.01 when adding non-selected teachers Te_4 and Te_5.

Model	TS	M-DNN	Te_1
Acc	75.48 ± 0.07	73.26 ± 0.08	69.67 ± 0.06
Model	Te_2	Subspace	MCTN
Acc	62.98 ± 0.01	67.66 ± 0.04	69.05 ± 0.11
Model	CCA	DCCA	CAS-AE
Acc	61.03 ± 0.47	72.70 ± 0.46	71.11 ± 0.01
Model	ADV	T-DNN	T-LATE
Acc	72.70 ± 0.05	72.27 ± 0.10	74.21 ± 0.01

Table 2: The classification accuracy for all the models trained on the union of ADNI and NACC datasets.

separately. For CAS-AE, we use 4 layers for encoder and 4 layers for decoder. The encoded features dimension is tuned in {128, 256}. For ADV, the hidden layer number for encoder and the discriminator is set to be 4. The node number is tuned in {256, 512, 1024}. For Subspace, we tune the rank in {32, 64, 128}. The projected feature dimension of CCA and DCCA is tuned in {32, 64, 128}. For MCTN, the hidden layer number is fixed to be 4 for encoder and decoder and the hidden node nunmber is tuned in {256, 512, 1024}. The prediction subnetwork hidden number is fixed to be 256. We random select 90% samples as training set and the rest as testing set. We repeat the experiment 5 times.

The average classification accuracy is reported in Table 2. We see our proposed method outperforms all other baselines. Te_1 is the teacher model trained on T1 MRI and Te_2 is the teacher model trained on the dMRI modality. For this dataset, all the samples have the first modality and only part of the samples have the second modality. So, the performance of Te_1 is much higher than the performance of Te_2. This is also reflected in the regularization parameters α and β. The best performance for our proposed model is reached when α is 0.7 and β is 0.0. Since dMRI modality is missing for some samples and T1 MRI modality is complete for all samples, the single teacher training on dRMI will be useless. Thus, when β is 0.0, the performance is the highest. We also show the top important features for Te_1, M-DNN and TS model in Figure 5. The features are ranked by the absolute weights value between the input layer and the first hidden layer. We sum all the absolute values of the weights that are connected with the input node as the relative importance of the associated input feature. We see there are some overlapping between the top important features of the three models but still some top features are very different for Te_1 and TS/M-DNN. For example, right isthmuscingulate thickness is ranked the third most important feature for the teacher models and the most important features for the student models. Left entorhinal volume is the second most important feature for M-DNN/TS but does not in the top 10 important features for the Te_1. Both two features have been proved to be related to Alzheimer's disease [7, 9]. The difference between the importance of the features causes T-DNN to be worst than TS model as T-DNN uses the features extracted by Te_1. Training with two modalities simultaneously leads to different



(a) Te_1 (b) M-DNN (c) TS

Figure 5: The top 10 important T1 MRI features for models trained on the union of NACC and ADNI datasets.

Rate	TS	M-DNN	Te_1	Te_2	CAE-AE	ADV	Subspace	DCCA	T-DNN	T-ENS	MCTN
40%	66.13 ± 0.03	62.66 ± 0.01	56.05 ± 0.01	45.73 ± 0.01	59.75 ± 0.02	59.37 ± 0.01	45.25 ± 0.02	41.94 ± 0.41	65.14 ± 0.02	63.69 ± 0.01	53.58 ± 0.01
30%	64.77 ± 0.01	60.59 ± 0.01	53.13 ± 0.01	42.59 ± 0.01	57.96 ± 0.01	57.83 ± 0.02	41.63 ± 0.01	41.64 ± 0.46	63.11 ± 0.01	61.91 ± 0.02	51.18 ± 0.02
20%	63.19 ± 0.01	57.18 ± 0.01	51.08 ± 0.01	41.63 ± 0.01	56.58 ± 0.01	56.37 ± 0.01	38.08 ± 0.01	33.53 ± 0.13	61.61 ± 0.01	59.70 ± 0.02	47.78 ± 0.03
10%	58.36 ± 0.01	50.33 ± 0.03	44.57 ± 0.01	37.93 ± 0.01	53.84 ± 0.01	53.60 ± 0.01	34.15 ± 0.01	32.86 ± 0.34	56.59 ± 0.01	56.13 ± 0.01	40.38 ± 0.02

Table 3: The classification accuracy of all the models trained on XRMB dataset.

