This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

Deep Generative Models for Weakly-Supervised Multi-Label Classification

Hong-Min Chu[1], Chih-Kuan Yeh[2], and Yu-Chiang Frank Wang[1]

[1] College of EECS, National Taiwan University, Taipei, Taiwan
{r04922031,ycwang}@ntu.edu.tw
[2] Machine Learning Department, Carnegie Mellon University, Pittsburgh, USA
cjyeh@cs.cmu.edu

Abstract. In order to train learning models for multi-label classification (MLC), it is typically desirable to have a large amount of fully annotated multi-label data. Since such annotation process is in general costly, we focus on the learning task of weakly-supervised multi-label classification (WS-MLC). In this paper, we tackle WS-MLC by learning deep generative models for describing the collected data. In particular, we introduce a sequential network architecture for constructing our generative model with the ability to approximate observed data posterior distributions. We show that how information of training data with missing labels or unlabeled ones can be exploited, which allows us to learn multi-label classifiers via scalable variational inferences. Empirical studies on various scales of datasets demonstrate the effectiveness of our proposed model, which performs favorably against state-of-the-art MLC algorithms.

Keywords: Multi-label classification, generative models, semi-supervised learning, weakly-supervised learning

1 Introduction

Multi-label classification (MLC) solves the problem of assigning multiple labels to a single input instance, which has been seen in a variety of applications in the fields of machine learning, computer vision, data mining, and bio-informatics [28, 2, 9].

Like most classification algorithms, one typically needs a large number of data with ground truth labels, so that the associated MLC model can be learned with satisfactory performance. However, for the task of MLC, collecting fully annotated data would take extensive efforts and costs. How to alleviate the above limitation for designing effective MLC models becomes a challenging yet practical task. To be more specific, it would be desirable to train MLC models using training data with only *partial* labels, or even some training data with *empty* label sets observed. Thus, learning MLC models under the above settings can be formalized as a weakly-supervised setting. The differences between weakly-supervised MLC and related MLC settings are summarized in Table 1. The goal of this paper is to present an effective weakly-supervised MLC (WS-MLC) model by advancing deep learning techniques.

A number of MLC approaches which utilize partially labeled data exist (i.e., some training data are only with partial ground truth label information observed) [31, 32,

Setting	fully-labeled data	partially-labeled data	unlabeled data
Supervised MLC	✓	×	×
Semi-supervised MLC	✓	×	✓
MLC with missing label	✓	✓	×
WS-MLC (Our work)	✓	✓	✓

Table 1: Different Settings for multi-label classification.

35, 13]. As a representative work, [31] handles missing labels by imposing a label smoothness regularization during the learning of their model. However, this type of approaches cannot easily leverage rich information from unlabeled training data, which might not be desirable in practical scenarios in which a majority of collected training data are totally unlabeled.

To address the above challenging (semi-supervised) MLC problems, graph-based [37] approaches are proposed [18, 5, 20, 33, 14]. While they exhibit impressive abilities in handling unlabeled data, take label propagation based algorithms [18, 5, 20] for example, they only work under the transductive setting but not the inductive setting. That is, prediction can only be made for the presented unlabeled data but *not* for future test inputs. Another family of manifold regularization based algorithms [33, 14], while applicable for inductive settings, are highly sensitive to graph structures and the associated distance measurements.

Deep generative models, on the other hand, have recently been widely applied to solving semi-supervised learning tasks [16, 24]. Take [16] as an example, it described a deep generative model for single-label data, and applied variational inference for semi-supervised learning via observing both labeled and unlabeled data. Nevertheless, despite the compelling probabilistic interpretation of observed data, existing works mainly apply deep generative models for single label learning tasks. While generative approaches for MLC have been investigated in literature [13, 23, 29], existing solutions typically require training data to be fully or at least partially labeled. In other words, they cannot be easily extended to solving semi-supervised MLC or even WS-MLC tasks.

In this paper, we tackle the challenging WS-MLC, which includes both semi-supervised MLC and MLC with missing labels as *special cases* as illustrated in Table 1. We achieve so by advancing novel deep generative models [16, 17]. Inspired by [22, 8, 30, 21], we approach WS-MLC by viewing MLC as a *sequential prediction* problem. We propose a *deep sequential generative model* to describe the multi-label data for WS-MLC with a unified probabilistic framework. In our proposed model, we present a *deep sequential classification model* for both prediction and approximation of posterior inference, and derive efficient learning algorithms with variational inference for addressing WS-MLC with promising performances.

The contributions of this paper are highlighted as follows:

- To the best of our knowledge, we are the first to advance deep generative models to tackle WS-MLC problems.
- We propose a probabilistic framework which integrates sequential prediction and generation processes with an efficient optimization procedure, so that information from unlabeled data or data with partially missing labels can be exploited.
- Our framework results in interpretable MLC models in weakly-supervised settings, and performs favorably against recent MLC approaches on multiple datasets.

2 Related Works

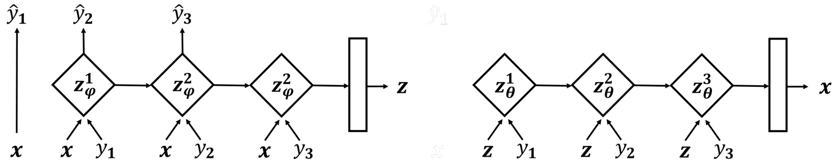
Multi-label classification (MLC) is among active research topics and benefits a variety of real-world applications [2, 9, 7]. Earlier studies of MLC algorithms typically utilize linear models as the building block [28, 22, 26]. Binary relevance [28], as a well-known example, trains a set of independent linear classifiers for each label.

In recent years, approaches based on deep neural networks (DNN) [30, 34, 10, 21] attract the attention of researchers in related fields. For example, [30] proposes to learn a linear embedding function, with label correlations modeled with a chain structure by recurrent neural networks (RNN). [21] further investigates different exploitation of RNN to perform MLC. On the other hand, [34] proposes to learn nonlinear embedding via deep canonical correlation analysis, while it decodes outputs labels with co-occurrence information preserved. Nevertheless, despite the success in applying DNN for MLC, training DNNs typically requires a large amount of labeled data, whose annotation process generally requires extensive manual efforts.

Weakly-supervised MLC (WS-MLC) is a practical setup that aims to learn MLC models from the dataset containing *fully-labeled* plus *partially-labeled* and/or *unlabeled* training data. As noted above, WS setting is particularly appealing regarding MLC tasks, as the cost to fullying annotate multi-label data is generally much more expensive than that for single-label data. MLC algorithms designed for dealing with partially-labeled data (or data with missing labels) exist [31, 32, 35]. For example, [32] formulates the problem of MLC with partially-labeled data as a convex quadratic optimization problems. [31] handles the missing labels by imposing a label smoothness regularization. Unfortunately, both [32] and [31] work only under transductive setting, i.e., the data to be predicted need to be presented during learning. While inductive MLC algorithms with missing labels are available [35], their incapability of exploiting information unlabeled data still makes them less desirable for practical scenarios.

