Information Fusion 68 (2021) 118–130

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Multimodal deep generative adversarial models for scalable doubly semi-supervised learning

Changde Du [a,b,c], Changying Du [d], Huiguang He [a,b,e,*]

[a] Research Center for Brain-inspired Intelligence and National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences, Beijing, 100190, China
[b] University of Chinese Academy of Sciences, Beijing, 100190, China
[c] Huawei Cloud BU EI Innovation Lab, Beijing 100085, China
[d] Huawei Noah's Ark Lab, Beijing 100085, China
[e] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing, 100190, China

ARTICLE INFO

Keywords:
Multiview learning
Multimodal fusion
Generative adversarial networks
Deep generative models
Semi-supervised learning

ABSTRACT

The comprehensive utilization of incomplete multi-modality data is a difficult problem with strong practical value. Most of the previous multimodal learning algorithms require massive training data with complete modalities and annotated labels, which greatly limits their practicality. Although some existing algorithms can be used to complete the data imputation task, they still have two disadvantages: (1) they cannot control the semantics of the imputed modalities accurately; and (2) they need to establish multiple independent converters between any two modalities when extended to multimodal cases. To overcome these limitations, we propose a novel doubly semi-supervised multimodal learning (DSML) framework. Specifically, DSML uses a modality-shared latent space and multiple modality-specific generators to associate multiple modalities together. Here we divided the shared latent space into two independent parts, the semantic labels and the semantic-free styles, which allows us to easily control the semantics of generated samples. In addition, each modality has its own separate encoder and classifier to infer the corresponding semantic and semantic-free latent variables. The above DSML framework can be adversarially trained by using our specially designed softmax-based discriminators. Large amounts of experimental results show that the DSML obtains better performance than the baselines on three tasks, including semi-supervised classification, missing modality imputation and cross-modality retrieval.

1. Introduction

With the development of sensor technology, researchers are more and more interested in the acquiring and modeling of multimodal data [1,2]. Distinct modalities offer complementary strengths, and in many cases, using multiple modalities together can yield a solution that is much better than either one by itself. Successful cases can be found in many applications, such as emotion recognition [3–6], object recognition [7,8], disease diagnosis [9,10], etc. Previous multimodal learning algorithms [11–13] generally assume that all available training instances have complete modalities and corresponding labels. However, in practice, the assumption does not hold, because (1) modality missing may occur at some data points due to some unforeseeable reasons, such as sensor failure; (2) in some applications, such as brain decoding [14,15], obtaining multimodal data is expensive, while collecting enough single-modal data is easy; and (3) the data labeling procedure is often labor-intensive, so in most cases we only have a small number of labeled samples available. Therefore the traditional multimodal learning approaches cannot effectively handle the incomplete multi-modality situations, especially in a semi-supervised manner.

Toward overcoming the aforementioned incomplete data problems, researchers have proposed several cross-modality data imputation methods [16–18]. For example, Luan et al. [16] proposed to stack a series of residual autoencoders on top of each other to capture the relatedness among different modalities. Du et al. [17] proposed the use of the adversarial multi-view autoencoding model to achieve the mutual prediction of multiple modalities. Cai et al. [18] regarded the existing modality as conditions and used conditional deep generative adversarial model to generate the target modality. Unfortunately, the approaches mentioned above still have two common shortcomings.

* Corresponding author at: Research Center for Brain-inspired Intelligence and National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences, Beijing, 100190, China.
E-mail addresses: duchangde@gmail.com (C. Du), ducyatict@gmail.com (C. Du), huiguang.he@ia.ac.cn (H. He).

https://doi.org/10.1016/j.inffus.2020.11.003
Received 6 May 2020; Received in revised form 7 October 2020; Accepted 7 November 2020
Available online 16 November 2020

First of all, these methods are not flexible enough to control the semantic information of the imputed modalities. Previous studies typically used unsupervised translation methods [19] to model the modality inference problems, which does not explicitly utilize the semantic information of the input modalities. How to utilize the semantic information of the input modality to flexibly control the semantics of the output modality is still a challenging problem. Second, these methods are not scalable enough to handle multiple modalities. Most of the existing methods need to learn two translator going in opposite directions for the mutual mapping between two modalities. When scaling up to the n ($n > 2$) modalities, they need to build $n(n-1)$ translators, which quickly becomes unfeasible as the number of modalities increases.

In this paper, we propose a scalable doubly semi-supervised multimodal learning framework (DSML) with deep generative adversarial models [20]. Instead of learning the direct modality mappings, our DSML uses the latent space shared by modalities and the modality-specific generators to model the relationships between different modalities. The benefits of using modality-shared latent space are threefold. (1) Mutual inference between modalities through the shared latent space only require $2n$ low-dimensional mappings. This is more scalable and efficient than the direct transformation between high-dimensional modalities. (2) The shared latent space allows us to generate a large number of paired synthetic samples, which can be used to augment the training set. (3) The trained low-dimensional latent space supports efficient similarity calculation and fast cross-modal retrieval.

We divide the shared latent variable into two parts $(\boldsymbol{c}, \boldsymbol{z})$, where $\boldsymbol{c}$ contains the semantic information (category labels), and $\boldsymbol{z}$ contains the semantic-free factors (background, color, etc.). The decoupling of $\boldsymbol{c}$ and $\boldsymbol{z}$ allows us to flexibly control the semantics of the generated samples. However, $\boldsymbol{c}$ and $\boldsymbol{z}$ might be entangled together in training phase without specific constrains. To solve this problem, we construct two inference networks (i.e., the classifier and encoder) for each modality to minimize the reconstruction error of $\boldsymbol{c}$ and $\boldsymbol{z}$ in the latent space, respectively. This architecture naturally enables efficient cross-modality translations through the shared latent space. Finally, we carefully design an adversarial training method to train all involved modules (the generators, classifiers and encoders).

Experimental results on multiple datasets demonstrate that the proposed DSML as a unified framework can simultaneously (1) achieve the state-of-the-art multimodal semi-supervised classification results; (2) recover the missing modality with high visual quality and correct intrinsic semantics; and (3) perform efficient cross-modality retrieval in the modality-shared latent space.

The main contributions can be summarized as follows.

- By matching three kinds of the joint distributions, we develop a doubly semi-supervised learning framework, which can simultaneously leverage the complete and incomplete multimodal data (missing labels, missing modalities or both) to improve the performance of downstream tasks.
- The proposed modality-shared and semantic disentangled latent space can (1) impute the missing modality more efficient, controllable and scalable; (2) naturally support for multimodal fusion, semi-supervised learning and cross-modality retrieval.
- We design a softmax-based multiclass discriminator to distinguish multiple distinct joint distributions in adversarial training procedures.
- We show the experimental results on a lot of downstream tasks, such as semi-supervised classification, missing modality imputation and cross-modality retrieval.

2. Related work

The proposed DSML framework focuses on doubly semi-supervised multimodal learning with deep generative adversarial networks (GANs) [21]. There has been recent interest in employing deep generative models to learn the joint distributions of two domains/modalities. They can be roughly divided into the following three categories. (1) *Generation and inference*: one domain consists of the unobservable latent variables, and the other domain consist of the observable data; (2) *Cross-modality translation*: both domains consist of the observable data samples, but some of the samples are described by only one modality; (3) *Semi-supervised classification*: one domain consists of the observable data samples, and the other domain consist of the corresponding data labels of partial samples. Below, we introduce them, respectively.

2.1. Generation and inference

Adversarially learned inference (ALI) [22] and BiGAN [23] are deep generative models which train a generative network and an inference network jointly using adversarial loss. These models provide a novel way to integrate inference model into the GAN framework, and hence they can learn the joint distribution of the data samples $\boldsymbol{x}$ and the latent codes $\boldsymbol{z}$. Specifically, the objective of ALI can be written as

$$\min_{G_x, E_z} \max_{D} \mathcal{L}_{\mathrm{ALI}} = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}),\ \tilde{\boldsymbol{z}}\sim p_z(\boldsymbol{z}|\boldsymbol{x})}\left[\log D(\boldsymbol{x}, \tilde{\boldsymbol{z}})\right] + \mathbb{E}_{\tilde{\boldsymbol{x}}\sim p_x(\boldsymbol{x}|\boldsymbol{z}),\ \boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log\left(1 - D(\tilde{\boldsymbol{x}}, \boldsymbol{z})\right)\right], \tag{1}$$

where $p_x(\boldsymbol{x}|\boldsymbol{z}) \equiv G_x$ is the generator for the data samples, $p_z(\boldsymbol{z}|\boldsymbol{x}) \equiv E_z$ is the encoder for the latent codes, and $D(\boldsymbol{x}, \boldsymbol{z})$ is the discriminator trained to distinguish the two kind of joint samples $(\boldsymbol{x}, \tilde{\boldsymbol{z}})$ and $(\tilde{\boldsymbol{x}}, \boldsymbol{z})$. To solve the non-identifiability issue (the mapping between random variables $\boldsymbol{x}$ and $\boldsymbol{z}$ is not specified) associated with ALI, the ALICE model [24] regularizes ALI using the conditional entropy framework, which is equivalent to the cycle-consistency principle in CycleGAN [19]. More recently, Du et al. [17] proposed a multi-view ALI (MALI) model based on a shared latent space for cross-domain generation.

2.2. Cross-modality translation

The purpose of a cross-modality translation is to predict one modality from another. The methods in this field can be classified as supervised [25,26], semi-supervised [24,27–29], and unsupervised [19,30] models, depending on whether the training set contains paired samples. As a general semi-supervised cross-modality translation model, Δ-GAN [27] can learn the bi-directional mappings between modalities $\boldsymbol{x}$ and $\boldsymbol{y}$. Actually, Δ-GAN can be considered as a combination of ALI and the conditional GAN [31]. Its goal is to match the three joint distributions: $p(\boldsymbol{x}, \boldsymbol{y})$, $p_x(\boldsymbol{x}, \boldsymbol{y}) = p_x(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})$ and $p_y(\boldsymbol{x}, \boldsymbol{y}) = p_y(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$. If this is achieved, we obtain the bidirectional mappings $p_x(\boldsymbol{x}|\boldsymbol{y})$ and $p_y(\boldsymbol{y}|\boldsymbol{x})$. Representative unsupervised cross-modality translation work includes the DiscoGAN [30], CycleGAN [19], etc. For instance, DiscoGAN uses two generators to build the relationships between the different modalities, and two discriminators to distinguish the real and fake data in each individual modality. Although the above cross-modality translation models have achieved encouraging results in some fields, they lack latent variable inference mechanism.

2.3. Semi-supervised classification

The GAN frameworks have been applied to semi-supervised classification tasks [32–35]. These models are trained to match the distributions characterized by the classifier and the generator with the real data distribution. For example, TripleGAN [32] achieves the state-of-the-art semi-supervised classification results by using adversarial joint distribution matching. Similar work includes SGAN [33], etc. Besides GANs, the variational autoencoders (VAEs) [36] have also been applied to semi-supervised learning [6,37]. For example, Du et al. [6] proposed a semi-supervised incomplete multi-view VAE model (SiMVAE), which consider the missing labels or modalities as latent variables and infer them automatically. Unlike SiMVAE, the latent variable of DSML has been divided into two disentangled parts. Further, SiMVAE can only be applied to semi-supervised classification, while our DSML model is naturally suitable for more other applications, i.e., missing modality imputation and cross-modality retrieval.

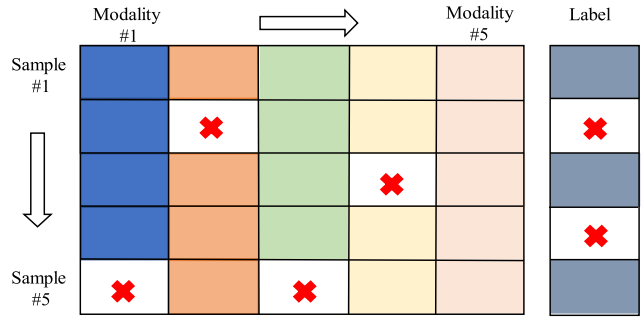


Fig. 1. In doubly semi-supervised learning, both the data labels and the data modalities are incomplete. The red cross indicates that the entry is missing or unavailable. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
The frequently used symbols and their definitions.

Symbol	Definition
$\boldsymbol{x}$	Data modality
$\boldsymbol{y}$	Data modality
$\boldsymbol{c}$	Modality-shared semantic label
$\boldsymbol{z}$	Modality-shared latent code
G_x, G_y	Modality-specific generators
C_x, C_y	Modality-specific classifiers
E_x, E_y	Modality-specific encoders
D_{xz}	Binary discriminator
D_{yz}	Binary discriminator
D_{xc}	3-way softmax discriminator
D_{yc}	3-way softmax discriminator
D_{xy}	4-way softmax discriminator
$\mathbb{E}[\cdot]$	Expectation operator

3. Methodology

3.1. Doubly semi-supervised learning

In multimodal learning, we are often faced with data scenarios in which both labels and modalities are incomplete (cf. Fig. 1). For simplicity, we first consider the case of two modalities, and it is straightforward to extend to multiple (more than two) modalities.