Rate	TS	M-DNN	Te_1	Te_2	CAE-AE	ADV	Subspace	DCCA	T-DNN	T-ENS	MCTN
40%	96.46 ± 0.01	93.70 ± 0.01	93.04 ± 0.01	78.82 ± 0.09	94.54 ± 0.01	94.98 ± 0.01	86.70 ± 0.01	87.38 ± 0.09	95.18 ± 0.01	95.90 ± 0.01	92.24 ± 0.01
30%	96.00 ± 0.01	92.04 ± 0.03	91.78 ± 0.01	74.52 ± 0.02	94.26 ± 0.01	94.42 ± 0.01	84.34 ± 0.02	84.70 ± 0.15	94.92 ± 0.01	94.74 ± 0.01	90.22 ± 0.01
20%	95.42 ± 0.01	89.04 ± 0.02	90.72 ± 0.02	69.66 ± 0.06	93.72 ± 0.01	94.32 ± 0.01	79.76 ± 0.04	81.60 ± 0.16	92.25 ± 0.01	94.44 ± 0.01	88.86 ± 0.01
10%	92.34 ± 0.01	86.46 ± 0.01	87.12 ± 0.01	57.08 ± 0.08	91.48 ± 0.01	91.74 ± 0.01	72.28 ± 0.06	76.72 ± 0.31	92.28 ± 0.04	90.50 ± 0.01	85.02 ± 0.02

Table 4: The classification accuracy of all the models trained on MNIST dataset.

Model	TS	M-DNN	T-DNN
Accuracy	55.57 ± 0.02	47.43 ± 0.05	45.57 ± 0.02
Model	Te_{12}	Te_{13}	Te_{23}
Accuracy	48.57 ± 0.02	54.43 ± 0.06	52.29 ± 0.06
Model	Te_1	Te_2	Te_3
Accuracy	48.14 ± 0.01	45.14 ± 0.31	47.43 ± 0.24
Model	CAS-AE	ADV	T-ENS
Accuracy	53.27 ± 0.02	53.04 ± 0.06	53.86 ± 0.02

Table 5: The classification accuracy for the models trained on Alzheimer's disease data from [38].

feature ranks since the two modalities are coupled and influence each other. Some features in one modality alone do not show to be important. But these features could be very important with the presence of some features from the other modality.

4.3 Experiments on other real-world datasets

In this section, we report the performance on three additional real-world datasets. The first one is Alzheimer's disease data from [38], which has 3 modalities and 3 classes available, i.e., MRI, PET, Proteomics. The feature dimensions for these 3 modalities are 305, 116 and 147, respectively. In this dataset, 648 subjects have MRI data. 372 subjects have PET data. 496 subjects have Proteomics data. Only 215 subjects have all three modalities. We randomly split the data into the training set and testing set with the ratio 0.9 : 0.1. The parameters are tuned the same way as Section 4.2. We repeat the experiments for 5 iterations. The average accuracy is shown in Table 5. From the table, we see the performance of M-DNN is even worst than Te_{13} since when training the M-DNN with all the three modalities, the sample size is much smaller than that used to train Te_{13}. But with the teaching step, the performance improves a lot and outperforms the performance of Te_{13}.

Another two real-world datasets we used are MNIST and XRMB [32]. For MNIST data, we subsample 10,000 as training data, 1,000 samples as validation data and 1,000 samples as testing data. The class number is 10. MNIST has two modalities with 784 features for each modality. For XRMB data, we subsample 19,500 samples for training, 1,950 for validation and 1,950 for testing. The class number for XRMB is 39. Two modalities are available for XRMB data with 273 and 112 features. Since these data do not have missing modalities, we randomly choose $a\%$ of samples to be the samples with complete modalities. And for the rest part of the data, we split them into two parts and remove one modality for each part. We change the rate of complete modalities in {40%, 30%, 20%, 10%}. The parameters are tuned the same way as Section 4.2 except for the node number. The hidden layer node number is tuned in {512, 1024, 2048}. The encoded feature dimension for CAS-ADV is tuned in {128, 256, 512}. The projected feature numbers for CCA, DCCA and Subspace are tuned in {128, 256, 512} for MNIST and {32, 64, 100} for XRMB. The experiments are repeated for 5 times and the results are shown in Table 3 and Table 4. We see that our method outperforms all other baselines under different missing rates.