Semi-supervised MLC algorithms which leverage information from both fully-labeled and unlabeled data have also been studied [18, 5, 20, 33, 14]. The majority of such methods focus on graph-based techniques to utilize the unlabeled data. Several graph-based algorithms consider label propagation techniques [18, 5, 20]. [18] is a representative example which designs a dynamic propagation procedures that explicitly considers the label correlation based on k-nearest-neighbors graph. Other graph-based algorithms exploit the information of unlabeled data by manifold regularization [33, 14]. For example, [14] imposes manifold regularization during the learning of MLC models by enforcing similar predictions for both labeled and unlabeled data that is also similar in feature space. Nevertheless, most label propagation based algorithms also require a transductive setting, limiting their applicability to real-wolrd scenarios. Manifold regularization based approaches are mainly inductive. However, the performance of these approaches critically depends on the predefined graph structures. Moreover, all the above semi-supervised MLC algorithms fail to generalize to handle data with partially observed labels.

We note that, generative leaning algorithms for semi-supervised single-label classification can be found in recent literature [1, 16]. Focusing on the task of MLC, several generative approaches have also been investigated [13, 23]. For example, [23] focuses on the mining of multi-labeled text data, where the data generative process is formulated



(a) (b)

Fig. 1: Architectures of (a) encoder $\boldsymbol{\phi}$ and (b) decoder $\boldsymbol{\theta}$ in our DSGM. The former sequentially encodes the input data and their labels into each stochastic variable $\mathbf{z}_{\boldsymbol{\phi}}^k$, and predicts the following label $\hat{y}_k$ of the k-th label conditioned on $\mathbf{x}$ and $y_1, \cdots, y_{k-1}$. The latter decodes $\mathbf{z}$ to $\mathbf{x}$ by sequentially incorporating each label y_k into stochastic variables $\mathbf{z}_{\boldsymbol{\theta}}^k$. Note that we take three labels y_1, y_2 and y_3 for illustration purposes.

based on Latent Dirichlet Analysis. Nevertheless, the above algorithms are not designed to handle tarining data with missing labels or unlabeled training data, and thus cannot be easily extended to WS-MLC. In the next section, we will introduce our proposed deep generative model for WS-MLC.

3 Our Proposed Method

3.1 Problem Formulation

In multi-label classification (MLC), we denote $\mathbf{x} \in \mathbb{R}^d$ as an instance with $\mathbf{y} \in \{0, 1\}^K$ as the corresponding label vector (i.e., $\mathbf{y}[k] = 1$ if the instance is associated with the k-th label (out of K labels), otherwise $\mathbf{y}[k] = 0$). For weakly-supervised MLC (WS-MLC), we observe a training dataset $\mathcal{D} = \mathcal{D}_\ell \cup \mathcal{D}_o \cup \mathcal{D}_u$, where $\mathcal{D}_\ell = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_\ell}$ denotes fully-labeled N_ℓ instances, $\mathcal{D}_o = \{(\mathbf{x}_j, \mathbf{y}_j^{\mathbf{o}})\}_{j=1}^{N_o}$ is the partially-labeled dataset with N_o instances, and $\mathcal{D}_u = \{\tilde{\mathbf{x}}_m\}_{m=1}^{N_u}$ is the unlabeled one with N_u instances. We use $\mathbf{y}^{\mathbf{o}}$ to indicates the partially labeled vector (see detailed settings in experiments). For the sake of simplicity, we omit the subscripts i, j and m if possible in the remaining of this paper. And, we use the term "weakly-labeled" when referring to a subset of training data that is either partially-labeled or unlabeled.

Now, given a training set $\mathcal{D}$, the goal of WS-MLC is to learn a classification model so that the multi-label vector $\hat{\mathbf{y}}$ of an *unseen* instance $\hat{\mathbf{x}}$ can be predicted. In WS-MLC, the size of fully-labeled dataset is typically much smaller than that of weakly-labeled dataset. Therefore, an effective WS-MLC algorithm to exploit the information from both $\mathcal{D}_o$ and $\mathcal{D}_u$ would be desirable, so that improved MLC performance can be expected.

3.2 Deep Sequential Generative Models for WS-MLC

Inspired by recent advances in deep generative models (particularly those for semi-supervised learning [16, 17]) and the use of sequential learning models for MLC [22, 8, 30, 21], we propose a novel **Deep Sequential Generative Model** (DSGM) to tackle the challenging problem of WS-MLC. As illustrated in Fig. 1, our DGSM can be viewed as

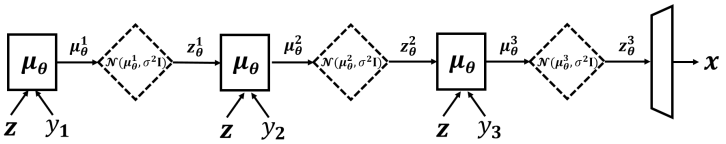

$\boldsymbol{\mu}_{\boldsymbol{\theta}}$
$\boldsymbol{\mu}_{\boldsymbol{\theta}}^1$
$\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}^1, \sigma^2\mathbf{I})$
$\mathbf{z}_{\boldsymbol{\theta}}^1$
$\boldsymbol{\mu}_{\boldsymbol{\theta}}$
$\boldsymbol{\mu}_{\boldsymbol{\theta}}^2$
$\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}^2, \sigma^2\mathbf{I})$
$\mathbf{z}_{\boldsymbol{\theta}}^2$
$\boldsymbol{\mu}_{\boldsymbol{\theta}}$
$\boldsymbol{\mu}_{\boldsymbol{\theta}}^3$
$\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}^3, \sigma^2\mathbf{I})$
$\mathbf{z}_{\boldsymbol{\theta}}^3$
$\boldsymbol{x}$
$\boldsymbol{z}$ y_1
$\boldsymbol{z}$ y_2
$\boldsymbol{z}$ y_3

Fig. 2: Illustration of our sequential generative process using decoder $\boldsymbol{\theta}$. This process sequentially takes the latent variable $\mathbf{z}$ (from encoder $\boldsymbol{\phi}$), stochastic variables $\mathbf{z}_{\boldsymbol{\theta}}^k$, and labels y_k for recovering input $\mathbf{x}$. Note that $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\cdot|\cdot)$ determines the mean of Gaussian distribution that generates each $\mathbf{z}_{\boldsymbol{\theta}}^k$.

an extension of conditional variational autoencoder (CVAE) [25] with sequential layers of stochastic variables $\{\mathbf{z}_{\boldsymbol{\phi}}^k\}_{k=1}^K$ and $\{\mathbf{z}_{\boldsymbol{\theta}}^k\}_{k=1}^K$ decided by each label y_k. In particular, our DGSM consists of sequential generative models which aims at describing the generation of multi-label data, followed by a deep classification model for MLC. This classification stage would jointly perform classifcation and approximated posterior inference, and the derivation of the learning objective based on variational inference (VI), so that multi-label prediction in such a weakly-supervised learning setting can be achieved. It is worth pointing out that, from the encoder-decoder perspective, Fig. 1a and Fig. 1b illustrates the framework of our classification and generative models, respectively. In the following subsections, we will detail the functionality and design for the above models.