For a data sample, we assume $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{c}$ denote the first modality, the second modality and the category label, respectively. Then, we have three forms of empirical distributions for unlabeled data, i.e., the paired data $p(\boldsymbol{x},\boldsymbol{y})$, the unpaired data $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$. Similarly, we have three forms of empirical distributions for labeled data, i.e., the paired data $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c})$, the unpaired data $p(\boldsymbol{x},\boldsymbol{c})$ and $p(\boldsymbol{y},\boldsymbol{c})$. Since the labeled data with complete modalities usually is insufficient, a good multimodal learning model should also benefit from the empirical distributions $p(\boldsymbol{x},\boldsymbol{c})$, $p(\boldsymbol{y},\boldsymbol{c})$, $p(\boldsymbol{x})$, $p(\boldsymbol{y})$ and $p(\boldsymbol{x},\boldsymbol{y})$ as much as possible, which is referred to as doubly semi-supervised learning (SSL).

To facilitate reading, the frequently used symbols and their definitions are listed in Table 1.

3.2. The proposed DSML framework

The illustration of the proposed doubly semi-supervised multimodal learning framework are shown in Fig. 2. We assume $\boldsymbol{x}$ and $\boldsymbol{y}$ are two distinct modalities of the same instance, and they are generated from a shared latent space through the corresponding generators G_x and G_y, respectively (see Fig. 2a). Here the shared latent variable is separated into two independent parts $(\boldsymbol{c},\boldsymbol{z})$, where $\boldsymbol{c}$ contains the specific semantic information, and $\boldsymbol{z}$ contains the semantic-free factors (background, color, etc.). If we don not impose any constraints on $\boldsymbol{c}$ and $\boldsymbol{z}$, they might be entangled together in model training. To address this issue, we

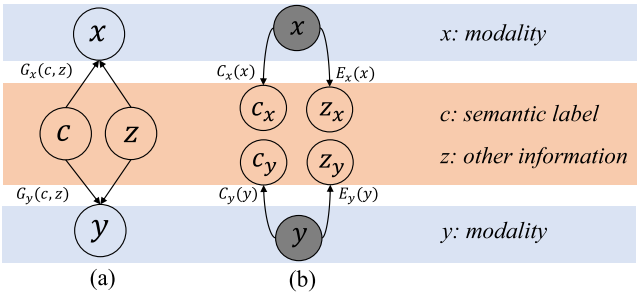


Fig. 2. Illustrations of the proposed doubly semi-supervised multimodal learning (DSML) framework. (a) the generators G_x, G_y; (b) the classifiers C_x, C_y and the encoders E_x, E_y. The gray and white units represent the observed and latent variables, respectively.

build two separate inference networks for each modality, one acts as a classifier and the other as a common encoder (cf. Fig. 2b). For modality $\boldsymbol{x}$, the encoder E_x and the classifier C_x are optimized to minimize the reconstruction errors of $\boldsymbol{c}$ and $\boldsymbol{z}$, respectively. The same is true for modality $\boldsymbol{y}$. We elaborate the key modules as follows.

- **Generators** G_x, G_y. We assume the following modality-specific generative processes for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively: $\boldsymbol{z} \sim p(\boldsymbol{z}), \boldsymbol{c} \sim p(\boldsymbol{c}), \tilde{\boldsymbol{x}} \sim p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z}) \equiv G_x, \tilde{\boldsymbol{y}} \sim p_{g_y}(\boldsymbol{y}|\boldsymbol{c},\boldsymbol{z}) \equiv G_y$, where $p(\boldsymbol{z})$ is specified as a simple prior (e.g., isotropic Gaussian), and $p(\boldsymbol{c})$ as an appropriate prior that meets our modeling needs (e.g. a categorical distribution). G_x and G_y take the common $(\boldsymbol{c},\boldsymbol{z})$ as input, and output the generated sample pair $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$.
- **Classifiers** C_x, C_y. We assume the following modality-specific inference process from $\boldsymbol{x}/\boldsymbol{y}$ to $\boldsymbol{c}$: $\boldsymbol{x} \sim p(\boldsymbol{x}), \tilde{\boldsymbol{c}}_x \sim p_{c_x}(\boldsymbol{c}|\boldsymbol{x}) \equiv C_x$, $\boldsymbol{y} \sim p(\boldsymbol{y}), \tilde{\boldsymbol{c}}_y \sim p_{c_y}(\boldsymbol{c}|\boldsymbol{y}) \equiv C_y$, where $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the empirical marginal distributions. C_x and C_y are trained to approximate the target posteriors $p(\boldsymbol{c}|\boldsymbol{x})$ and $p(\boldsymbol{c}|\boldsymbol{y})$, respectively. If $\boldsymbol{c}$ is categorical label, both C_x and C_y reduce to two N-way classifiers.
- **Encoders** E_x, E_y. We assume the following modality-specific inference process from $\boldsymbol{x}/\boldsymbol{y}$ to $\boldsymbol{z}$: $\boldsymbol{x} \sim p(\boldsymbol{x}), \tilde{\boldsymbol{z}}_x \sim p_{e_x}(\boldsymbol{z}|\boldsymbol{x}) \equiv E_x$, $\boldsymbol{y} \sim p(\boldsymbol{y}), \tilde{\boldsymbol{z}}_y \sim p_{e_y}(\boldsymbol{z}|\boldsymbol{y}) \equiv E_y$. E_x and E_y are trained to approximate the target posteriors $p(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{y})$, respectively, and we force their outputs to be close to each other for a given data pair $(\boldsymbol{x},\boldsymbol{y})$. Note that, E_x and E_y can also be seen as generators for the latent code $\boldsymbol{z}$, which encodes the style information.

All of the above modules are implemented as deep neural networks (DNNs), whose architecture depends on specific applications such as deconvolution neural networks for image generation. In the next subsection, we show how to use adversarial learning to jointly optimize all modules.

3.3. Jointly adversarial training of DSML

We employ the idea of adversarial learning to train each module of the above DSML model. The adversarial training flowchart is shown in Fig. 3. In the following, we will describe the whole training process in detail.

3.3.1. Matching the joint distribution of data and code

Inspired by the ALI [22] model, we first adversarially match the joint distributions of data and its latent code. For modality $\boldsymbol{x}$, our goal is to align $p_{g_x}(\boldsymbol{x},\boldsymbol{z}) = \int p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})p(\boldsymbol{c})p(\boldsymbol{z})d\boldsymbol{c}$ with $p_{e_x}(\boldsymbol{x},\boldsymbol{z}) = p_{e_x}(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})$. To draw samples from $p_{g_x}(\boldsymbol{x},\boldsymbol{z})$, we first draw the tuple $(\tilde{\boldsymbol{x}},\boldsymbol{c},\boldsymbol{z})$ following $\boldsymbol{c} \sim p(\boldsymbol{c}), \boldsymbol{z} \sim p(\boldsymbol{z}), \tilde{\boldsymbol{x}} \sim p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})$, and then only taking $(\tilde{\boldsymbol{x}},\boldsymbol{z})$ as needed. This implicitly integrates out $\boldsymbol{c}$. On the other hand, drawing samples from $p_{e_x}(\boldsymbol{x},\boldsymbol{z})$ is straightforward: $\boldsymbol{x} \sim p(\boldsymbol{x}), \tilde{\boldsymbol{z}} \sim p_{e_x}(\boldsymbol{z}|\boldsymbol{x})$.

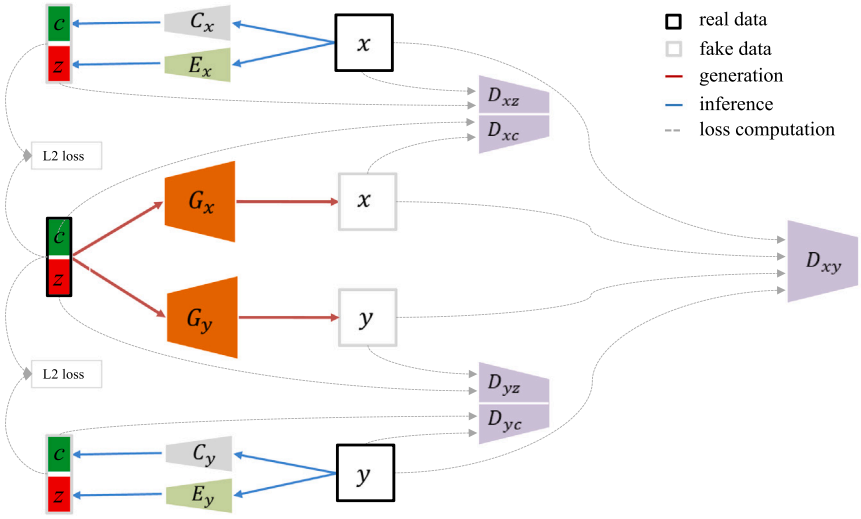
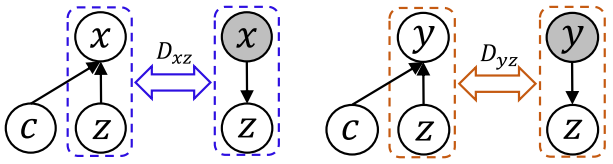
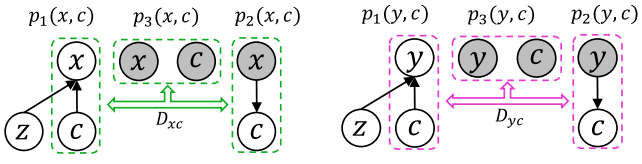


Fig. 3. The adversarial training flowchart of the proposed DSML. x and y modalities are generated from a shared latent variable through the generators G_x and G_y, respectively. Here the shared latent variable contains two independent parts (c, z), where c means the category labels, and z encodes the other information. There are two separate inference networks for each modality, where C_x (or C_y) acts as a classifier and E_x (or E_y) as a common encoder. The generators, classifiers and encoders are adversarially joint trained using the specially designed discriminators D_{xz}, D_{xc}, D_{yz}, D_{yc} and D_{xy} with the latent variable reconstruction regularization.



(a) $\mathcal{L}_{xz}$ (b) $\mathcal{L}_{yz}$

Fig. 4. Adversarial games $\mathcal{L}_{xz}$ (blue) and $\mathcal{L}_{yz}$ (orange). The discriminator D_{xz} (or D_{yz}) is trained to distinguish two different kinds of joint distributions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) $\mathcal{L}_{xc}$ (b) $\mathcal{L}_{yc}$

Fig. 5. Adversarial games $\mathcal{L}_{xc}$ (green) and $\mathcal{L}_{yc}$ (pink). The discriminator D_{xc} (or D_{yc}) is trained to distinguish three different kinds of joint distributions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The objective function can then be expressed as the following minimax optimization problem:

$$\min_{G_x,E_x}\max_{D_{xz}} \mathcal{L}_{xz} = \mathbb{E}_{(\tilde{\boldsymbol{x}},\boldsymbol{z})\sim p_{g_x}(\boldsymbol{x},\boldsymbol{z})}\left[\log D_{xz}(\tilde{\boldsymbol{x}},\boldsymbol{z})\right] + \mathbb{E}_{(\boldsymbol{x},\tilde{\boldsymbol{z}})\sim p_{e_x}(\boldsymbol{x},\boldsymbol{z})}\left[\log\left(1-D_{xz}(\boldsymbol{x},\tilde{\boldsymbol{z}})\right)\right], \tag{2}$$

where the discriminator D_{xz} is trained to distinguish the joint pairs sampled from $p_{g_x}(\boldsymbol{x},\boldsymbol{z})$ and $p_{e_x}(\boldsymbol{x},\boldsymbol{z})$, respectively (cf. Fig. 4a). G_x and E_x reach the optimal solution if and only if $p_{g_x}(\boldsymbol{x},\boldsymbol{z}) = p_{e_x}(\boldsymbol{x},\boldsymbol{z})$ [22]. Similarly, we have the minimax game $\mathcal{L}_{yz}$ for modality $\boldsymbol{y}$,

$$\min_{G_y,E_y}\max_{D_{yz}} \mathcal{L}_{yz} = \mathbb{E}_{(\tilde{\boldsymbol{y}},\boldsymbol{z})\sim p_{g_y}(\boldsymbol{y},\boldsymbol{z})}\left[\log D_{yz}(\tilde{\boldsymbol{y}},\boldsymbol{z})\right] + \mathbb{E}_{(\boldsymbol{y},\tilde{\boldsymbol{z}})\sim p_{e_y}(\boldsymbol{y},\boldsymbol{z})}\left[\log\left(1-D_{yz}(\boldsymbol{y},\tilde{\boldsymbol{z}})\right)\right]. \tag{3}$$