5 CONCLUSION

In this paper, we proposed a novel framework to fuse the supplementary information of multiple modalities for the datasets with missing modalities. We first trained models on each modality with all the available data to obtain teacher models. Then, we used these teacher models to teach a multimodal DNN network by knowledge

distillation. Since the teacher models were trained on relatively larger datasets compared with the datasets used to train the student model, the teachers were experts on each modality and the expertise could help the student to improve the performance. The experiment results on both synthetic and real-world data verified the effectiveness of the proposed method.

ACKNOWLEDGMENTS

This research is supported in part by NSF under Grant IIS-1749940 (JZ), IIS-1615597 (JZ), and IIS-1837956 (LZ), NIA under Grand AG05 6782 (LZ), and NIH under Grant U54EB020403 (PT), R56AG058854 (PT), R01MH121246 (PT), and RF1AG051710 (PT). Data used in the preparation of this article were obtained from the ADNI database (http://adni.loni.usc.edu) and NACC database (https://www.alz.washington.edu).

REFERENCES

[1] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *International Conference on Machine Learning*. 1247–1255.

[2] Lei Cai, Zhengyang Wang, Hongyang Gao, Dinggang Shen, and Shuiwang Ji. 2018. Deep adversarial learning for multi-modality missing data completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1158–1166.

[3] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage* 31, 3 (2006), 968–980.

[4] Craig K Enders. 2010. *Applied missing data analysis*. Guilford press.

[5] Jean A Frazier, Sufen Chiu, Janis L Breeze, Nikos Makris, Nicholas Lange, David N Kennedy, Martha R Herbert, Eileen K Bent, Vamsi K Koneru, Megan E Dieterich, et al. 2005. Structural brain magnetic resonance imaging of limbic and thalamic volumes in pediatric bipolar disorder. *American Journal of Psychiatry* 162, 7 (2005), 1256–1265.

[6] Hatice Gunes and Massimo Piccardi. 2005. Affect recognition from face and body: early fusion vs. late fusion. In *2005 IEEE international conference on systems, man and cybernetics*, Vol. 4. IEEE, 3437–3443.

[7] Leticia Gutiérrez-Galve, Manja Lehmann, Nicola Z Hobbs, Matthew J Clarkson, Gerard R Ridgway, Sebastian Crutch, Sebastien Ourselin, Jonathan M Schott, Nick C Fox, and Josephine Barnes. 2009. Patterns of cortical thickness according to APOE genotype in Alzheimers disease. *Dementia and geriatric cognitive disorders* 28, 5 (2009), 461–470.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[9] K Juottonen, MP Laakso, R Insausti, M Lehtovirta, A Pitkänen, K Partanen, and H Soininen. 1998. Volumes of the entorhinal and perirhinal cortices in Alzheimers disease. *Neurobiology of aging* 19, 1 (1998), 15–22.

[10] Sham M Kakade and Dean P Foster. 2007. Multi-view regression via canonical correlation analysis. In *International Conference on Computational Learning Theory*. Springer, 82–96.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[12] Weifeng Liu and Dacheng Tao. 2013. Multiview hessian regularization for image annotation. *IEEE Transactions on Image Processing* 22, 7 (2013), 2676–2687.

[13] Rahil Mehrizi, Xi Peng, Zhiqiang Tang, Xu Xu, Dimitris Metaxas, and Kang Li. 2018. Toward marker-free 3D pose estimation in lifting: A deep multi-view solution. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 485–491.

[14] Nicolai Meinshausen and Peter Bühlmann. 2010. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72, 4 (2010), 417–473.

[15] Yongsheng Pan, Mingxia Liu, Chunfeng Lian, Tao Zhou, Yong Xia, and Dinggang Shen. 2018. Synthesizing missing PET from MRI with cycle-consistent generative adversarial networks for Alzheimers disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 455–463.

[16] Geoffrey JM Parker, Hamied A Haroon, and Claudia AM Wheeler-Kingshott. 2003. A framework for a streamline-based probabilistic index of connectivity (PICo) using a structural interpretation of MRI diffusion measurements. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* 18, 2 (2003), 242–254.

[17] Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. 2019. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6892–6899.