Sequential Generative Models for Multi-Label Classification To address WS-MLC using sequential generative models, we assume that each instance $\mathbf{x}$ is generated from $\mathbf{y}$ with an additional latent variable $\mathbf{z}$. Without the loss of generality and following most exisint generative models [16, 17], we further assume that $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$, and have factorization of $p(\mathbf{y})$ as $p(\mathbf{y}) = \prod_{k=1}^K \mathtt{Bern}(y_k|\gamma_k)$, where y_k is the k-th label of $\mathbf{y}$ and $\boldsymbol{\gamma}_k$ is the parameter of Bernoulli distribution for y_k. We note that, one might consider a more representative prior based on the factorization of $p(\mathbf{y}) = p(y_1) \cdot \prod_{k=2}^K p(y_k|y_1, \ldots, y_{k-1})$. For simplicity, we consider the generation of different labels to be independent, and such an alternative prior is sufficiently satisfactory as confirmed later by our experiments. And, following the setting of [16], the priors $p(\mathbf{y})$ and $p(\mathbf{z})$ are set to be marginally independent.

Inspired by recent sequential methods for MLC [22, 8, 30, 21], our propose model also aims at leveraging information from multiple observed labels in a sequential manner during the learning process. More specifically, we choose to describe $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z})$, i.e., generation of multi-label data $\mathbf{x}$, as a *sequential generative process* with an additional set of intermediate stochastic variables $\{\mathbf{z}_{\boldsymbol{\theta}}^k\}_{k=1}^K$ as shown in Fig. 2. To be more precise, this generative process is formulated as follows:

$$\begin{aligned} &p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}) = g(\mathbf{x}|\mathbf{z}_{\boldsymbol{\theta}}^K; \boldsymbol{\theta}); \\ &\mathbf{z}_{\boldsymbol{\theta}}^k \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_{\boldsymbol{\theta}}^{k-1}, y_k, \mathbf{z}), \sigma^2\mathbf{I});\ 1 \le k \le K \\ &\mathbf{z}_{\boldsymbol{\theta}}^0 = \mathbf{0}, \end{aligned} \tag{1}$$

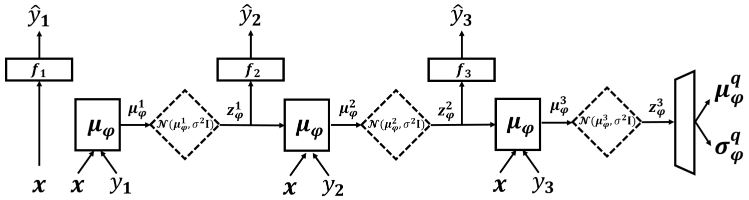


Fig. 3: Illustration of our sequential classification architecture ϕ in DSGM, which sequentially encodes input $\mathbf{x}$ and labels y_k into stochastic variables $\mathbf{z}_\phi^k$, with prediction layers f_k for determining label outputs $\hat{y}_k$.

where $g(\cdot|\cdot;\boldsymbol{\theta})$ is a likelihood function with parameters determined by non-linear transformation of $\boldsymbol{\theta}^K$. For example, Gaussian distribution can be utilized for $g(\cdot|\cdot)$ to describe the features with continuous values. In our framework, such a sequential generation process is realized by recurrent neural networks (RNN). That is, $\boldsymbol{\mu_\theta}(\cdot,\cdot,\cdot)$ outputs the mean vector from the non-linear transformation of $\mathbf{z}$, $\mathbf{z}_{\boldsymbol{\theta}}^{k-1}$ and y_k, which is implemented as the RNN cell that shares the model parameters across all labels y_k.

With the above generative model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y},\mathbf{z})$, we are able to learn the model parameters $\boldsymbol{\theta}$ by maximizing the marginal likelihood of $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})$ (from fully-labeled data), $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y^o})$ (from partially-labeled data), and/or $p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})$ (from unlabeled data). In other words, we are able to obtain $\boldsymbol{\theta}$ by solving

$$\arg\max_{\boldsymbol{\theta}} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_\ell} \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) + \sum_{(\mathbf{x},\mathbf{y^o})\in\mathcal{D}_o} \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y^o}) + \sum_{\tilde{\mathbf{x}}\in\mathcal{D}_u} \log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}). \quad (2)$$

To perform MLC with a given $\mathbf{x}$, classification can be achieved by $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ with model parameter $\boldsymbol{\theta}$.

Sequential Classification Model for Variational Inference of DSGM Unfortunately, learning (i.e., exact inference) of $\boldsymbol{\theta}$ by solving (2) is computationally prohibitive due to the need to compute intractable integral when applying Bayes rules. To enable an efficient approximated inference of $\boldsymbol{\theta}$, we design a novel learning algorithm based on the principle of variational inference [17, 4]. In particular, we propose a *deep sequential classification model* for posterior inference approximation. We then derive the variational lower bound and the corresponding optimization procedure accordingly.

We now discuss the design of our sequential classification model for the variational inference of $\boldsymbol{\theta}$. The key ingredient of variational inference is to introduce a fixed form distribution $q_{\boldsymbol{\phi}}(\cdot|\cdot)$, so that the posterior inference from observed variables to the latent ones can be achieved via $q_{\boldsymbol{\phi}}(\cdot|\cdot)$ instead of using $p_{\boldsymbol{\theta}}(\cdot|\cdot)$ which is in practice intractable. In the case of learning with fully-labeled training data, we seek to infer $\mathbf{z}$ from $(\mathbf{x},\mathbf{y})$ directly. For dealing with weakly-labeled data, unobserved labels are viewed as latent variables, which need to be inferred from $\mathbf{x}$ and the observed labels (if available).

With the above observation and motivation, the goal of $q_{\boldsymbol{\phi}}(\cdot|\cdot)$ is to achieve the following approximation of posterior inference:

$$\begin{aligned}
&q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{y}) \approx p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x},\mathbf{y});\ \forall(\mathbf{x},\mathbf{y}) \in \mathcal{D}_\ell \\
&q_{\boldsymbol{\phi}}(\mathbf{y^m},\mathbf{z}|\mathbf{x},\mathbf{y^o}) \approx p_{\boldsymbol{\theta}}(\mathbf{y^m},\mathbf{z}|\mathbf{x},\mathbf{y^o});\ \forall(\mathbf{x},\mathbf{y^o}) \in \mathcal{D}_o \\
&q_{\boldsymbol{\phi}}(\mathbf{y},\mathbf{z}|\tilde{\mathbf{x}}) \approx p_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{z}|\tilde{\mathbf{x}});\ \forall(\tilde{\mathbf{x}}) \in \mathcal{D}_u,
\end{aligned}$$

where we have partially-labeled data $(\mathbf{x},\mathbf{y^o}) \in \mathcal{D}_o$ in which $\mathbf{y^m}$ indicates the label vectors with missing ground truth. It is worth noting that, $q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{x}})$ essentially performs *classification as inference* for unlabeled data, and will be applied as the classification model for the testing stage. Inspired by [22, 8, 30, 21] which approach MLC by solving the task of label sequence prediction, and to meet the sequential nature of our proposed generative process, our *deep sequential classification model* would serve as $q_{\boldsymbol{\phi}}(\cdot|\cdot)$ for addressing WS-MLC (see Fig. 3 for illustration).