In order to make $\boldsymbol{z}$ completely capture the semantic-free information without entangling with $\boldsymbol{c}$, we force $\boldsymbol{z}$ can be reconstructed from the generated data $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$ through the corresponding encoders. The meaning of $\boldsymbol{z}$ is not preassigned but learned automatically on a given dataset. For example, on the MNIST-to-MNIST-transpose dataset the learned meanings of $\boldsymbol{z}$ are thicknesses, inclination and so on. Assume $\hat{\boldsymbol{z}}_x \sim p_{e_x}(\boldsymbol{z}|\tilde{\boldsymbol{x}})$ and $\hat{\boldsymbol{z}}_y \sim p_{e_y}(\boldsymbol{z}|\tilde{\boldsymbol{y}})$ denote the latent code reconstructions via $(\boldsymbol{c},\boldsymbol{z}) \to \tilde{\boldsymbol{x}} \to \hat{\boldsymbol{z}}_x$ and $(\boldsymbol{c},\boldsymbol{z}) \to \tilde{\boldsymbol{y}} \to \hat{\boldsymbol{z}}_y$, respectively. Then reconstruction losses can be written as

$$\min_{G_x,G_y,E_x,E_y} \mathcal{R}_{\boldsymbol{z}} = \mathbb{E}_{\boldsymbol{c},\boldsymbol{z},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\hat{\boldsymbol{z}}_x,\hat{\boldsymbol{z}}_y}\left[\|\hat{\boldsymbol{z}}_x-\boldsymbol{z}\|_2 + \|\hat{\boldsymbol{z}}_y-\boldsymbol{z}\|_2\right]. \tag{4}$$

Intuitively, minimizing $\mathcal{R}_{\boldsymbol{z}}$ will yield small $\|E_x(G_x(\boldsymbol{c}_1,\boldsymbol{z})) - E_x(G_x(\boldsymbol{c}_2,\boldsymbol{z}))\|_2$ and $\|E_y(G_y(\boldsymbol{c}_1,\boldsymbol{z})) - E_y(G_y(\boldsymbol{c}_2,\boldsymbol{z}))\|_2$, $\forall \boldsymbol{c}_1,\boldsymbol{c}_2 \sim p(\boldsymbol{c})$, which indicates that $\boldsymbol{z}$ has been disentangled from $\boldsymbol{c}$. The reconstruction of $\boldsymbol{z}$ can also be interpreted as applying the cycle-consistency principle [19] in the latent space. For the observable data pairs $(\boldsymbol{x},\boldsymbol{y})$, their latent codes $\tilde{\boldsymbol{z}}_x \sim p_{e_x}(\boldsymbol{z}|\boldsymbol{x})$ and $\tilde{\boldsymbol{z}}_y \sim p_{e_y}(\boldsymbol{z}|\boldsymbol{y})$ should be used to stabilize the training of E_x and E_y before G_x and G_y can generate real-like data pairs $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$. Overall, the regularization term for latent code can be expressed as:

$$\min_{G_x,G_y,E_x,E_y} \mathcal{R}^*_{\boldsymbol{z}} = \mathcal{R}_{\boldsymbol{z}} + \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{z}}_x,\tilde{\boldsymbol{z}}_y}\left[\|\tilde{\boldsymbol{z}}_x-\tilde{\boldsymbol{z}}_y\|_2\right]. \tag{5}$$

3.3.2. Matching the joint distribution of data and label

The objective of adversarial semi-supervised classification is to force the joint distribution of data and label formed by the generator and classifier both to converge to the empirical joint distribution. For modality $\boldsymbol{x}$, we need to match the joint distribution of data and label pairs $(\boldsymbol{x},\boldsymbol{c})$ drawn from $p_1(\boldsymbol{x},\boldsymbol{c})$, $p_2(\boldsymbol{x},\boldsymbol{c})$ and $p_3(\boldsymbol{x},\boldsymbol{c})$, respectively, where

$$p_1(\boldsymbol{x},\boldsymbol{c}) = \int_{\boldsymbol{z}} p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})p(\boldsymbol{c})p(\boldsymbol{z})d\boldsymbol{z},$$
$$p_2(\boldsymbol{x},\boldsymbol{c}) = p(\boldsymbol{x})p_{c_x}(\boldsymbol{c}|\boldsymbol{x}), \quad p_3(\boldsymbol{x},\boldsymbol{c}) = p(\boldsymbol{x},\boldsymbol{c}). \tag{6}$$

Here $p_3(\boldsymbol{x},\boldsymbol{c})$ is the empirical joint distribution formed by the observable data and label pairs $(\boldsymbol{x},\boldsymbol{c})$. In adversarial training, joint pairs $(\boldsymbol{x},\boldsymbol{c})$ are drawn from these three kinds of joint distributions, and a discriminator D_{xc} is learned to distinguish among them (cf. Fig. 5), while the generator G_x and the classifier C_x are trained to mislead the discriminator.

We can naively use two binary discriminators to distinguish the three different data pairs [27]. However, the results of the two binary discriminators may conflict during training [28]. To more consistently distinguish the three joint pairs, we design the discriminator D_{xc} as a neural network with 3-way softmax on the top layer, i.e., $\sum_{k=1}^{3} D_{xc}(\boldsymbol{x},\boldsymbol{c})[k] = 1$ and $D_{xc}(\boldsymbol{x},\boldsymbol{c})[k] \in (0,1)$, where $D_{xc}(\boldsymbol{x},\boldsymbol{c})[k]$ is an entry of $D_{xc}(\boldsymbol{x},\boldsymbol{c})$. The objective function can be expressed as

$$\min_{G_x,C_x}\max_{D_{xc}} \mathcal{L}_{\boldsymbol{xc}} = \sum_{k=1}^{3} \mathbb{E}_{p_k(\boldsymbol{x},\boldsymbol{c})}\Big[\log D_{xc}(\boldsymbol{x},\boldsymbol{c})[k]\Big]. \tag{7}$$

Our softmax-based discriminator can be considered as sharing the parameters between two binary discriminators except the top layer, thus reducing the number of parameters.

However, in practice, there is little supervision to tell the generator $p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})$ what $\boldsymbol{c}$ essentially represents. The result is that G_x may produce a low quality sample that is inconsistent with their labels. To solve this problem, we force the classifier $p_{c_x}(\boldsymbol{c}|\boldsymbol{x})$ to reconstruct $\boldsymbol{c}$ by using the regularization term $\mathcal{R}_{\boldsymbol{xc}}$,

$$\min_{G_x,C_x} \mathcal{R}_{\boldsymbol{xc}} = \underbrace{\mathbb{E}_{(\boldsymbol{x},\boldsymbol{c})\sim p_3(\boldsymbol{x},\boldsymbol{c})}[-\log p_{c_x}(\boldsymbol{c}|\boldsymbol{x})]}_{\text{Classification loss on real data}} + \underbrace{\mathbb{E}_{(\boldsymbol{x},\boldsymbol{c})\sim p_1(\boldsymbol{x},\boldsymbol{c})}[-\log p_{c_x}(\boldsymbol{c}|\boldsymbol{x})]}_{\text{Classification loss on generated data}}. \tag{8}$$

Intuitively, in order to assign semantic labels to the variable $\boldsymbol{c}$, the optimization of Eq. (8) will minimize the classification loss of the classifier C_x on real and generated data. On the one hand, minimizing the first term w.r.t. C_x on real data guides $p_{c_x}(\boldsymbol{c}|\boldsymbol{x})$ toward the true posterior $p(\boldsymbol{c}|\boldsymbol{x})$. On the other hand, minimizing the second term w.r.t. G_x and C_x on generated data can backpropagate the gradient to G_x to enhance its controllability — it minimizes the chance that G_x could generate samples that would otherwise be falsely predicted by C_x. Once G_x can generate high-quality samples that match the semantic label $\boldsymbol{c}$, we can make full use of the generated data pairs $(\boldsymbol{x},\boldsymbol{c}) \sim p_1(\boldsymbol{x},\boldsymbol{c})$ to improve the generalization ability of C_x. This idea has been proven to be effective in SSL [32,33].

Since we use a similar strategy for the modality $\boldsymbol{y}$, the corresponding adversarial game $\mathcal{L}_{\boldsymbol{yc}}$ is

$$\min_{G_y,C_y}\max_{D_{yc}} \mathcal{L}_{\boldsymbol{yc}} = \sum_{k=1}^{3} \mathbb{E}_{p_k(\boldsymbol{y},\boldsymbol{c})}\Big[\log D_{yc}(\boldsymbol{y},\boldsymbol{c})[k]\Big], \tag{9}$$

and the regularizer is

$$\min_{G_y,C_y} \mathcal{R}_{\boldsymbol{yc}} = \mathbb{E}_{(\boldsymbol{y},\boldsymbol{c})\sim p_3(\boldsymbol{y},\boldsymbol{c})}[-\log p_{c_y}(\boldsymbol{c}|\boldsymbol{y})] + \mathbb{E}_{(\boldsymbol{y},\boldsymbol{c})\sim p_1(\boldsymbol{y},\boldsymbol{c})}[-\log p_{c_y}(\boldsymbol{c}|\boldsymbol{y})]. \tag{10}$$

Because the classifier is modality-specific, we handle the classification of multi-modality data by combining the results of multiple classifiers. For example, the predicted label for two-modality data can be written as: label = softmax$(O_x + O_y)$, where O_x and O_y are the outputs before the softmax layer of each classifiers, respectively.

3.3.3. Matching the joint distribution of two modalities

Model performance can be improved by introducing an additional discriminator D_{xy} to drive $p_1(\boldsymbol{x},\boldsymbol{y})$, $p_2(\boldsymbol{x},\boldsymbol{y})$ and $p_3(\boldsymbol{x},\boldsymbol{y})$ to converge to the empirical joint distribution $p_4(\boldsymbol{x},\boldsymbol{y})$ (cf. Fig. 6), where $p_1(\boldsymbol{x},\boldsymbol{y}), \ldots, p_4(\boldsymbol{x},\boldsymbol{y})$ denote the distributions of joint pairs $(\boldsymbol{x}_{\text{fake}},\boldsymbol{y}_{\text{fake}})$, $(\boldsymbol{x}_{\text{fake}},\boldsymbol{y}_{\text{real}})$, $(\boldsymbol{x}_{\text{real}},\boldsymbol{y}_{\text{fake}})$ and $(\boldsymbol{x}_{\text{real}},\boldsymbol{y}_{\text{real}})$, respectively, and

$$\begin{aligned}
p_1(\boldsymbol{x},\boldsymbol{y}) &= \int_{\boldsymbol{c}}\int_{\boldsymbol{z}} p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})p_{g_y}(\boldsymbol{y}|\boldsymbol{c},\boldsymbol{z})p(\boldsymbol{c})p(\boldsymbol{z})d\boldsymbol{c}d\boldsymbol{z},\\
p_2(\boldsymbol{x},\boldsymbol{y}) &= p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})p_{c_y}(\boldsymbol{c}|\boldsymbol{y})p_{e_y}(\boldsymbol{z}|\boldsymbol{y})p(\boldsymbol{y}),\\
p_3(\boldsymbol{x},\boldsymbol{y}) &= p_{g_y}(\boldsymbol{y}|\boldsymbol{c},\boldsymbol{z})p_{c_x}(\boldsymbol{c}|\boldsymbol{x})p_{e_x}(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x}),\\
p_4(\boldsymbol{x},\boldsymbol{y}) &= p(\boldsymbol{x},\boldsymbol{y}),
\end{aligned} \tag{11}$$

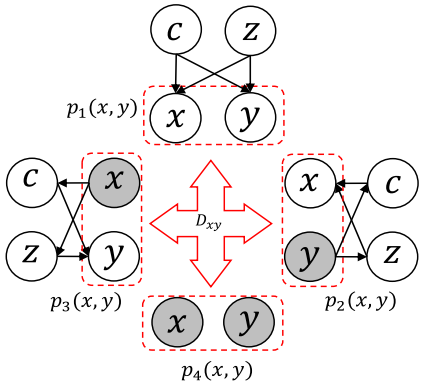


Fig. 6. Adversarial game $\mathcal{L}_{xy}$ (red). The discriminator D_{xy} is trained to distinguish these four kinds of joint distributions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Similar to Eq. (7), here D_{xy} should be a 4-way softmax-based discriminator, and the objective function can then be expressed as the following minimax optimization problem:

$$\min_{\boldsymbol{G},\boldsymbol{E},\boldsymbol{C}}\max_{D_{xy}} \mathcal{L}_{\boldsymbol{xy}} = \sum_{k=1}^{4} \mathbb{E}_{p_k(\boldsymbol{x},\boldsymbol{y})}\Big[\log D_{xy}(\boldsymbol{x},\boldsymbol{y})[k]\Big], \tag{12}$$

where $\boldsymbol{G} = \{G_x, G_y\}$, $\boldsymbol{E} = \{E_x, E_y\}$ and $\boldsymbol{C} = \{C_x, C_y\}$. Since the empirical distribution $p_4(\boldsymbol{x},\boldsymbol{y})$ consists of only a few pairs of observable samples, it may be biased. Fortunately, once G_x and G_y can generate high-quality samples, we can correct this bias with the generated high-quality samples $(\boldsymbol{x},\boldsymbol{y}) \sim p_1(\boldsymbol{x},\boldsymbol{y})$.