[18] Snehashis Roy, John A Butman, Daniel S Reich, Peter A Calabresi, and Dzung L Pham. 2018. Multiple sclerosis lesion segmentation from brain MRI via fully convolutional neural networks. *arXiv preprint arXiv:1803.09172* (2018).

[19] Alexander G Schwing and Raquel Urtasun. 2015. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351* (2015).

[20] Weixiang Shao, Xiaoxiao Shi, and S Yu Philip. 2013. Clustering on multiple incomplete datasets via collective kernel learning. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1181–1186.

[21] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 399–402.

[22] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 909–912.

[23] Qiuling Suo, Weida Zhong, Fenglong Ma, Ye Yuan, Jing Gao, and Aidong Zhang. 2019. Metric learning on healthcare data with incomplete modalities. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 3534–3540.

[24] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, and Zili Zhang. 2018. Incomplete multi-view weak-label learning.. In *IJCAI*. 2703–2709.

[25] Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin. 2017. Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1405–1414.

[26] Grigorios Tzortzis and Aristidis Likas. 2012. Kernel-based weighted multi-view clustering. In *2012 IEEE 12th international conference on data mining*. IEEE, 675–684.

[27] Manik Varma and Bodla Rakesh Babu. 2009. More generality in efficient multiple kernel learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 1065–1072.

[28] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. 2018. Partial multi-view clustering via consistent GAN. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1290–1295.

[29] Qi Wang, Lei Guo, Paul M Thompson, Clifford R Jack Jr, Hiroko Dodge, Liang Zhan, Jiayu Zhou, Alzheimers Disease Neuroimaging Initiative, et al. 2018. The added value of diffusion-weighted MRI-derived structural connectome in evaluating mild cognitive impairment: A multi-cohort validation. *Journal of Alzheimer's Disease* 64, 1 (2018), 149–169.

[30] Qi Wang, Mengying Sun, Liang Zhan, Paul Thompson, Shuiwang Ji, and Jiayu Zhou. 2017. Multi-Modality Disease Modeling via Collective Deep Matrix Factorization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1155–1164.

[31] Qi Wang, Liang Zhan, Paul M Thompson, Hiroko H Dodge, and Jiayu Zhou. 2016. Discriminative fusion of multiple brain networks for early mild cognitive impairment detection. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 568–572.

[32] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *International Conference on Machine Learning*. 1083–1092.

[33] Jennifer Williams, Steven Kleinegesse, Ramona Comanescu, and Oana Radu. 2018. Recognizing emotions in video using multimodal dnn feature fusion. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*. 11–19.

[34] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* (2013).

[35] Qiyue Yin, Shu Wu, and Liang Wang. 2015. Incomplete multi-view clustering via subspace learning. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 383–392.

[36] Qiyue Yin, Shu Wu, and Liang Wang. 2017. Unified subspace learning for incomplete and unlabeled multi-view data. *Pattern Recognition* 67 (2017), 313–327.

[37] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit Dhillon. 2012. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 765–774.

[38] Lei Yuan, Yalin Wang, Paul M Thompson, Vaibhav A Narayan, and Jieping Ye. 2012. Multi-source learning for joint analysis of incomplete multi-modality neuroimaging data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1149–1157.

[39] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4320–4328.