We now elaborate the architecture of our sequential classification model $q_{\boldsymbol{\phi}}(\cdot|\cdot)$, and explain in details on how to perform posterior inference given either fully-labeled, partially-labeled or unlabeled data via a set of intermediate latent variables $\{\mathbf{z}^k_{\boldsymbol{\phi}}\}_{k=1}^K$.

For labeled data, the sequential posterior inference $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{y})$ is performed as follows:

$$\begin{aligned}
&q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{y},\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}^q_{\boldsymbol{\phi}}(\mathbf{z}^K_{\boldsymbol{\phi}}),\boldsymbol{\sigma}^q_{\boldsymbol{\phi}}(\mathbf{z}^K_{\boldsymbol{\phi}})); && (3)\\
&\mathbf{z}^k_{\boldsymbol{\phi}} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}^{k-1}_{\boldsymbol{\phi}},\mathbf{x},y_k),\sigma^2\mathbf{I}),\ 1 \le k \le K; && (4)\\
&\hat{y}_k \sim \mathtt{Bern}(f^k_{\boldsymbol{\phi}}(\mathbf{z}^{k-1}_{\boldsymbol{\phi}})),\ 2 \le k \le K && (5)\\
&\hat{y}_1 \sim \mathtt{Bern}(f^1_{\boldsymbol{\phi}}(\mathbf{x})); && (6)\\
&\mathbf{z}^0_{\boldsymbol{\phi}} = \mathbf{0},
\end{aligned}$$

where $\hat{y}_k$ denotes the prediction of k-th label. Here $\boldsymbol{\mu}^q_{\boldsymbol{\phi}}(\cdot)$ and $\boldsymbol{\sigma}^q_{\boldsymbol{\phi}}(\cdot)$ are the deterministic functions that calculate the mean vector and diagonal covariance matrix for $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{y})$, respectively. On the other hand, $f^k_{\boldsymbol{\phi}}(\cdot,\cdot)$ determines the parameter of Bernouli distribution for prediction of $\hat{y}_k$. The main intuition behind such a design of $q_{\boldsymbol{\phi}}(\cdot|\cdot)$ is to encode $\mathbf{z}^k_{\boldsymbol{\phi}}$ with the information from $(\mathbf{x}, y_1, \ldots, y_k)$. Such encoding allows us to resemble the following factorization by predicting each $\hat{y}_{k+1}$ with $\mathbf{z}^k_{\boldsymbol{\phi}}$:

$$q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}) = q_{\boldsymbol{\phi}}(y_1|\mathbf{x})\prod_{k=2}^{K} q_{\boldsymbol{\phi}}(y_k|\mathbf{x}, y_1, \ldots, y_{k-1}).$$

We see that, $\mathbf{z}^K_{\boldsymbol{\phi}}$ with such relation would encode information from all observed variables, and thus can be directly used to determine $\mathbf{z}$ from $\mathbf{x}$ and $\mathbf{y}$.

For partially-labeled data in WS-MLC, we adopt the same posterior inference procedure as (3)-(6) except that we now consider the meanfield variational family $q_{\boldsymbol{\phi}}(\mathbf{y^m},\mathbf{z}|\mathbf{x},\mathbf{y^o}) = q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{y^o})q_{\boldsymbol{\phi}}(\mathbf{y^m}|\mathbf{x},\mathbf{y^o})$. Nevertheless, despite the factorized probability representation, information from *all labels* should still be exploited to infer $\mathbf{z}$, which is achieved by utilizing the predicted label $\hat{y}_k$ instead (in the case where y_k is missing). To be more precise, we modify (4) to calculate $\mathbf{z}_{\boldsymbol{\phi}}^k$ by.

$$\mathbf{z}_{\boldsymbol{\phi}}^k \sim \begin{cases} \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}_{\boldsymbol{\phi}}^{k-1}, \mathbf{x}, y_k), \sigma^2 \mathbf{I}), \text{ if } y_k \in \mathbf{y^o} \\ \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}_{\boldsymbol{\phi}}^{k-1}, \mathbf{x}, \hat{y}_k), \sigma^2 \mathbf{I}), \text{ if } y_k \notin \mathbf{y^o}, \end{cases} \tag{7}$$

where $\hat{y}_k \in \{0, 1\}$ is a binary sample based on the predicted probability that $y_k = 1$ via (5). With such modification, our sequential classification model is able to infer $\mathbf{z}$ with information from all labels even if some are unobserved.

As for unlabeled data in WS-MLC, we also utilize the meanfiled variational family $q_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{z}|\tilde{\mathbf{x}}) = q_{\boldsymbol{\phi}}(\mathbf{z}|\tilde{\mathbf{x}}) q_{\boldsymbol{\phi}}(\mathbf{y}|\tilde{\mathbf{x}})$. By realizing that unlabeled data is the data with all label missing, we perform posterior sequential posterior inference in exactly the same way as that for partially-labeled data. In this case, (7) would degenerate to the case with each $y_k \notin \mathbf{y^o}$.

Finally, we implement the above sequntial posterior inference with $q_{\boldsymbol{\phi}}(\cdot|\cdot)$ via RNN, as depicted in Fig. 3. That is, $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\cdot, \cdot, \cdot)$ used in both (4) and (7) is realized as an RNN cell, which is the same as those in our sequential generative model.

Objective for Variational Lower Bound In this subsection, we discuss how we derive the objective of the lower bound for variational inference. Note that the calculation of $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ requires the integral over the samples of $\{\mathbf{z}_{\boldsymbol{\phi}}^k\}_{k=1}^K$ even if $\mathbf{x}$ and $\mathbf{y}$ are known. This would be undesirable for the derivation of the variational lower bound, as such integrals cannot be analytically calculated.

By applying location-scale transform of Gaussian distributions, one would be able to determine the exact parameters of distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ (i.e., mean and covariance) by introducing a set of random variables $\{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k\}_{k=1}^K$. Thus, (4) can be rewritten as

$$\mathbf{z}_{\boldsymbol{\phi}}^k = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}_{\boldsymbol{\phi}}^{k-1}, \mathbf{x}, y_k) + \sigma^2 \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k;\ 1 \leq k \leq K, \tag{8}$$

where each $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k \sim \mathcal{N}(0, \mathbf{I})$ is an independent sample of standard Gaussian distribution. Consequently, one can derive the variational lower bound by taking

$$\mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^1, \ldots, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^K} [\mathcal{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^1, \ldots, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^K) \parallel p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}, \mathbf{y}))]$$

as a starting point, where $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^1, \ldots, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^K)$ denotes the fixed distribution given the sampled value of $\{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k\}_{k=1}^K$.