3.3.4. Full objective function

In summary, DSML is fully differentiable and can be trained end-to-end. The full objective can be written as

$$\begin{aligned}
\min_{\boldsymbol{G},\boldsymbol{E},\boldsymbol{C}}\max_{\boldsymbol{D}} \mathcal{L}_{\text{DSML}} &= \alpha_1\mathcal{L}_{\boldsymbol{xz}} + \alpha_2\mathcal{L}_{\boldsymbol{yz}} + \beta_1\mathcal{L}_{\boldsymbol{xc}} + \beta_2\mathcal{L}_{\boldsymbol{yc}} + \gamma\mathcal{L}_{\boldsymbol{xy}}\\
&\quad + \lambda_1\mathcal{R}_{\boldsymbol{xc}} + \lambda_2\mathcal{R}_{\boldsymbol{yc}} + \lambda_3\mathcal{R}^{*}_{\boldsymbol{z}},
\end{aligned} \tag{13}$$

where $\boldsymbol{D} = \{D_{xz}, D_{yz}, D_{xc}, D_{yc}, D_{xy}\}$. Every term of the above formula is meaningful, and all of them are complementary to each other. The whole training procedures are described in Algorithm 1. As GAN-based framework is inherently difficult to train due to the unbalance between discriminators and generators, many methods have been proposed to stabilize and improve the adversarial training, e.g., spectral normalization GAN [38], Wasserstein GAN [39] and energy-based GAN [40], etc. These techniques can also be applied to our framework to improve the training processes. Here, we adopt two off-the-shelf strategies in the implementation of DSML to ease the training. First, we used spectral normalization [38] in the discriminators. Second, We use multiple (e.g., 5) discriminator update steps per generator update step during training. Moreover, to guarantee that the classifiers could be properly trained, we pretrain C_x by minimizing the first term of $\mathcal{R}_{\boldsymbol{yc}}$, and pretrain C_y by minimizing the first term of $\mathcal{R}_{\boldsymbol{yc}}$ on the available labeled training data.

3.3.5. Convergence analysis

Depending on the type of learning, the terms in the objective function (13) can be divided into two categories, namely, distribution matching terms ($\mathcal{L}_{\boldsymbol{xz}}$, $\mathcal{L}_{\boldsymbol{yz}}$, $\mathcal{L}_{\boldsymbol{xc}}$, $\mathcal{L}_{\boldsymbol{yc}}$ and $\mathcal{L}_{\boldsymbol{xy}}$) and regularization terms ($\mathcal{R}_{\boldsymbol{xc}}$, $\mathcal{R}_{\boldsymbol{yc}}$ and $\mathcal{R}^{*}_{\boldsymbol{z}}$). For the distribution matching problem, the ideal way is to directly match the joint distribution of all of the involved variables $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c},\boldsymbol{z})$ (there are 16 different cases). However, in doubly semi-supervised learning where both modalities and labels are incomplete, the joint draws from $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c},\boldsymbol{z})$ and $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c})$ may not be

Algorithm 1 Training of DSML in doubly SSL.

Input: Available complete data: $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{c})$; Available incomplete data: $(\boldsymbol{x},\boldsymbol{c})$, $(\boldsymbol{y},\boldsymbol{c})$, $(\boldsymbol{x},\boldsymbol{y})$, $\boldsymbol{x}$, $\boldsymbol{y}$; Hyper-parameters: $\alpha_1,\alpha_2,\beta_1,\beta_2,\gamma,\lambda_1,\lambda_2,\lambda_3$, K.

Pretrain C_x by minimizing the first term of Eq. (8) w.r.t. C_x using labeled data pairs $(\boldsymbol{x},\boldsymbol{c})$, and C_y by minimizing the first term of Eq. (10) w.r.t. C_y using labeled data pairs $(\boldsymbol{y},\boldsymbol{c})$
for number of training iterations **do**
  Sample batches of $\boldsymbol{x}$ and $\boldsymbol{y}$: $\boldsymbol{x}\sim p(\boldsymbol{x})$, $\boldsymbol{y}\sim p(\boldsymbol{y})$.
  Sample batches of $(\boldsymbol{x},\boldsymbol{z})$ from $p_{g_x}(\boldsymbol{x},\boldsymbol{z})$ and $p_{e_x}(\boldsymbol{x},\boldsymbol{z})$; sample batches of $(\boldsymbol{y},\boldsymbol{z})$ from $p_{g_y}(\boldsymbol{y},\boldsymbol{z})$ and $p_{e_y}(\boldsymbol{y},\boldsymbol{z})$.
  Sample batches of $(\boldsymbol{x},\boldsymbol{c})$ from $p_1(\boldsymbol{x},\boldsymbol{c})$,⋯, $p_3(\boldsymbol{x},\boldsymbol{c})$; sample batches of $(\boldsymbol{y},\boldsymbol{c})$ from $p_1(\boldsymbol{y},\boldsymbol{c})$,⋯, $p_3(\boldsymbol{y},\boldsymbol{c})$.
  Sample batches of $(\boldsymbol{x},\boldsymbol{y})$ from $p_1(\boldsymbol{x},\boldsymbol{y})$,⋯, $p_4(\boldsymbol{x},\boldsymbol{y})$.
  for $k=1\rightarrow K$ **do**
    Train D_{xz}, D_{yz} by maximizing $\mathcal{L}_{\boldsymbol{xz}}$ and $\mathcal{L}_{\boldsymbol{yz}}$ using batches of $(\boldsymbol{x},\boldsymbol{z})$ and $(\boldsymbol{y},\boldsymbol{z})$, respectively.
    Train D_{xc}, D_{yc} by maximizing $\mathcal{L}_{\boldsymbol{xc}}$ and $\mathcal{L}_{\boldsymbol{yc}}$ using batches of $(\boldsymbol{x},\boldsymbol{c})$ and $(\boldsymbol{y},\boldsymbol{c})$, respectively.
    Train D_{xy} by maximizing $\mathcal{L}_{\boldsymbol{xy}}$ using batches of $(\boldsymbol{x},\boldsymbol{y})$.
  end for
  Train E_x,E_y by minimizing $\alpha_1\mathcal{L}_{\boldsymbol{xz}}+\alpha_2\mathcal{L}_{\boldsymbol{yz}}+\lambda_3\mathcal{R}^*_{\boldsymbol{z}}$ using batches of $(\boldsymbol{x},\boldsymbol{z})$ and $(\boldsymbol{y},\boldsymbol{z})$.
  Train C_x,C_y by minimizing $\beta_1\mathcal{L}_{\boldsymbol{xc}}+\beta_2\mathcal{L}_{\boldsymbol{yc}}+\lambda_1\mathcal{R}_{\boldsymbol{xc}}+\lambda_2\mathcal{R}_{\boldsymbol{yc}}$ using batches of $(\boldsymbol{x},\boldsymbol{c})$ and $(\boldsymbol{y},\boldsymbol{c})$.
  Train G_x,G_y by minimizing $\mathcal{L}_{\text{DSML}}$ using involved data pairs.
end for

Output: The optimized DSML model.

easy to access. By contrast, the draws from the marginal distributions $p(\boldsymbol{x},\boldsymbol{c})$, $p(\boldsymbol{y},\boldsymbol{c})$ and $p(\boldsymbol{x},\boldsymbol{y})$ are easier to obtain. Therefore, in practice, we assume that only empirical draws from the marginal distributions $p(\boldsymbol{x},\boldsymbol{c})$, $p(\boldsymbol{y},\boldsymbol{c})$ and $p(\boldsymbol{x},\boldsymbol{y})$ are available, and use $\mathcal{L}_{\boldsymbol{xc}}$, $\mathcal{L}_{\boldsymbol{yc}}$ and $\mathcal{L}_{\boldsymbol{xy}}$ in the objective function. Since $\boldsymbol{z}$ denotes the latent representation of $\boldsymbol{x}$ or $\boldsymbol{y}$, it is impossible to obtain the empirical draws from $p(\boldsymbol{x},\boldsymbol{z})$ or $p(\boldsymbol{y},\boldsymbol{z})$. Fortunately, we can take an ALI-like approach [22] to optimize $\mathcal{L}_{\boldsymbol{xz}}$ and $\mathcal{L}_{\boldsymbol{yz}}$.

In Propositions 1–3, we proved that each of the five distribution matching subproblem ($\mathcal{L}_{\boldsymbol{xz}}$, $\mathcal{L}_{\boldsymbol{yz}}$, $\mathcal{L}_{\boldsymbol{xc}}$, $\mathcal{L}_{\boldsymbol{yc}}$ or $\mathcal{L}_{\boldsymbol{xy}}$) has its own optimal solution, and give the necessary conditions to obtain that optimal solution. In Proposition 4, we proved that optimizing with the regularization terms $\mathcal{R}_{\boldsymbol{xc}}$, $\mathcal{R}_{\boldsymbol{yc}}$ and $\mathcal{R}^*_{\boldsymbol{z}}$ will not change the optimal solutions of the distribution matching subproblems. Therefore, our DSML framework theoretically has clear convergence properties through carefully adversarial training.

Proposition 1. *The equilibrium for the minimax objective $\mathcal{L}_{\boldsymbol{xz}}$ is achieved if and only if $p_{g_x}(\boldsymbol{x},\boldsymbol{z})=p_{e_x}(\boldsymbol{x},\boldsymbol{z})$ with the optimal discriminator $D^*_{xz}(\boldsymbol{x},\boldsymbol{z})=\frac{1}{2}$. Similarly, the equilibrium for the minimax objective $\mathcal{L}_{\boldsymbol{yz}}$ is achieved if and only if $p_{g_y}(\boldsymbol{y},\boldsymbol{z})=p_{e_y}(\boldsymbol{y},\boldsymbol{z})$ with the optimal discriminator $D^*_{yz}(\boldsymbol{y},\boldsymbol{z})=\frac{1}{2}$.*

Proof. The proof can be found in [22].

Proposition 2. *The equilibrium for the minimax objective $\mathcal{L}_{\boldsymbol{xc}}$ is achieved if and only if $p_1(\boldsymbol{x},\boldsymbol{c})=p_2(\boldsymbol{x},\boldsymbol{c})=p_3(\boldsymbol{x},\boldsymbol{c})$ with the optimal discriminator $D^*_{xc}(\boldsymbol{x},\boldsymbol{c})[k]=\frac{1}{3}$. Similarly, the equilibrium for the minimax objective $\mathcal{L}_{\boldsymbol{yc}}$ is achieved if and only if $p_1(\boldsymbol{y},\boldsymbol{c})=p_2(\boldsymbol{y},\boldsymbol{c})=p_3(\boldsymbol{y},\boldsymbol{c})$ with the optimal discriminator $D^*_{yc}(\boldsymbol{y},\boldsymbol{c})[k]=\frac{1}{3}$.*

Proof. The proof is provided in Appendix A.

Proposition 3. *The equilibrium for the minimax objective $\mathcal{L}_{\boldsymbol{xy}}$ is achieved if and only if $p_1(\boldsymbol{x},\boldsymbol{y})=p_2(\boldsymbol{x},\boldsymbol{y})=p_3(\boldsymbol{x},\boldsymbol{y})=p_4(\boldsymbol{x},\boldsymbol{y})$ with the optimal discriminator value $D^*_{xy}(\boldsymbol{x},\boldsymbol{y})[k]=\frac{1}{4}$.*

Proof. The proof is provided in Appendix B.

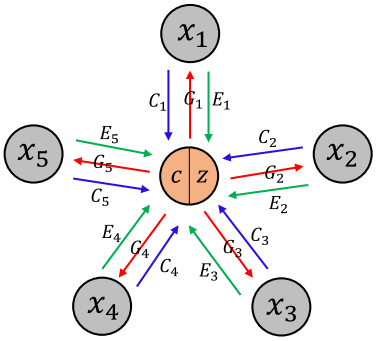


Fig. 7. The framework of DSML when extend to multiple modalities. Here G_i, E_i and C_i $(i=1,\ldots,5)$ represent the generator, encoder and classifier for ith modality, respectively.

