Information Sciences 626 (2023) 738–753

Contents lists available at ScienceDirect

Information Sciences

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

Learning multiple gaussian prototypes for open-set recognition

Jiaming Liu [a], Jun Tian [a], Wei Han [a], Zhili Qin [a], Yulu Fan [b], Junming Shao [a,*]

[a] Data Mining Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

Keyword:
Open-set recognition
Novelty detection
Gaussian prototype
Variational auto-encoder

ABSTRACT

Open-set recognition aims to deal with unknown classes that do not exist in the training phase. The key is to learn effective latent feature representations for classifying the already known classes as well as detecting new emerging ones. In this paper, we learn multiple Gaussian prototypes to better represent the complex classes distribution in both generative and discriminative ways. With the generative constraint, the latent variables of the same class clusters compactly around the corresponding Gaussian prototypes, preserving extra space for the samples of unknown classes. The discriminative constraint separates the Gaussian prototypes of different classes, which further improves the discrimination capability for the known classes. Importantly, the entire framework can be directly derived from the Bayesian inference, thus providing theoretical support for open-set recognition. Experimental results of different datasets verify the reliability and effectiveness of the proposed method. Our code is available at: https://github.com/LiuJMzzZ/MGPL.

1. Introduction

In recent years, the field of deep learning has progressed rapidly and received widespread attention. Classification tasks in different application scenarios such as medical images [1], remote sensing [2] and multivariate time series [3] have achieved amazing performances. Traditional classification tasks are assumed in a closed set, where all categories in the test set are from the training set. However, in practical application scenarios, some unknown categories that have not appeared during training may occur in the testing phase [4]. Under these circumstances, the closed-set classifiers can only make the prediction by choosing one of the known categories, which limits the applicability in dynamic and changing scenarios.

To overcome this limitation, the open-set recognition task is introduced by [5], in which the open-set model is able to classify the known classes and detect new emerging classes at the same time. In the open-set scenario, categories from the training set are defined as known classes, and the newly appeared categories in the testing set are called unknown classes. Therefore, the key to handle the open-set problem is to define a feasible confidence score selection strategy [6,7] or to learn an effective latent feature representation space [8,9].

For the novel unknown classes in testing, it is intuitive to set a threshold on the output confidence of a network to detect the unknowns. However, the Softmax-based deep network tends to be overfitting on known classes [10]. Using Softmax to obtain the probability over the known classes will also cause the model to generate a higher degree of confidence for the unknown classes, and in

* Corresponding author.
E-mail addresses: liujiaming@std.uestc.edu.cn (J. Liu), juntian@std.uestc.edu.cn (J. Tian), weihan@std.uestc.edu.cn (W. Han), qinzhili@outlook.com (Z. Qin), ylfan@std.uestc.edu.cn (Y. Fan), junmshao@uestc.edu.cn (J. Shao).

https://doi.org/10.1016/j.ins.2023.01.062
Received 22 February 2022; Received in revised form 6 January 2023; Accepted 8 January 2023
Available online 12 January 2023

turn make the confidence thresholds infeasible to select. To tackle this limit, OpenMax [11] used a calibrated score to replace the Softmax layer in the traditional networks. The core idea is to model the distribution of distances to the mean activation of each class and define the reject threshold based on extreme value theory (EVT). In this way, an unknown sample can be detected by measuring the distance to the mean activation of known classes. On this basis, some works have been further proposed [12,13]. These methods estimated the class distribution in latent space by computing the average class activation value after training the model which cannot model the real class distribution. In fact, for complex datasets in real scenarios, the distribution of each category may be more complicated, and it is difficult to describe the distribution of each class with only one center.

Another focus for open-set recognition is to learn a better latent feature space. Traditional closed-set classification methods are mostly of discriminative models. These closed-set classifiers only focus on learning the decision boundaries between different known classes rather than obtaining the intrinsic distribution of the known classes, and features of the known classes tend to occupy the entire latent space, reserving no space for the newly appeared classes. Therefore, it would be much better for open-set learning if the latent features of the same class are gathered together compactly and the features of different classes are far from each other. In this way, effective feature representations for known classes can be obtained, and meanwhile extra space can also be reserved for unknown classes in changing scenarios. Some works have made contributions to model the distribution in the latent space from the perspective of generation [14–16]. For example, CGDL [15] constrained the hidden space to a Gaussian distribution related to the category through the conditional VAE. However, these methods learned the feature space only in the generative way which will reduce the ability of classification. Some other hybrid models have been proposed recently [17,18], but these methods are all heuristic and lack the theoretical framework.