Based on location-transform techniques of $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}) = \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}}[p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_{\boldsymbol{\theta}})]$, where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}} = \{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^k\}_{k=1}^K$ is another set of independent samples from standard Gaussian distribution in addition to $\{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k\}_{k=1}^K$ for $q_{\boldsymbol{\phi}}(\cdot|\cdot)$, the lower bound for the labeled data $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_\ell$ can now be expressed as $\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}}[-\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})]$, where $\boldsymbol{\epsilon}_{\boldsymbol{\phi}} = \{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k\}_{k=1}^K$. Finally, by Jensen's inequality (for concave functions), we have

$$\begin{aligned} -\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) = &\log p(\mathbf{y}) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_{\boldsymbol{\theta}})]] \\ &-\mathcal{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) \| p(\mathbf{z})). \end{aligned} \tag{9}$$

In order to deal with partially-labeled data $(\mathbf{x}, \mathbf{y^o}) \in \mathcal{D}_o$, we need both $q_{\boldsymbol{\phi}}(\mathbf{y^m}|\mathbf{x}, \mathbf{y^o})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y^o})$ for deriving the associated variational lower bound objective. However, as noted above, the sequential posterior inference with $q_{\boldsymbol{\phi}}(\cdot, \cdot)$ using (7) involves sampling for unobserved labels. Even with the technique of location-scale transform on $\{\mathbf{z}_{\boldsymbol{\phi}}^k\}_{k=1}^K$, oue still needs to marginalize out the sampling regarding $\mathbf{y}^{\mathbf{m}}$ to obtain $q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x}, \mathbf{y}^{\mathbf{o}})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}^{\mathbf{o}})$. To alleviate this problem, we choose to rewrite (7) as

$$\mathbf{z}_{\boldsymbol{\phi}}^k = \begin{cases} \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}_{\boldsymbol{\phi}}^{k-1}, \mathbf{x}, y_k) + \sigma^2 \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k, & \text{if } y_k \in \mathbf{y}^{\mathbf{o}} \\ [\![\alpha_k \geq p_k]\!] \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}_{\boldsymbol{\phi}}^{k-1}, \mathbf{x}, 0) + [\![\alpha_k < p_k]\!] \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{z}_{\boldsymbol{\phi}}^{k-1}, \mathbf{x}, 1) + \sigma^2 \boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k, & \text{if } y_k \notin \mathbf{y}^{\mathbf{o}} \end{cases} \tag{10}$$

where $\alpha_k \sim U(0,1)$, $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \mathbf{I})$, $[\![\cdot]\!]$ is the indicator function and p_k is the probability that $y_k = 1$ from (5) and (6). We see that (10) is effectively a reparameterization of sampling of $\mathbf{z}_{\boldsymbol{\phi}}^k$ with $\boldsymbol{\alpha} = \{\alpha_k\}_{k=1}^K$ and $\boldsymbol{\epsilon}_{\boldsymbol{\phi}} = \{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}^k\}_{k=1}^K$, which allows exact determinination of $q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x}, \mathbf{y}^{\mathbf{o}})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y}^{\mathbf{o}})$ with a set of $(\boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})$. With this observation, we have the lower bound objective for partially-labeled data as $\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \geq \mathbb{E}_{\boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}}[-\mathcal{M}(\mathbf{x}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})]$, where

$$\begin{aligned} &- \mathcal{M}(\mathbf{x}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\log p(\mathbf{y})] \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \boldsymbol{\alpha}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x}, \boldsymbol{\alpha}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}_{\boldsymbol{\theta}})]]] \\ &- \mathcal{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \boldsymbol{\alpha}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) \| p(\mathbf{z})) + \mathcal{H}(q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x}, \boldsymbol{\alpha}, \mathbf{y}^{\mathbf{o}}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})) \end{aligned} \tag{11}$$

where $\mathcal{H}(\cdot)$ is the entropy function by again realizing that the meanfield assumption still holds even with $\boldsymbol{\alpha}, \boldsymbol{\epsilon}$ included due to the design of $q_{\boldsymbol{\phi}}(\cdot, \cdot)$.

Finally, as for observation of unlabeled data $\tilde{\mathbf{x}} \in \mathcal{D}_u$, the variational lower bound can be derived similarly to that of partially-labeled data by realizing that the reparameterization of $\{\mathbf{z}_{\boldsymbol{\phi}}^k\}_{k=1}^K$ using (10) degenerates to the case with each $y_k \notin \mathbf{y}^{\mathbf{o}}$. Consequently, the lower bound objective for unlabeled data can be expressed as $\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \geq \mathbb{E}_{\boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}}[-\mathcal{U}(\tilde{\mathbf{x}}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})]$, where

$$\begin{aligned} &- \mathcal{U}(\tilde{\mathbf{x}}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\log p(y)] \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})}[\mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\theta}})]]] \\ &- \mathcal{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) \| p(\mathbf{z})) + \mathcal{H}(q_{\boldsymbol{\phi}}(\mathbf{y}|\tilde{\mathbf{x}}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})). \end{aligned} \tag{12}$$

With (9), (11) and (12), we now obtain the lower bound objective of marginal likelihood regarding the data with weakly labels. As suggested in [16], it is preferable to have the classifier directly perform label prediction. Thus, we further augment the derived lower bound objective with a discriminative loss on the observed labels for fully-labeled and partial labeled data, resulting in the following final minimization objective:

$$\begin{aligned} &\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_\ell} \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\phi}}}[\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) - \log q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})] \\ &+ \sum_{(\mathbf{x}, \mathbf{y}^{\mathbf{o}}) \in \mathcal{D}_o} \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\alpha}, \boldsymbol{\phi}}}[\mathcal{M}(\mathbf{x}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}) - \log q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{o}}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})] \\ &+ \sum_{\tilde{\mathbf{x}} \in \mathcal{D}_u} \mathbb{E}_{\boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}}}[\mathcal{U}(\tilde{\mathbf{x}}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})]. \end{aligned} \tag{13}$$

With the set-up of the above objectives, the resulting $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\epsilon}_{\boldsymbol{\phi}})$ will be used to recognize future unseen test data. For testing, we determine the binary prediction of each label $\hat{y}_k$ by directly thresholding the predicted probability with threshold of 0.5 (which is implemented by setting all $\alpha_k = 0.5$).