Proposition 4. *Minimizing $\mathcal{R}^*_{\boldsymbol{z}}$ w.r.t. $\boldsymbol{E}$ will not change the equilibrium of the minimax objective $\mathcal{L}_{\boldsymbol{xz}}$ and $\mathcal{L}_{\boldsymbol{yz}}$. Similarly, minimizing $\mathcal{R}_{\boldsymbol{xc}}$ w.r.t. C_x or minimizing $\mathcal{R}_{\boldsymbol{yc}}$ w.r.t. C_y will not change the equilibrium of $\mathcal{L}_{\boldsymbol{xc}}$ and $\mathcal{L}_{\boldsymbol{yc}}$, respectively.*

Proof. Since $\mathcal{R}^*_{\boldsymbol{z}}$ is always non-negative, the optimum is obtained if and only if $\hat{\boldsymbol{z}}_x\sim p_{e_x}(\boldsymbol{z}|\tilde{\boldsymbol{x}})=\hat{\boldsymbol{z}}_y\sim p_{e_y}(\boldsymbol{z}|\tilde{\boldsymbol{y}})=\boldsymbol{z}$, where $\tilde{\boldsymbol{x}}\sim p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})$, $\tilde{\boldsymbol{y}}\sim p_{g_y}(\boldsymbol{y}|\boldsymbol{c},\boldsymbol{z})$, which is equivalent to $p_{g_x}(\boldsymbol{x},\boldsymbol{z})=p_{e_x}(\boldsymbol{x},\boldsymbol{z})$. The proof for $\mathcal{R}_{\boldsymbol{xc}}$ and $\mathcal{R}_{\boldsymbol{yc}}$ are similar.

3.3.6. Exploiting cycle consistencies

Previous studies [19] have shown that cycle consistency regularization is able to improve the effect of modality translation. Under the DSML framework, one can naively force $\boldsymbol{x}\rightarrow(\tilde{\boldsymbol{c}},\tilde{\boldsymbol{z}})\rightarrow\hat{\boldsymbol{y}}\rightarrow(\hat{\boldsymbol{c}},\hat{\boldsymbol{z}})\rightarrow\hat{\boldsymbol{x}}$ to yield small $\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2$, and $\boldsymbol{y}\rightarrow(\tilde{\boldsymbol{c}},\tilde{\boldsymbol{z}})\rightarrow\hat{\boldsymbol{x}}\rightarrow(\hat{\boldsymbol{c}},\hat{\boldsymbol{z}})\rightarrow\hat{\boldsymbol{y}}$ to yield small $\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2$. However, the ℓ_2 losses often lead to the blurry results. Moreover, it is usually hard to tune the regularization parameter balancing the adversarial loss and cycle consistency loss. To address the issues, we equivalently implement the cycle consistency principle by using the following adversarial regularizer:

$$\min_{G,E,C}\max_{D_{xx},D_{yy}}\mathcal{R}_{\boldsymbol{xy}}=\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}),\hat{\boldsymbol{x}}}\left[\log\left(D_{xx}(\boldsymbol{x},\boldsymbol{x})\cdot(1-D_{xx}(\boldsymbol{x},\hat{\boldsymbol{x}}))\right)\right]+\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y}),\hat{\boldsymbol{y}}}\left[\log\left(D_{yy}(\boldsymbol{y},\boldsymbol{y})\cdot(1-D_{yy}(\boldsymbol{y},\hat{\boldsymbol{y}}))\right)\right],\tag{14}$$

where D_{xx} and D_{yy} are the discriminators introduced to distinguish between two kind of joints, respectively. We will verify the role of cycle consistency in the experiments.

3.4. Extended to multiple modalities

The above DSML framework can be naturally extended to multiple modalities. Fig. 7 shows a schematic of the extension to five modalities. Without loss of generality, when the model is extended to n-modalities, we need n modality-specific generators/encoders/classifiers. As for the discriminator D_{xy} in this case, we adopt a special design. Specifically, its input is a combination of all modalities, and we consider the following $(n+2)$ kinds of combinations: (1) all n modalities are real, (2) all n modalities are fake, and (3) Only one of the n modalities is fake, and the other $(n-1)$ modalities are real. In such case, the output layer of the discriminator resembles a $(n+2)$-way classifier.

Although our DSML method also becomes more complex as the number of modality increases, it still has a huge advantage over the traditional cross-modal translation methods. In practice, if distinct modalities have different properties (e.g., image vs. text), a set of generator, encoder and classifier must be provided for each modality separately, which complicates the model, but theoretically unavoidable. However, if the properties of multiple modalities are similar

(e.g., image vs. image), we can further reduce the complexity of the model by sharing some model parameters among distinct modalities.

Differences with StarGAN and RadialGAN. StarGAN [41] and RadialGAN [42] are two important methods for multi-source data translation task. In StarGAN, the authors propose a framework for transforming data across multiple categories or attributes. However, StarGAN does not has a share latent space, and only has a general generator, which is not applicable when multiple data sources vary widely. RadialGAN is a framework for multi-source data augmentation. Each data source can be augmented by the other data sources through a shared low-dimensional space. Here the data sources refer to the related but different datasets. The shared space of RadialGAN is only used to align the distribution of different datasets rather than the multiple modalities of the same instance. In contrast, our DSML framework has a modality-shared latent space and multiple modality-specific generators/encoders/classifiers, which inherently supports multimodal semi-supervised classification, missing modality imputation and cross-modality retrieval tasks. Therefore, the proposed DSML framework is different from StarGAN and RadialGAN, both in structures and applications.

4. Experiments

4.1. Datasets

- **RGB-D** [43]. This dataset contains a total of 41, 877 samples, each of which consists of two modalities, RGB and depth image. The data samples were collected from 51 object categories. In the experiments, we first resize the resolution of both modality images to 64 × 64. We then interpolate the missing pixel values in the depth modality using the mean of 5 × 5 nearest pixel values. Finally, the surface normal processing method [44] is used to extend the single-channel depth image to three channels. The authors [43] provide ten different training/test data splits. To simulate semi-supervised learning scenarios, for each data split, we randomly selected 5% samples from each class as labeled data, and the rest as unlabeled data. All methods will be trained on the labeled and unlabeled training data, and evaluated on the test data.
- **MNIST-to-MNIST-transpose** [27,45,46]. This dataset contains 50,000 training samples and 10,000 test samples. Each sample consists of two modalities, one is the original MNIST image and the other is the transpose image.
- **ImageNet-EEG** [47]. This dataset contains Electroencephalogram (EEG) data recorded from six subjects when they are presented visual stimuli. The visual stimuli presented to the subjects were selected from 40 different ImageNet categories, 50 in each category, for a total of 2000 images. After screening by the original author, the dataset contains a total of 11,466 EEG samples, each of which was shaped like [500,128] (500 time points, 128 channel electrodes). Each EEG sample has a corresponding visual stimulus. To simulate the semi-supervised learning scenario, we used a sliding window of shape [200,128] to construct the paired and unpaired EEG data. To obtain the paired EEG data, we intercepted the first 200 time points of all EEG data and obtained the result with a shape of [11466, 200, 128]. To construct a large amount of unpaired EEG data, we applied the above sliding window with 10 different offsets ($\{20, 40, \ldots, 200\}$) to the original EEG sequences. To obtain a large amount of unpaired image data, we selected all the images of the involved category from the ImageNet database as unpaired images. The resolution of all the images used in the experiment was resized to 64 × 64 pixels. For the ImageNet-EEG, 90% of the paired samples were randomly selected as training samples, and the remaining 10% as test data.

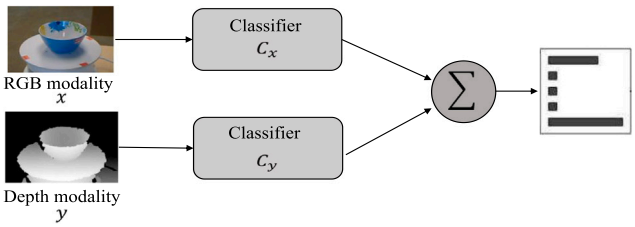


Fig. 8. The pipeline of multimodal semi-supervised classification in DSML. We handle the classification of multi-modality data by combining the results of multiple classifiers, i.e., label = softmax($O_x + O_y$), where O_x and O_y are the outputs before the softmax layer of each classifiers, respectively.

4.2. Experimental settings

We evaluate our approach on the following three different tasks.

- **Multimodal semi-supervised classification** on the RGB-D object dataset.
- **Missing modality imputation** on the MNIST-to-MNIST-transpose dataset and ImageNet-EEG dataset.
- **Cross-modality retrieval** on the ImageNet-EEG dataset.

In the experiments, we set the dimension of the latent variables $\boldsymbol{z}$ to 100, and all $\boldsymbol{z}$ samples are drawn from a standard multivariate Gaussian distribution. The regularization parameter λ_3 is empirically set to $\lambda_3 = 10$, and the rest regularization parameters are all set to 1. In model training, we use the Adam optimizer [48] for parameter optimization, and the learning rate is set to 0.0002. For the comparison methods used in our experiment, we consider the same settings (network architectures, learning rate, etc.) as our method's to make the comparison fair.

4.3. Multimodal semi-supervised classification

It is natural to use the proposed DSML framework for multimodal semi-supervised classification (cf. Fig. 8). DSML enjoys two class-conditional generators G_x and G_y with good controllability, with which, one can synthesize arbitrary number of labeled paired samples to augment the training of classifiers C_x and C_y. Once the classifiers become more accurate, more available labeled samples (by predicting the labels for unlabeled data) can be used to lower the bias brought by the small set of the labeled data, which in return can prevent the generators from collapsing into a biased joint distribution. Consequently, mutual boosting cycle between the generator and classifier is formed for each modality.

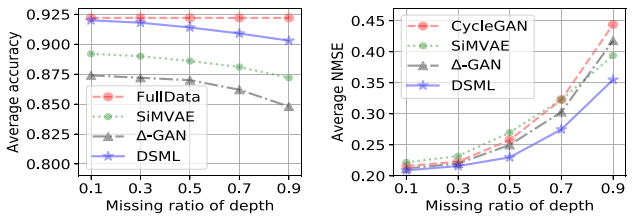
To validate this, we conduct RGB-D object recognition experiments on the RGB-D dataset [43]. Here, we consider three scenarios: (1) both the class labels and the modalities are complete; (2) only the class labels are incomplete, and there are no missing modalities; (3) both the class labels and the modalities are incomplete.

For the first and second scenarios, we compare our DSML with several strong competitors, including both the unimodal and multimodal methods. For unimodal methods, we evaluate their performance on each modality and on the concatenation of two modalities, respectively. The comparisons of classification accuracy on the partially labeled and completely labeled RGB-D dataset are shown in Table 2, in which each result is averaged over the given 10 different test sets. For latent variable based deep generative competitors (M2 [49], SDGM [50], SMVAE [6] and TripleGAN [32]), we set the latent dimensions as 100, which are the same as our method. Particularly, in TripleGAN [32], we set its hyper-parameter $\alpha = 0.5$, which means the relative importance of generation and classification are equal. In SDGM [50], we set the scaling constant $\beta = 0.1$. For the other competitors (CT+SVM [44], DCNN [8] and AMGL [51]), we used their default settings.

Table 2
The classification accuracy (%) on the RGB-D dataset (both modalities are complete).