SUPPLEMENTARY

5.1 The settings of the baselines for synthetic data experiment Setting 1.

For T-DNN, we use the layer before the output layer of the teacher models as the abstract features. These abstract features are concatenated to form new features. Then, we train a DNN model with the new features. The DNN model has 3 hidden layers with node number being tuned in $\{64, 128, 256\}$. For each block of the autoencoder in the CAS-AE model, the encoder has 3 layers and decode has 3 layers. The encoded feature dimension is fixed to be 64 since the original data has 32 features for each modality. The node number for the hidden layer of the encoder and decoder is tuned in $\{128, 256, 512\}$. We follow the steps in the [25] to tune the number of the autoencoder block, i.e., the joint optimization of the entire network is performed when adding one autoencoder block. During the training phase, we randomly choose half samples from the complete samples to remove one modality and the other half to remove the other modality. Then, we train the CAS-AE to reconstruct the removed modalities. After the training, the CAS-AE is used to impute the missing modalities for the incomplete samples. Finally, we train a multimodal DNN using all the imputed samples and the complete samples together. The structure of the multimodal DNN used here is the same as the student model of TS and M-DNN model. For ADV, the encoder part is a 3 layer DNN, the hidden node number is tuned in $\{128, 256, 512\}$. The structure of discriminator is a 3-layer DNN with hidden number be tuned in $\{128, 256, 512\}$. Since ADV can only impute one modality in one time, we first use the first modality to impute the second modality with the complete samples as the training data. Then, we use the imputed samples and the complete samples as training data to train a second model to impute the first modality. After we impute all the missing part, we train a multimodal DNN to perform the classification. The structure of the multimodal DNN is the same with the student model of TS and M-DNN model. The formulation of Subspace baseline is identical to Eq. (2) in [24]. We initialize the latent factors by SVD of the concatenation of two modality to improve the performance of this model. The latent factor rank is tuned in $\{16, 32, 64\}$. For CCA and DCCA, the projected feature dimension is tuned in $\{16, 32\}$. The hidden node of DCCA is tuned in $\{64, 128, 256, 512\}$ and the hidden layer number is fixed to be 3. For T-LATE, we first use the training samples to learn the optimal weights for each teacher. Then, we use the learned weights and teacher models to label the testing samples. For MCTN, the hidden node of encoder and decoder is tuned in $\{64, 128, 256, 512\}$ and the hidden layer number is fixed to be 3. The prediction subnetwork has one hidden layer and the hidden node number is fixed to be 128.

5.2 Parameters study for the union of ADNI and NACC data

Figure 6 shows how the accuracy changes with the parameter α and β. In this figure, we change α when fixing β to be 0.0 and change β when fixing α to be 0.7. The performance decreases with the increasing of β. Meanwhile, the teacher trained with T1 MRI improves the performance a lot with a large α.

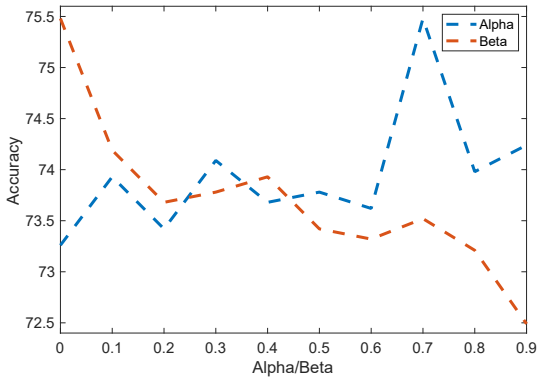
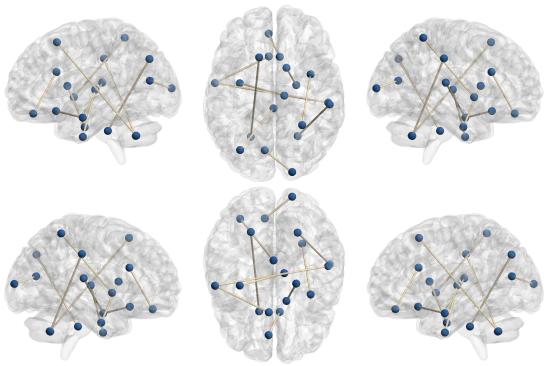
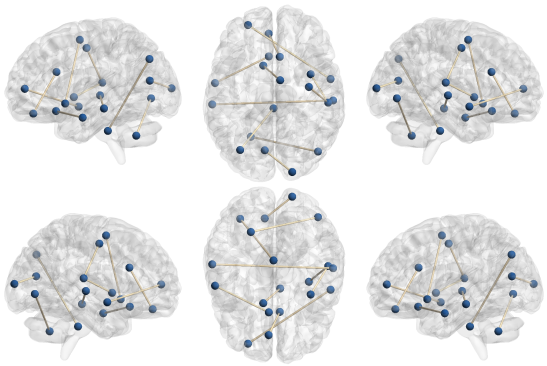


Figure 6: Accuracy with different α and β. α is fixed to be 0.7 while changing β and β is fixed to be 0.0 while changing α.



(a) Te_2



(b) TS/M-DNN

Figure 7: The top 10 important dMRI features for models trained on the union of NACC and ADNI datasets.

5.3 Top ranked dMRI features for different models on the union of ADNI and NACC datasets

Figure 7 shows the top ranked features/connections for Te_2 and TS/M-DNN. For this modality, M-DNN and TS have the same top ranked features since the best performance reached when $\beta = 0$. The important features/connections are quite different for Te_1 and TS/M-DNN.