Learning of DSGM We now detail the learning and optimization of our DGSM with the objective functions introduced above. For the loss of labeled data in (9), the KL-divergence term can be analytically computed for any $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}$ as both $q_{\boldsymbol{\phi}}(\mathbf{z}|\cdot)$ and $p(\mathbf{z})$ are Gaussian distributions. For the part $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\cdot)}[\mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y},\mathbf{z},\boldsymbol{\epsilon}_{\boldsymbol{\theta}})]]$ of the loss function, the gradient can be efficiently estimated using the reparameterization trick on $q_{\boldsymbol{\phi}}(\mathbf{z}|\cdot)$ [17] and a single sample of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$. Since the gradient of the outer expectation $\mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}}[\cdot]$ in (13) for the loss of fully-labeled data can be efficiently estimated using a single sample of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, we advance techniques of stochastic gradient descent (SGD) for optimizing our network parameters $(\boldsymbol{\theta},\boldsymbol{\phi})$ using fully-labeled data.

For the loss of partially-labeled data in (11), we apply aforementioned techniques with a sample of $\boldsymbol{\alpha}$ for gradient estimation. However, other optimization issues need to be addressed. First, we need to calculate the expectation $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x},\mathbf{y}^{\mathbf{o}},\cdot)}\log p_{\boldsymbol{\theta}}(\mathbf{x}|\cdot)$ with respect to the predicted probability of missing labels $\mathbf{y}^{\mathbf{o}}$ given $(\mathbf{x},\mathbf{y}^{\mathbf{o}})$ (note that we omit $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ and $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}$ for presentation simplicity). The other issue is that, we need to handle the discontinuous indicator function in (7).

Regarding the calculation of $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x},\mathbf{y}^{\mathbf{o}},\cdot)}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\cdot)]$, explicitly marginalizing out $q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x},\mathbf{y}^{\mathbf{o}},\cdot)$ is a possible solution. However, it would take $\mathcal{O}(2^{|\mathbf{y}^{\mathbf{m}}|})$ time due to the need to examine each combination of missing labels, making it computationally prohibitive when the number of missing labels is large. To resolve the issue, we reparameterize the expectation to have the form

$$\mathbb{E}_{\boldsymbol{\beta}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\cdot,\boldsymbol{\beta},q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x},\mathbf{y}^{\mathbf{o}},\cdot))],$$

where $\boldsymbol{\beta}=\{\beta_k\}_{k=1}^{K}$, $\beta_k\sim U(0,1)$ by rewriting the sampling of $\mathbf{z}_{\boldsymbol{\theta}}^{k}$ in (1) as

$$\mathbf{z}_{\boldsymbol{\theta}}^{k}=\begin{cases}\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_{\boldsymbol{\theta}}^{k-1},y_k,\mathbf{z})+\sigma^2\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^{k}, & \text{if } y_k\notin\mathbf{y}^{\mathbf{m}}\\ [\![\beta_k\geq p_k]\!]\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_{\boldsymbol{\theta}}^{k-1},0,\mathbf{z})+[\![\beta_k<p_k]\!]\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_{\boldsymbol{\theta}}^{k-1},1,\mathbf{z})+\sigma^2\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^{k}, & \text{if } y_k\in\mathbf{y}^{\mathbf{m}}.\end{cases} \tag{14}$$

where p_k is the predicted probability that $y_k=1$. It can be seen that, the above reparameterization is analogous to that of sampling $\mathbf{z}_{\boldsymbol{\phi}}^{k}$ with (7) for the sequential posterior inference with $q_{\boldsymbol{\phi}}\cdot|\cdot$. This allows us to efficiently estimate the gradient of the expectation $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y}^{\mathbf{m}}|\mathbf{x},\mathbf{y}^{\mathbf{o}},\cdot)}\log p_{\boldsymbol{\theta}}(\mathbf{x}|\cdot)$ with an extra single sample of $\boldsymbol{\beta}$.

As for dealing with the discontinuity of the indicator function, we adopt the straight-through estimator (STE) in [3] for addressing this problem. More precisely, we calculate the loss in (11) in the forward pass using the normal indicator function, and replace the indicator function with identity function during the backward pass to calculate the gradient. The use of STE leads to promising performance as noted in [3].

For calculating the loss term of unlabeled data, we apply the same techniques and reparameterizing $\mathbf{z}_{\boldsymbol{\theta}}^{k}$ with (14) by observing that (14) degenerates to the case with each $y_k\in\mathbf{y}^{\mathbf{m}}$ for unlabeled data.

With the above explanation and derivations, we are able to efficiently obtain the estimation of gradient with respect to (13) for both fully-labeled and weakly-labeled data. As a result, the final discriminative classifier $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$ in our DSGM can be learned by updating $(\boldsymbol{\theta},\boldsymbol{\phi})$ with SGD techniques.

3.3 Discussions

Finally, we discuss the connection and difference between our proposed DSGM and recent models for related learning tasks.

The use of deep generative models for semi-supervised mutli-class classification (not MLC) has been recently studied in [16]. In particular, [16] jointly trains their classification network $q_{\boldsymbol{\phi}}(y|\mathbf{x})$, inference network $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, y)$, and generative network $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, y)$ by optimizing the variational lower bound for likelihood of observed labeled and unlabeled data. However, applying the models of [16] for (semi-supervised) MLC requires explicit examination of all 2^K possible label combinations when calculating the lower bound objective for unlabeled data. This is quite computationally infeasible especially when K is large. In contrast, the sequential architectures and the corresponding optimization procedure in our proposed DSGM provides linear dependency of K, which is in practice more applicable for semi-supervised or weakly-supervsied MLC tasks.

On the other hand, formulating MLC as a sequential label prediction has been studied in recent literature [22, 8, 30, 21]. For example, [30, 21] advances RNNs to sequentially predict the labels while implicitly observing their dependency. While the use of sequential label prediction has been widely investigated with promising performances, existing models cannot be easily extended to handle partially-labeled and unlabeled data (i.e., WS-MLC tasks). Such robustness is particularly introduced into our DSGM. As confirmed later by the experiments, our DSGM performs favorably against state-of-the-art deep MLC models in such challenging settings.

4 Experiments

4.1 Experiment Settings

To evaluate the performance of our proposed DSGM for WS-MLC, we consider the following datasets: *iaprtc12*, *espgame*, *mirflickr*, *NUS-WIDE*, and *MSCOCO*. The first three datasets are image recognition datasets used in [11], where 1000-dimensional of bag-of-words features based on SIFT. *NUS-WIDE* [7] and *MSCOCO* [19] are two other large scale datasets typically used for evaluation of image annotation. We summarize the key statistics of the above datasets in the appendix. For all datasets, we discard the instances with no positive labels as done in [10]. For *NUS-WIDE* and *MSCOCO*, we use the bottom four convolutional layers of a ResNet-152 [12] trained on Imagenet without fine-tuning to extract 2048-dimensional feature vectors in order to utilize both the high-level and low-level information of raw images.