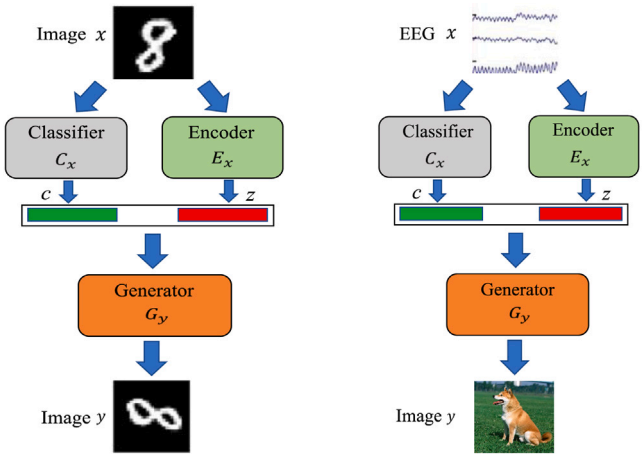
	Algorithms	5% labeled data			100% labeled data		
		RGB	Depth	RGB-D	RGB	Depth	RGB-D
Unimodal baselines	M2 [49]	85.6 ± 1.6	72.0 ± 1.7	86.4 ± 1.6	86.7 ± 1.4	77.3 ± 1.5	88.9 ± 1.5
	SDGM [50]	85.8 ± 1.5	75.4 ± 1.7	86.7 ± 1.5	87.7 ± 1.5	78.6 ± 1.6	89.2 ± 1.6
	TripleGAN [32]	86.4 ± 1.7	82.9 ± 1.8	87.2 ± 1.8	87.9 ± 1.6	84.9 ± 1.7	90.2 ± 1.6
	Δ-GAN [27]	86.5 ± 1.8	82.6 ± 1.9	87.6 ± 1.7	88.2 ± 1.6	84.6 ± 1.8	90.8 ± 1.8
Multimodal baselines	CT+SVM [44]	82.6 ± 1.3	71.4 ± 1.4	83.7 ± 1.3	86.2 ± 1.5	78.6 ± 1.2	88.4 ± 1.1
	DCNN [8]	85.9 ± 0.7	74.0 ± 1.2	89.2 ± 1.3	87.8 ± 1.2	80.3 ± 1.4	91.8 ± 1.2
	AMGL [51]	84.2 ± 2.1	72.4 ± 1.8	86.4 ± 1.5	87.5 ± 1.8	79.8 ± 1.6	91.2 ± 1.3
	SMVAE [6]	85.4 ± 1.4	81.2 ± 1.5	89.5 ± 1.8	88.7 ± 1.6	84.5 ± 1.3	92.3 ± 1.4
Proposed	DSML−$\mathcal{L}_{xc}$−$\mathcal{L}_{yc}$	52.6 ± 3.2	44.8 ± 3.5	57.4 ± 3.4	86.6 ± 1.8	79.4 ± 1.4	88.6 ± 1.5
	DSML−$\mathcal{L}_{xz}$−$\mathcal{L}_{yz}$	77.7 ± 2.5	65.3 ± 2.6	82.5 ± 2.4	87.2 ± 1.4	82.5 ± 1.6	90.4 ± 1.5
	DSML−$\mathcal{R}_z^*$	84.6 ± 2.2	81.7 ± 1.9	88.6 ± 1.8	87.9 ± 1.5	83.2 ± 1.7	91.3 ± 1.5
	DSML	86.4 ± 1.9	83.0 ± 1.8	**92.2**± 1.7	88.4 ± 1.5	84.8 ± 1.6	**92.7**± 1.6

DSML− ∗ means DSML without ∗ term in the objective function.



(a) Classification accuracies (b) Imputation errors

Fig. 9. The comparisons of classification accuracy and modality imputation errors with different missing ratios of the depth modality.



(a) MNIST-to-MNIST-transpose (b) ImageNet-EEG

Fig. 10. The pipelines of missing modality imputation on both datasets. Given one modality $\boldsymbol{x}$, we first use the learnt classifier C_x and encoder E_x to obtain the latent variables $\boldsymbol{c}$ and $\boldsymbol{z}$, and then use the generator G_y to predict the other modality $\boldsymbol{y}$.

From Table 2, we observe that the performance of the proposed DSML significantly surpasses the compared methods in multi-modality setting whether in semi-supervised learning (5% labeled data) or supervised learning (100% labeled data). The reasons are threefold: (1) our method can match the joint distribution of each modality and its labels adversarially; (2) our DSML method effectively captures the high-level common representation of both modality through its shared latent space; (3) our method can synthesize a large number of paired data in training process, which plays the role of data augmentation. In addition, our method nearly achieves the classification performance of 100% labeled data by using only 5% labeled data, which demonstrates the effectiveness of its semi-supervised learning. We also observe that, if $\mathcal{L}_{xc}$ and $\mathcal{L}_{yc}$ are removed the semi-supervised classification ability of the model will be seriously affected; if $\mathcal{L}_{xz}$ and $\mathcal{L}_{yz}$ are removed, the ability of data generation and modality fusion will be seriously affected; and if $\mathcal{R}_z^*$ or $\mathcal{L}_{xy}$ is removed, the cross-modal prediction ability will be affected.

For the third scenario, we randomly select a fraction of samples from all (both labeled and unlabeled) training samples as modality-incomplete samples. Specifically, we assume the selected samples are only with RGB modality, and their depth modality are missing. In the experiment, we varied the missing ratio from 0.1 to 0.9 with an interval of 0.2, and assume the test samples are with complete modalities. We compared our DSML with SiMVAE [6], Δ-GAN, CycleGAN [19] and FullData, where FullData represents the case of DSML with complete modalities. For SiMVAE [6], we set the scaling constants $c_1 = c_2 = 0.5$, and the latent dimension was set to 100. For Δ-GAN and CycleGAN [19], we used their default settings. The comparisons of all methods w.r.t. classification accuracy and modality imputation errors are shown in Fig. 9(a) and (b), respectively. Here the imputation errors are measured by the metric of Normalized Mean Squared Error (NMSE). Assume $\hat{\mathbf{X}}$ denote the imputed result, and $\mathbf{X}$ denote the groundtruth, the NMSE can be calculated by NMSE $= \frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F}$, where $\|\cdot\|_F$ is the Frobenius norm. Although Δ-GAN can be applied to semi-supervised classification or missing modality imputation, it cannot be used to accomplish these two tasks at the same time. Therefore, we first train a Δ-GAN model for missing modality imputation task, and then train another Δ-GAN model to perform semi-supervised classification. By contrast, SiMVAE can be used to solve these two tasks in an end-to-end manner.

From the comparisons in Fig. 9, we observe that the proposed DSML model performs significantly better than the compared methods. When the missing ratio is lower than 0.5, DSML and FullData have very close performance. Even when the missing ratio is higher than 0.5, our DSML method still achieves comparable results to FullData. Another observation is that the semi-supervised methods DSML and SiMVAE achieve better results than the unsupervised methods CycleGAN and Δ-GAN in missing modality imputation task when the missing ratio is very high. This demonstrates that the data label can also play an important role when learning the modality mapping without sufficient paired data. In addition to being able to effectively utilize the label information, our DSML can augment the number of paired data by synthesizing the fake data. Large pairs of synthesized data may also be used to improve model performance, and this is an advantage that is not available in other methods.

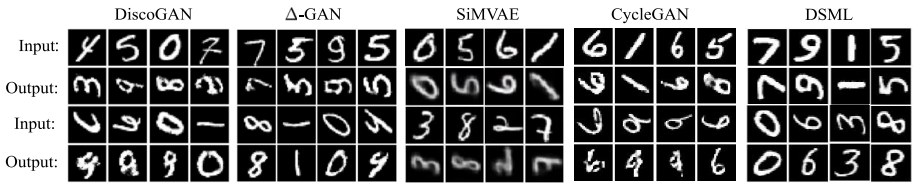


Fig. 11. The results of modality imputation on the test dataset of MNIST-to-MNIST-transpose. We used 10% paired data for our DSML, SiMVAE and Δ-GAN method. Note that DiscoGAN [30] and CycleGAN do not need paired data.

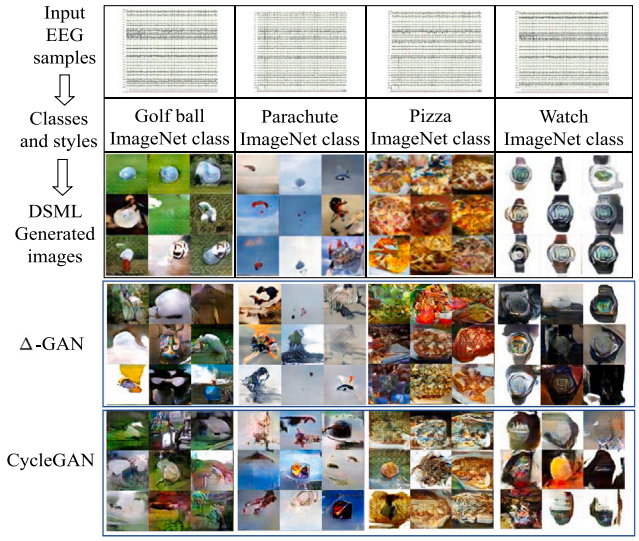


Fig. 12. The results of modality imputation on the test dataset of ImageNet-EEG.

Table 3
The Fréchet Inception Distance (FID) and Inception Score (IS) of the imputated images produced by our DSML and the compared methods on the test dataset of ImageNet-EEG.

Metrics	DSML	Δ-GAN	CycleGAN
FID	32.6 ± 1.5	38.3 ± 1.4	46.4 ± 1.8
IS	7.42 ± 0.06	7.17 ± 0.07	6.63 ± 0.05

4.4. Missing modality imputation

To assess the controllability of our DSML in missing modality imputation, we conduct experiments based on the MNIST-to-MNIST-transpose [27] and ImageNet-EEG [47] datasets (cf. Fig. 10). On both datasets, we simulate the missing modality scenario by removing one modalities for part of training samples. In the training stage, we assume the instances with complete modalities are also with the corresponding class labels, and the instances without complete modalities are also without the corresponding class labels. The experimental results on the MNIST-to-MNIST-transpose dataset are shown in Fig. 11, and the experimental results on the ImageNet-EEG dataset are shown in Fig. 12. For SiMVAE [6], Δ-GAN and CycleGAN [19] on the MNIST-to-MNIST-transpose dataset, their parameter settings are the same as the above. For Δ-GAN and CycleGAN on the ImageNet-EEG dataset, we modified their network architectures to accommodate the non-image (EEG) modality. From these two figures, we can see that the proposed DSML model generally recovers the missing modality better than the compared methods, and its results show good visual quality and strictly follow the intrinsic semantics of the images. Additional EEG-to-image results can be found in Fig. 13.

To evaluate the experimental results quantitatively, we first compute the Fréchet Inception Distance (FID) and Inception Score (IS) of the imputated images produced by our DSML and the compared methods on the test dataset of ImageNet-EEG in Table 3. The IS highly correlates with human judgment, which allows us to avoid relying on human evaluations. From Table 3, we see that DSML achieves better FID and IS than the compared methods. Then, we use a pre-trained classifier as the gold-standard tool to classify the imputed images. When calculating the accuracy, the ground truth are the labels of test data. The classification accuracy of the gold-standard classifier approaches 99.4% on the test set of MNIST. Therefore, It is trustworthy to evaluate

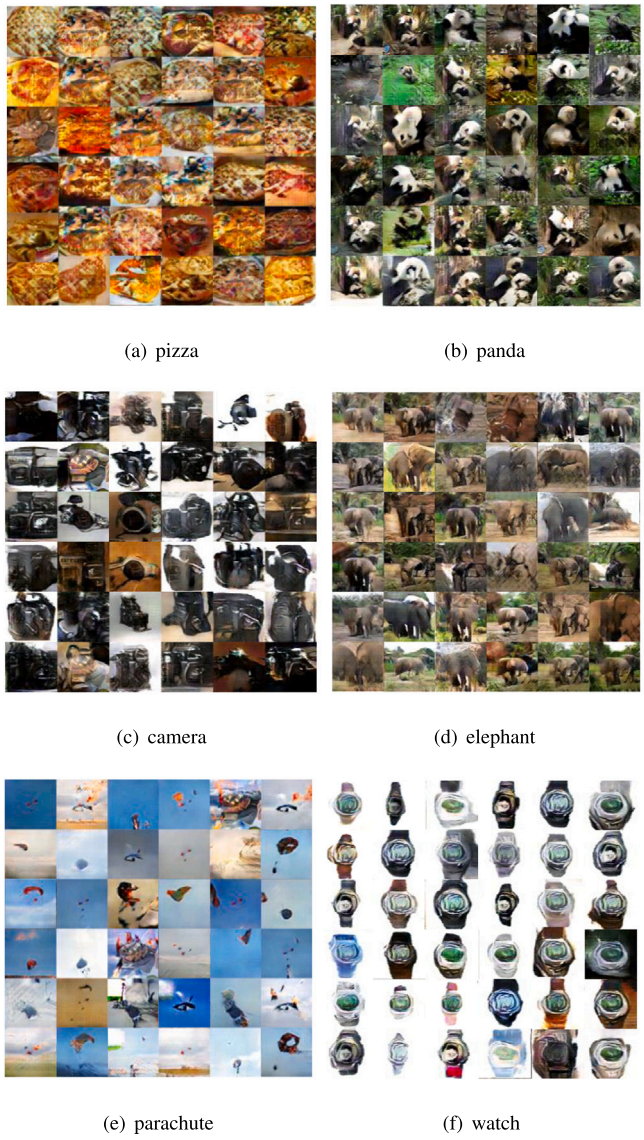
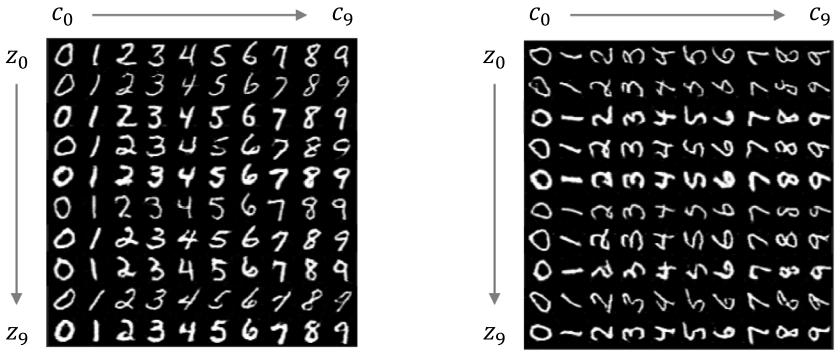


Fig. 13. Different categories of images inferred from EEG on the ImageNet-EEG test set using our DSML method.



(a) Generated modality $\boldsymbol{x}$ (b) Generated modality $\boldsymbol{y}$

Fig. 14. The generated data pairs on the MNIST-to-MNIST-transpose dataset.

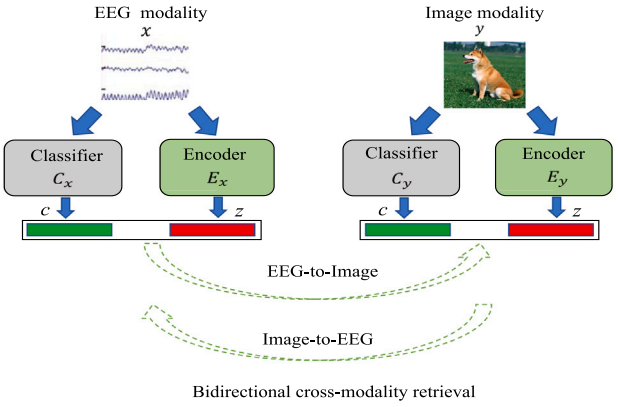
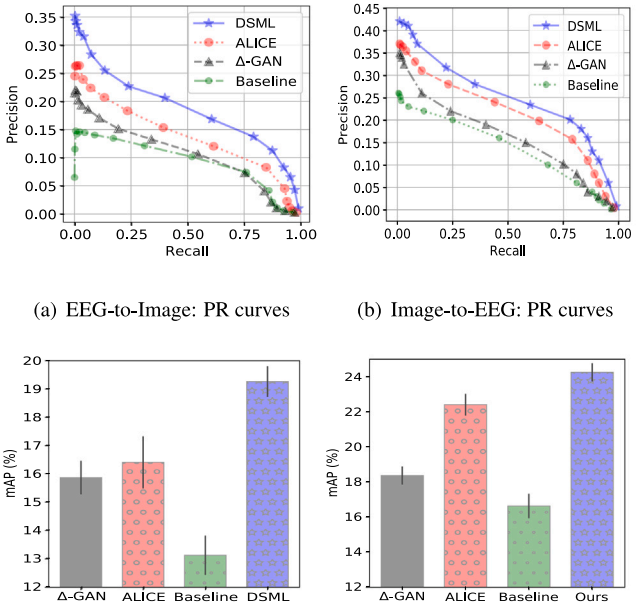


Fig. 15. The pipeline of cross-modality retrieval on the ImageNet-EEG dataset. Given the modalities $\boldsymbol{x}$ and $\boldsymbol{y}$, we first use the learnt classifier C_x (or C_y) and encoder E_x (or E_y) to obtain the latent variables $\boldsymbol{c}$ and $\boldsymbol{z}$. Then, the bidirectional cross-modality retrieval experiments are performed in the modality-shared latent space based on $\boldsymbol{c}$ and $\boldsymbol{z}$.



(a) EEG-to-Image: PR curves (b) Image-to-EEG: PR curves

(c) EEG-to-Image: mAP (d) Image-to-EEG: mAP

Fig. 16. Cross-modality retrieval results on the ImageNet-EEG dataset. Each result in (a), (b), (c) and (d) was averaged over six subjects. The vertical lines on the histogram in (c) and (d) indicate the error bars across different subjects.

the results. For ImageNet-EEG, we choose the pre-trained Inception-v3 model [52] as the gold-standard classifier. The quantitative results are shown in Table 4. From the results, we see that the proposed DSML model outperforms the state-of-the-art methods Triple GAN and Δ-GAN. These results indicate that DSML have good ability to control the semantics of imputed results. Furthermore, we also see that the adversarial game $\mathcal{L}_{xy}$ can effectively improve DSML's performance.

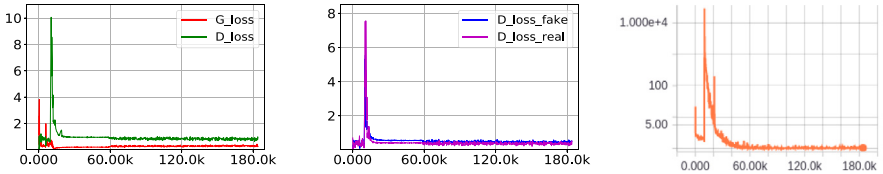
We also evaluated DSML with the additional cycle consistency term $\mathcal{R}_{xy}$, and find that the difference between them (with or without $\mathcal{R}_{xy}$) is not so obvious (80.95% on the ImageNet-EEG dataset, and 98.82%, 99.04%, 99.21% on the 100 paired, 1000 paired, all paired MNIST-to-MNIST-transpose dataset, respectively). In other words, the cycle consistency constraint does not lead to conspicuous performance gains, especially if there are enough paired samples. Considering the complexity of the DSML model, we do not use the cycle consistency constraint by default.

Table 4
Classification accuracy (%) of the imputed images. The results are averaged over five runs with different random data splits. DSML−$\mathcal{L}_{xy}$ means DSML without $\mathcal{L}_{xy}$.

Algorithms	MNIST-to-MNIST-transpose			ImageNet-EEG
	#100 paired	#1000 paired	All paired	90% paired
DiscoGAN	–	–	15.00 ± 0.20	–
CycleGAN	76.85 ± 2.02	85.46 ± 1.89	91.74 ± 1.66	58.75 ± 2.43
Δ-GAN	83.20 ± 1.88	88.98 ± 1.50	93.34 ± 1.46	66.02 ± 1.09
SiMVAE	90.38 ± 2.03	94.98 ± 1.69	95.17 ± 1.82	69.33 ± 1.15
DSML−$\mathcal{L}_{xy}$	98.21 ± 1.71	98.45 ± 1.64	98.66 ± 1.61	76.68 ± 1.34
DSML	**98.67** ± 1.43	**99.02** ± 1.41	**99.23** ± 1.30	**81.24** ± 0.98

4.5. Disentanglement of $\boldsymbol{c}$ and $\boldsymbol{z}$

To demonstrate that $\boldsymbol{c}$ and $\boldsymbol{z}$ are disentangled from each other after the training of DSML, we generate images with different combinations of $\boldsymbol{c}$ and $\boldsymbol{z}$ on the MNIST-to-MNIST-transpose dataset. The results are shown in Fig. 14. Clearly, the two generated modalities are semantically consistent with $\boldsymbol{c}$, and change their styles as $\boldsymbol{z}$ changes. For example, $\boldsymbol{z}_8$ encodes the slope information (slant to the right), while $\boldsymbol{z}_9$ encodes the bold information. This verifies that DSML correctly disentangles the semantic information and other information.



(a) Loss of G and D (b) Real vs. Fake (c) Loss of C

Fig. 17. Illustration of the training process. The vertical axis represents loss, while the horizontal axis represents the global steps. From (b), we observe that the two curves are difficult to separate from each other after 60,000 steps, which indicates that the real and fake data have similar distributions.

4.6. Cross-modality retrieval

DSML has a modality-shared low-dimensional latent space, and we can perform cross-modality retrieval in that space (cf. Fig. 15). Recall that DSML's latent representation contains two parts of variables, i.e., the semantic label c and the semantic-free code z. Here the cross-modality retrieval experiments are performed on the ImageNet-EEG dataset by using both c and z. Specifically, we conduct two different cross-modality retrieval tasks: (1) using the given EEG samples to retrieval the corresponding visual images (EEG-to-Image) and (2) using the given image samples to retrieval the corresponding EEG signals (Image-to-EEG).

We first randomly draw five samples from each class of the test dataset as queries. For each selected query, we further get the corresponding c and z by using the trained classifier and encoder, respectively. Based on the predicted c and z, we then find its $n \in \{2^0, 2^1, \ldots, 2^{15}\}$ nearest neighbors by using similarity search. The corresponding images/EEGs of these neighbors are returned as the cross-modality retrieval results. In similarity search, we adopt a two-step strategy. In the first step, we match the semantic label c with all candidates. In the second step, we perform ranking w.r.t. z based on the Euclidean distance. Note that, in the ranking, c matched samples are ahead of c unmatched samples. We plot the precision–recall (PR) curves to evaluate the retrieval performances of different methods (in Fig. 16(a) and (b)). Here, we set the number of relevant instance as 100. In addition, we use the mean average precision (mAP) as another metric (in Fig. 16(c) and (d)). Note that the mAP value represents the area under the PR curve and it reflects the overall retrieval performance. The formula of AP is defined as $\mathrm{AP} = \frac{1}{T}\sum_{n=1}^{N}\frac{T_n}{n}\times \mathrm{rel}(n)$, where T is the number of relevant instances in the test dataset (here, $T = 100$), N is the total number of instances, T_n is the number of relevant instance in top n returned results, $\mathrm{rel}(n)$ is an indicator whose value is 1 if the rank n of the returned results is a relevant instance and 0 otherwise. Since mAP is the mean value of AP for each query, hence it is defined as $\mathrm{mAP} = \frac{1}{Q}\sum_{q=1}^{Q}\mathrm{AP}(q)$, where Q is the number of queries (here, $Q = 200$).

From Fig. 16, we can see that the joint distribution matching methods DSML, Δ-GAN and ALICE [24] consistently outperform the baseline method. The baseline is constructed as follow: given an query sample, we first find its nearest neighbor based on the Euclidean distance in the corresponding data space, and then we perform image/EEG retrieval using the corresponding image/EEG of that nearest neighbor EEG/image sample. In particular, the proposed DSML model achieved better performance than Δ-GAN and ALICE. The result was caused by the fact that our DSML model can augment the training data by synthesizing lots of paired data, which facilitates the learning of modality mappings. Last but not least, Δ-GAN and ALICE methods can only conduct inefficient cross-modality retrieval in the original high-dimensional data space. By contrast, our DSML model performs the cross-modality retrieval task in the modality-shared latent space. This is more efficient and practical for high-dimensional data.

4.7. Convergence and stability of DSML

Fig. 17 illustrates the convergence and stability of the DSML training on the RGB-D object dataset. Empirically, we find that DSML works well in practice. The reason for the front part of the curve in (c) fluctuate significantly is that the large number of the generated data $p_{g_x}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})$ and $p_{g_y}(\boldsymbol{y}|\boldsymbol{c},\boldsymbol{z})$ are used to augment the training of classifiers C_x and C_y, respectively. In the initial stages of model training, the pretrained classifiers C_x and C_y were prone to misdirected by the generated data, whose quality were not good enough. But with the improvement of image quality, the classifiers converge rapidly and stably.

5. Conclusion

We presented a unified framework for joint classification, generation and retrieval through doubly semi-supervised multimodal adversarial learning. The proposed DSML framework consists of four kinds of components: generators, encoders, classifiers and discriminators, all jointly trained via adversarial learning. With a modality-shared latent space, DSML enjoys many advantages, such as accurately controlling the semantics of imputed modalities, augmenting the training set with synthesized samples and scaling well to multiple modalities. In the experiments, we have shown that DSML can be applied to a wide range of applications. The experimental results on multiple datasets demonstrate that DSML as a unified framework can simultaneously (1) achieve the state-of-the-art multimodal semi-supervised classification results; (2) recover the missing modality with high visual quality and correct intrinsic semantics; and (3) perform efficient cross-modality retrieval in the modality-shared latent space.

CRediT authorship contribution statement

Changde Du: Conceptualization, Methodology, Software, Writing - original draft, Visualization, Investigation. **Changying Du:** Conceptualization, Software, Formal analysis, Writing - review & editing. **Huiguang He:** Project conecpt design, Revise and proof the paper, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61976209, Grant 62020106015, and Grant 61906188; in part by the Chinese Academy of Sciences (CAS) International Collaboration Key, China, Project under Grant 173211KYSB20190024; and in part by the Strategic Priority Research Program of CAS, China, under Grant XDB32040000.