In our DSGM architecture, we use gated-recurrent unit [6] as the recurrent cells to model $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$. The dimensions of latent variables $\{\mathbf{z}_{\boldsymbol{\theta}}^k\}_{k=1}^K$ and $\{\mathbf{z}_{\boldsymbol{\phi}}^k\}_{k=1}^K$ are set to 128, while the dimension of $\mathbf{z}$ is fixed as 64. The variance σ^2 of $\{\mathbf{z}_{\boldsymbol{\theta}}^k\}_{k=1}^K$ and $\{\mathbf{z}_{\boldsymbol{\phi}}^k\}_{k=1}^K$ is set to 0.005. When applying DSGM for WS-MLC, we reduce the dimension of feature vector $\mathbf{x}$ to 512 by a linear transformation, which is parameterized as a fully connected layer without activation. Following [30, 21], the label order for sequential learning/prediction is set from the most frequent one to the rarest one (see such suggestions in [21]). Nevertheless, in our experiment, we do not observe significant differences between the choices of different label orders. To perform stochastic gradient descent for

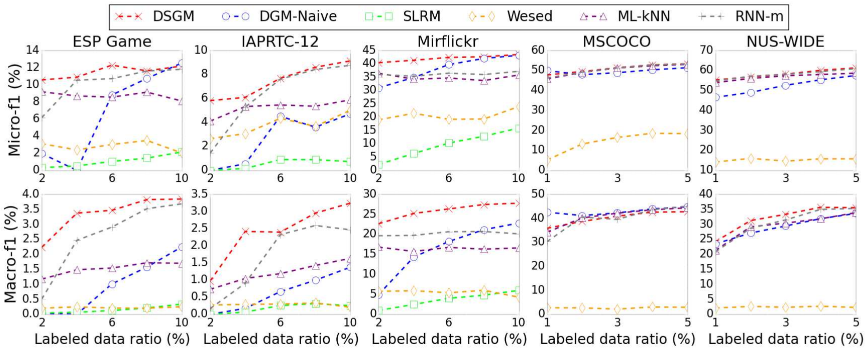


Fig. 4: Performance comparisons in terms of Micro-f1 and Macro-f1 for datasets with varying labeled data ratios.

optimization, we exploit Adam [15] with a fixed learning rate of 0.003, and the batch size is fixed to 100. Note that for the experiment of semi-supervised MLC, we pretrai our DSGM with only labeled data for 100 epochs for faster convergence. Finally, both Micro-f1 and Macro-f1 are considered as evaluation metrics [27].

4.2 Comparisons with Semi-Supervised MLC Algorithms

We first evaluate our DSGM on the task of semi-supervised MLC (SS-MLC), where the dataset contains both fully-labeled and unlabeled data. Two state-of-the-arts SS-MLC algorithms are considered: Semi-supervised Low-rank Mapping (SLRM) [14] and Weakly Semi-supervised Deep Learning (Wesed) [33] (with its inductive setting viewed as a semi-supervised setting). In addition, we consider a naïve extension of state-of-the-art deep generative approach for semi-supervised multi-class classification (SS-MCC) [16], DGM-Naïve (detailed in supplementary). For completeness, we also include two well-known or state-of-the-art supervised MLC algorithms, ML-kNN [36] and RNN-m [21]. We follow [14, 33] to set the hyper parameters for SLRM and Wesed, and fix $k = 5$ for ML-kNN. For RNN-m, we use the same RNN cell and the feature transformation as those in our DSGM, and set the learning rate as 0.001. The training details of DGM-Naïve are discussed in the supplementary material. For each dataset, we randomly split into two subsets with equal sizes for training and testing. The average results of five random splits are presented.

The comparison results are shown in Figure 4, where the horizontal axis represents the ratio of labeled data with respect to the entire training set. Note that experiments of SLRM are not conducted on two large-scale datasets *NUS-WIDE* and *MSCOCO*, as SLRM requires pairwise information between training instances. From Figure 4, we see that our DSGM achieved improved results when comparing to the state-of-the-art SS-MLC methods of SLRM and Wesed, as well as the extension of the recent generative SS-MCC approach, DGM-Naïve. We note that, for large-scale datasets of *NUS-WIDE* and *MSCOCO*, supervised MLC method of ML-kNN and RNN-m achieved comparable

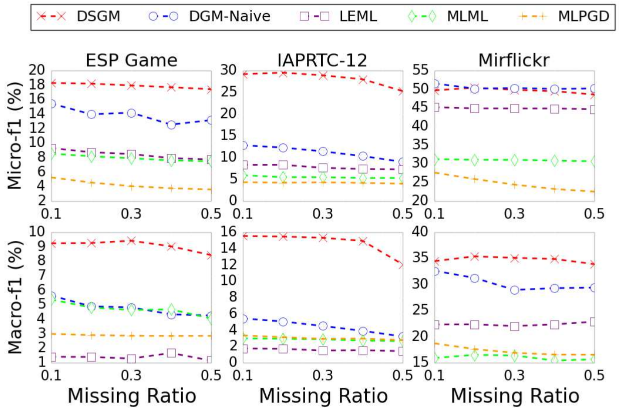


Fig. 5: Micro-f1 and Macro-f1 with different missing label ratios.

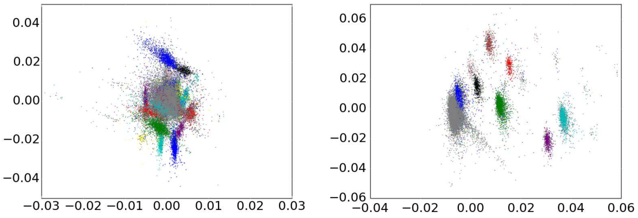
results as ours did. The above empirical results support our exploitation of unlabeled data for MLC tasks.

Further inspection on Figure 4 reveals that the use of more powerful features (i.e., those calculated by Resnet 152) generally resulted in favorable performances (e.g., DGM-Naïve, ML-kNN, RNN-m, and ours). On the other hand, when it comes to the datasets *iaprtc12*, *espgame* and *mirflickr* using low-level features, our DSGM clearly outperformed DGM-Naïve, ML-kNN and RNN-m in most of the cases. Moreover, we observe that DSGM remarkably performed against DGM-Naïve on the above three datasets especially when the labeled data ratio becomes smaller. This demonstrates the effectiveness of our sequential architecture in exploiting unlabeled data for MLC.

4.3 Comparisons with Algorithms for MLC with Missing Labels

Next, we compare our proposed model with state-of-the-art MLC algorithms for WS-MLC, particularly the observation of partially-labeled data, or data with missing labels. The methods to be compared to include LEML [35], Multi-label Learning with Missing Labels (MLML) [31], and ML-PGD (Multi-label Learning with Missing Labels Using Mixed Graph) [32]. We also modify DSGM-Naïve to handle data missing label data for comparison of state-of-the-art deep generative approach in such settings. The scenario of missing labels is simulated by randomly dropping the ground truth labels, with ratio varying from 10% to 50%.