Appendix A. Proof of Proposition 2

Supposing $f_1, \dots, f_K$ are K variables, let us first consider a general optimization problem as the following

$$\min \mathcal{L}(f_1, \dots, f_K) = \sum_{k=1}^{K} p_k(\boldsymbol{x}, \boldsymbol{c}) \log \frac{f_k}{\sum_{i=1}^{K} f_i}.$$

For any f_k, if we fix all other variables, and let

$$\frac{\partial \mathcal{L}}{\partial f_k} = \frac{p_k(\boldsymbol{x}, \boldsymbol{c})}{f_k} - \sum_{j=1}^{K} \frac{p_j(\boldsymbol{x}, \boldsymbol{c})}{\sum_{i=1}^{K} f_i} = 0,$$

we can obtain the optimal f_k^* for $k = 1, \dots, K$:

$$f_k = p_k(\boldsymbol{x}, \boldsymbol{c}) \frac{\sum_{i=1, i\neq k}^{K} f_i}{\sum_{j=1, j\neq k}^{K} p_j(\boldsymbol{x}, \boldsymbol{c})} = C \cdot p_k(\boldsymbol{x}, \boldsymbol{c}),$$

where $C \neq 0$ is a constant. Let $\hat{f}_k = \frac{f_k}{\sum_{i=1}^{K} f_i}$, the global optimal is achieved at $\hat{f}_k = \frac{p_k(\boldsymbol{x}, \boldsymbol{c})}{\sum_{j=1}^{K} p_j(\boldsymbol{x}, \boldsymbol{c})}$.

Let $K = 3$ and $\hat{f}_k = D_{xc}(\boldsymbol{x}, \boldsymbol{c})[k]$. This indicates that with fixed G_x and C_x, the optimal discriminator $D_{xc}(\boldsymbol{x}, \boldsymbol{c})$ in the main text is achieved at

$$D_{xc}^*(\boldsymbol{x}, \boldsymbol{c})[k] = \frac{p_k(\boldsymbol{x}, \boldsymbol{c})}{\sum_{j=1}^{K} p_j(\boldsymbol{x}, \boldsymbol{c})}.$$

With optimal $D_{xc}^*(\boldsymbol{x}, \boldsymbol{c})[k]$, the objective (7) in the main text can be expressed as

$$\begin{aligned}
\mathcal{L}_{xc} &= \sum_{k=1}^{3} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{c}) \sim p_k(\boldsymbol{x}, \boldsymbol{c})} \log \frac{p_k(\boldsymbol{x}, \boldsymbol{c})}{\sum_{j=1}^{3} p_j(\boldsymbol{x}, \boldsymbol{c})} \\
&= -3 \log 3 + \sum_{k=1}^{3} \mathrm{KL}\left(p_k(\boldsymbol{x}, \boldsymbol{c}) \middle\| \frac{\sum_{j=1}^{3} p_j(\boldsymbol{x}, \boldsymbol{c})}{3} \right) \\
&= -3 \log 3 + 3 \cdot JSD\Big(p_1(\boldsymbol{x}, \boldsymbol{c}), p_2(\boldsymbol{x}, \boldsymbol{c}), p_3(\boldsymbol{x}, \boldsymbol{c}) \Big) \\
&\geq -3 \log 3
\end{aligned}$$

where $JSD_{\pi_1, \dots, \pi_K}(p_1, p_2, \dots, p_K) = H\left(\sum_{j=1}^{K} \pi_j p_j \right) - \sum_{j=1}^{K} \pi_j H(p_j)$ is the Jensen–Shannon divergence, which is always non-negative, and zero only when the probability distribution $p_1, p_2, \dots, p_K$ are equal. Here, $\pi_1, \dots, \pi_K$ are weights that are selected for $p_1, p_2, \dots, p_K$, and $H(p_j)$ is the entropy for distribution p_j. In the three-distribution case described above, we set $K = 3$ and $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$.

Therefore, the global minimum of (7) is achieved at $p_1(\boldsymbol{x}, \boldsymbol{c}) = p_2(\boldsymbol{x}, \boldsymbol{c}) = p_3(\boldsymbol{x}, \boldsymbol{c})$ with the optimal $D_{xc}^*(\boldsymbol{x}, \boldsymbol{c})[k] = \frac{1}{3}$, and the optimum value is $-3 \log 3$. The proof for $\mathcal{L}_{yc}$ is similar.

Appendix B. Proof of Proposition 3

Based on the proof of Proposition 2, let $K = 4$ and $\hat{f}_k = D_{xy}(\boldsymbol{x}, \boldsymbol{y})[k]$, the optimal discriminator $D_{xy}(\boldsymbol{x}, \boldsymbol{c})$ is achieved at

$$D_{xy}^*(\boldsymbol{x}, \boldsymbol{y})[k] = \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1}^{K} p_j(\boldsymbol{x}, \boldsymbol{y})}.$$

With optimal $D_{xy}^*(\boldsymbol{x}, \boldsymbol{y})[k]$, the objective (12) in the main text can be expressed as

$$\begin{aligned}
\mathcal{L}_{xy} &= \sum_{k=1}^{4} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim p_k(\boldsymbol{x}, \boldsymbol{y})} \log \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1}^{4} p_j(\boldsymbol{x}, \boldsymbol{y})} \\
&= -4 \log 4 + \sum_{k=1}^{4} \mathrm{KL}\left(p_k(\boldsymbol{x}, \boldsymbol{y}) \middle\| \frac{\sum_{j=1}^{4} p_j(\boldsymbol{x}, \boldsymbol{y})}{4} \right) \\
&= -4 \log 4 + 4 \cdot JSD\Big(p_1(\boldsymbol{x}, \boldsymbol{y}), p_2(\boldsymbol{x}, \boldsymbol{y}), p_3(\boldsymbol{x}, \boldsymbol{y}), p_4(\boldsymbol{x}, \boldsymbol{y}) \Big) \\
&\geq -4 \log 4
\end{aligned}$$

Therefore, the global minimum of (12) is achieved at $p_1(\boldsymbol{x}, \boldsymbol{y}) = p_2(\boldsymbol{x}, \boldsymbol{y}) = p_3(\boldsymbol{x}, \boldsymbol{y}) = p_4(\boldsymbol{x}, \boldsymbol{y})$ with the optimal $D_{xy}^*(\boldsymbol{x}, \boldsymbol{y})[k] = \frac{1}{4}$, and the optimum value is $-4 \log 4$.

References

[1] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: A survey and taxonomy, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2) (2018) 423–443.
[2] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: Recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.
[3] J. Zhang, Z. Yin, P. Chen, S. Nichele, Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review, Inf. Fusion 59 (2020) 103–126.
[4] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A review of affective computing: From unimodal analysis to multimodal fusion, Inf. Fusion 37 (2017) 98–125.
[5] W.L. Zheng, W. Liu, Y. Lu, B.L. Lu, A. Cichocki, EmotionMeter: A multimodal framework for recognizing human emotions, IEEE Trans. Cybern. (2018) 1–13.
[6] C. Du, C. Du, H. Wang, J. Li, W.-L. Zheng, B.-L. Lu, H. He, Semi-supervised deep generative modelling of incomplete multi-modality emotional data, in: Proc. ACM MM, 2018.
[7] A. Wang, J. Lu, J. Cai, T.J. Cham, G. Wang, Large-margin multi-modal deep learning for RGB-D object recognition, IEEE Trans. Multimed. 17 (11) (2015) 1887–1898.
[8] Y. Cheng, X. Zhao, R. Cai, Z. Li, K. Huang, Y. Rui, Semi-supervised multimodal deep learning for RGB-D object recognition, in: Proc. IJCAI, 2016.
[9] Q. Wang, M. Sun, L. Zhan, P. Thompson, S. Ji, J. Zhou, Multi-modality disease modeling via collective deep matrix factorization, in: Proc. SIGKDD, 2017.
[10] S. Zhe, Z. Xu, Y. Qi, P. Yu, Sparse Bayesian multiview learning for simultaneous association discovery and diagnosis of Alzheimer's disease, in: Proc. AAAI, 2015.
[11] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proc. ICML, 2011.
[12] G. Andrew, R. Arora, J.A. Bilmes, K. Livescu, Deep canonical correlation analysis, in: Proc. ICML, 2013.
[13] W. Wang, R. Arora, K. Livescu, J.A. Bilmes, On deep multi-view representation learning, in: Proc. ICML, 2013.
[14] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, M. Shah, Brain2Image: Converting brain signals into images, in: Proc. ACM MM, 2017.
[15] C. Du, C. Du, L. Huang, H. He, Reconstructing perceived images from human brain activities with Bayesian deep multiview learning, IEEE Trans. Neural Netw. Learn. Syst. 30 (8) (2018) 2310–2323.
[16] T. Luan, X. Liu, J. Zhou, R. Jin, Missing modalities imputation via cascaded residual autoencoder, in: Proc. CVPR, 2017.
[17] C. Du, C. Du, X. Xie, C. Zhang, H. Wang, Multi-view adversarially learned inference for cross-domain joint distribution matching, in: Proc. SIGKDD, 2018.
[18] L. Cai, Z. Wang, H. Gao, D. Shen, S. Ji, Deep adversarial learning for multi-modality missing data completion, in: Proc. SIGKDD, 2018.
[19] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proc. ICCV, 2017.
[20] C. Du, C. Du, H. He, Doubly semi-supervised multimodal adversarial learning for classification, generation and retrieval, in: Proc. ICME, 2019.
[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proc. NeurIPS, 2014.
[22] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, A. Courville, Adversarially learned inference, in: Proc. ICLR, 2017.
[23] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, in: Proc. ICLR, 2017.
[24] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, L. Carin, Alice: Towards understanding adversarial learning for joint distribution matching, in: Proc. NeurIPS, 2017.
[25] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proc. CVPR, 2017.
[26] B. Lei, Z. Xia, F. Jiang, X. Jiang, Z. Ge, Y. Xu, J. Qin, S. Chen, T. Wang, S. Wang, Skin lesion segmentation via generative adversarial networks with dual discriminators, Med. Image Anal. (2020) 101716.
[27] Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, L. Carin, Triangle generative adversarial networks, in: Proc. NeurIPS, 2017.
[28] Y. Pu, S. Dai, Z. Gan, W. Wang, G. Wang, Y. Zhang, R. Henao, L. Carin, JointGAN: Multi-domain joint distribution learning with generative adversarial nets, in: Proc. ICML, 2018.
[29] M. Wu, N. Goodman, Multimodal generative models for scalable weakly-supervised learning, in: Proc. NeurIPS, 2018, pp. 5575–5585.
[30] T. Kim, M. Cha, H. Kim, J. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in: Proc. ICML, 2017.
[31] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.
[32] C. Li, K. Xu, J. Zhu, B. Zhang, Triple generative adversarial nets, in: Proc. NeurIPS, 2017.
[33] Z. Deng, H. Zhang, X. Liang, L. Yang, S. Xu, J. Zhu, E.P. Xing, Structured generative adversarial networks, in: Proc. NeurIPS, 2017.
[34] S. Wu, G. Deng, J. Li, R. Li, Z. Yu, H.-S. Wong, Enhancing TripleGAN for semi-supervised conditional instance synthesis and classification, in: Proc. CVPR, pp. 10091–10100.

[35] S. Wang, X. Wang, Y. Hu, Y. Shen, Z. Yang, M. Gan, B. Lei, Diabetic retinopathy diagnosis using multichannel generative adversarial network with semisupervision, IEEE Trans. Autom. Sci. Eng. (2020) 2981637.
[36] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: Proc. ICLR, 2014.
[37] M. Suzuki, Y. Matsuo, Semi-supervised multimodal learning with deep generative models, in: Proc. ICLR, Workshop, 2018.
[38] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, in: Proc. ICLR, 2018.
[39] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, 2017, arXiv preprint arXiv:1701.07875.
[40] J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial network, in: Proc. ICLR, 2016.
[41] Y. Choi, M. Choi, M. Kim, StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation, in: Proc. CVPR, 2016.
[42] J. Yoon, J. Jordon, M. van der Schaar, RadialGAN: Leveraging multiple datasets to improve target-specific predictive models using Generative Adversarial Networks, in: Proc. ICML, 2018.
[43] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view RGB-D object dataset, in: ICRA, 2011, pp. 1817–1824.
[44] Y. Cheng, X. Zhao, K. Huang, T. Tan, Semi-supervised learning and feature evaluation for RGB-D object recognition, Comput. Vis. Image Underst. 139 (C) (2015) 149–160.
[45] S. Karatsiolis, C.N. Schizas, N. Petkov, Modular domain-to-domain translation network, Neural Comput. Appl. 32 (11) (2020) 6779–6791.
[46] R. Kuznetsova, O. Bakhteev, A. Ogaltsov, Variational learning across domains with triplet information, in: Proc. NeurIPS, 2018.
[47] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, M. Shah, Deep learning human mind for automated visual classification, in: Proc. CVPR, 2017.
[48] D. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[49] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Proc. NeurIPS, 2014.
[50] L. Maaløe, C.K. Sønderby, S.K. Sønderby, O. Winther, Auxiliary deep generative models, in: Proc. ICML, 2016.
[51] F. Nie, J. Li, X. Li, et al., Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification, in: Proc. IJCAI, 2016.
[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. CVPR, 2016.