Figure 5 illustrates and compares the performances, where the horizontal axis indicates the missing ratio. The results from Figure 5 demonstrate that our algorithm performed against state-of-the-art algorithms, and confirmed the robustness of our methods for different MLC tasks (i.e., SS-MLC and MLC with missing labels). With a closer inspection between the results of DSGM and DGM-Naïve, we see that while DGM-Naïve reported promising and satisfactory results, DSGM in general still remarkably outperformed DGM-Naïve. This reflects the importance and the advantage of our se-



(a) Standard CVAE

(b) DSGM

Fig. 6: Visualization of the derived latent vectors on *mirflickr* for standard CVAE and DSGM.

quential architecture which integrate generative models and discriminative classifiers for WS-MLC.

4.4 Qualitative Studies

Finally, we provide qualitative studies regarding our sequential architecture of generative models. Specifically, we train a standard Conditional Variational Auto Encoder (CVAE) [25] and the sequential generative models of DSGM on *mirflickr* using the first 10 labels with the dimension of $\mathbf{z}$ set to 2. This allows us to visualize the inferred latent vectors for each instance $\mathbf{x}$. We plot the derived latent vectors of each $\mathbf{x}$ corresponding to the 10 most common label combinations in Figure 6, which reflects label correlation information. The 10 most common combinations cover over 95% of instances, and each latent vector is colored based on its associated label combination in Figure 6.

From the visualization results shown in Figure 6, it is clear that our proposed sequential generative model resulted in more representative latent vectors when comparing to those of CVAE. From this figure, we see that our model better represents and describes the relationship between data with different labels. This also supports the use of our proposed deep generative model of multi-labeled data in the weakly-supervised setting.

5 Conclusion

We proposed a deep generative model, DSGM for solving WS-MLC problems. DSGM integrates a unique deep sequential generative model to descrbe multi-label data as well as a novel deep sequential classification model for both posterior inference and classification. The variational lower bound is derived for the learning of our sequential generative model together with an efficient optimization procedure. Experiment results confirmed the superiority of our proposed DSGM over state-of-the-art semi-supervised MLC approaches, and those designed to handle MLC with missing labels. We further demonstrated that DSGM would be more effective than naïve utilization of deep generative models regarding WS-MLC, i.e., SS-MLC and MLC with missing labels.

References

1. Adams, R.P., Ghahramani, Z.: Archipelago: nonparametric bayesian semi-supervised learning. In: ICML 2019. pp. 1–8 (2009)
2. Bello, J.P., Chew, E., Turnbull, D.: Multilabel classification of music into emotions. In: ICMIR 2008. pp. 325–330 (2008)
3. Bengio, Y., Léonard, N., Courville, A.C.: Estimating or propagating gradients through stochastic neurons for conditional computation. CoRR **abs/1308.3432** (2013)
4. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. CoRR **abs/1601.00670** (2016)
5. Chen, G., Song, Y., Wang, F., Zhang, C.: Semi-supervised multi-label learning by solving a sylvester equation. In: SDM 2008. pp. 410–419 (2008)
6. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: SSST@EMNLP 2014. pp. 103–111 (2014)
7. Chua, T., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from national university of singapore. In: CIVR 2009 (2009)
8. Dembczynski, K., Cheng, W., Hüllermeier, E.: Bayes optimal multilabel classification via probabilistic classifier chains. In: ICML. pp. 279–286 (2010)
9. Elisseeff, A., Weston, J.: A kernel method for multilabelled classification. In: NIPS 2001 (2001)
10. Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation. CoRR **abs/1312.4894** (2013)
11. Guillaumin, M., Mensink, T., Verbeek, J.J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: ICCV. pp. 309–316 (2009)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
13. Jain, V., Modhe, N., Rai, P.: Scalable generative models for multi-label learning with missing labels. In: ICML 2017. pp. 1636–1644 (2017)
14. Jing, L., Yang, L., Yu, J., Ng, M.K.: Semi-supervised low-rank mapping learning for multi-label classification. In: CVPR 2015. pp. 1483–1491 (2015)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
16. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: NIPS 2014. pp. 3581–3589 (2014)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. CoRR **abs/1312.6114** (2013)
18. Lin, G., Liao, K., Sun, B., Chen, Y., Zhao, F.: Dynamic graph fusion label propagation for semi-supervised multi-modality classification. Pattern Recognition **68**, 14–23 (2017)
19. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV. pp. 740–755 (2014)
20. de Lucena, D.C.G., Prudêncio, R.B.C.: Semi-supervised multi-label k-nearest neighbors classification algorithms. In: BRCIS 2015. pp. 49–54 (2015)
21. Nam, J., Loza Mencía, E., Kim, H.J., Fürnkranz, J.: Maximizing subset accuracy with recurrent neural networks in multi-label classification. In: NIPS 2017. pp. 5419–5429 (2017)
22. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning **85**(3), 333–359 (2011)
23. Rubin, T.N., Chambers, A., Smyth, P., Steyvers, M.: Statistical topic models for multi-label document classification. Machine Learning **88**(1-2), 157–208 (2012)
24. Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. CoRR **abs/1606.03498** (2016)

25. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: NIPS 2015. pp. 3483–3491 (2015)
26. Tai, F., Lin, H.: Multilabel classification with principal label space transformation. Neural Computation **24**(9), 2508–2542 (2012)
27. Tang, L., Rajan, S., Narayanan, V.K.: Large scale multi-label classification via metalabeler. In: WWW 2009. pp. 211–220 (2009)
28. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, 2nd ed., pp. 667–685 (2010)
29. Wang, H., Huang, M., Zhu, X.: A generative probabilistic model for multi-label classification. In: ICDM 2008. pp. 628–637 (2008)
30. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: A unified framework for multi-label image classification. In: CVPR 2016. pp. 2285–2294 (2016)
31. Wu, B., Liu, Z., Wang, S., Hu, B., Ji, Q.: Multi-label learning with missing labels. In: ICPR 2014. pp. 1964–1968 (2014)
32. Wu, B., Lyu, S., Ghanem, B.: ML-MG: multi-label learning with missing labels using a mixed graph. In: ICCV 2015. pp. 4157–4165 (2015)
33. Wu, F., Wang, Z., Zhang, Z., Yang, Y., Luo, J., Zhu, W., Zhuang, Y.: Weakly semi-supervised deep learning for multi-label image annotation. IEEE Trans. Big Data **1**(3), 109–122 (2015)
34. Yeh, C., Wu, W., Ko, W., Wang, Y.F.: Learning deep latent space for multi-label classification. In: AAAI 2017. pp. 2838–2844 (2017)
35. Yu, H., Jain, P., Kar, P., Dhillon, I.S.: Large-scale multi-label learning with missing labels. In: ICML 2014. pp. 593–601 (2014)
36. Zhang, M., Zhou, Z.: ML-KNN: a lazy learning approach to multi-label learning. Pattern Recognition **40**(7), 2038–2048 (2007)
37. Zhu, X., Goldberg, A.B.: Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2009)