Aiming at the above problems, we propose a novel method called *Multiple Gaussian Prototypes Learning* (MGPL) to learn the effective latent space representation. As shown in Fig. 1, the distribution of known classes in the latent space can be represented by Gaussian prototypes, each of which is formed by a Gaussian distribution with specific parameters and a predefined class label. To handle the complex data distribution, multiple Gaussian prototypes are assigned to each class. These Gaussian prototypes are derivable and can be optimized by gradient descent methods with the entire framework from both the generation and discrimination perspectives. Under the generative constraint, the latent variables tend to gather around the corresponding Gaussian prototypes compactly and preserve some space for the unknown classes. To further improve the discrimination ability, Gaussian prototypes of different classes are separated from each other under the discriminative constraint. Importantly, the entire framework is derived from the Bayesian inference, providing theoretical support for open-set recognition tasks. Variable experiments with the existing methods on the benchmark datasets are conducted, and our proposed method exhibits very competitive performances with satisfactory reliability and effectiveness on the open-set recognition tasks.

Contributions of our work are summarized as follows:

- We have established a new open-set recognition framework, where the multiple Gaussian prototypes are learned to represent the classes distribution in the latent space. The classification capability for known classes is maintained while recognizing the unknowns;
- We derive the entire framework through Bayesian inference from both the generation and discrimination perspectives, providing theoretical support for open-set recognition;
- We conduct variable experiments on the benchmark datasets used for open-set recognition tasks, and our proposed method shows very competitive results and outperforms the existing methods by a large margin in several cases.

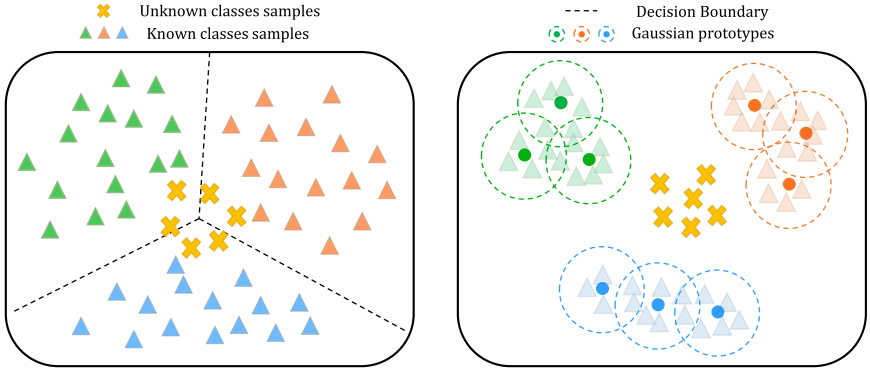


Fig. 1. The latent space of the traditional closed-set classifier (CNN) compare to our proposed MGPL open-set model. We can infer that the closed-set classifier only aims to learn the decision boundary that split the whole latent feature space for known classes. Therefore, the newly appeared classes samples are more likely to be recognized as a known class. In our proposed MGPL model, the latent features of known classes compact around the corresponding Gaussian prototypes, and the Gaussian prototypes of different classes are separated, reserving extra space for the detection of unknown classes. Moreover, multiple Gaussian prototypes are assigned to represent the same class to handle the complex data distribution.

2. Related Work

2.1. Open-set Recognition

While closed-set classification has been widely studied, more attention is paid to open-set recognition for its capability of detecting the unknown classes while classifying the known ones. The open-set recognition concept was first introduced by [5]. At present, open-set recognition methods can be broadly divided into three categories: discriminative methods, generative methods, and hybrid methods.

Discriminative Methods. In the early stages, some traditional machine learning methods were proposed to solve the open-set recognition problem. For example, based on the extreme value theory (EVT), [19] proposed a Weibull-calibrated SVM which considered a distribution of decision scores for unknown detection. A similar method was also presented in [20]. Extreme value machine (EVM) [21] modeled the probabilities of known classes inclusion through theoretical analysis. [22] proposed a sparse-representation-based method called SROSR, which also utilized the EVT to identify unknowns through reconstruction error. The open-set nearest neighbor method [23] recognized unknown samples using similarity scores between the nearest and second nearest classes. [24] proposed a three-way clustering approach for novelty detection. In recent years, deep learning approaches had shown great performances in closed-set classification tasks due to their powerful representation abilities. In the origin of deep open-set methods [11], Openmax was proposed to detect unknown classes by modeling the distance between output logits and activation vectors with a Weibull distribution model. [25] replaced the softmax layer in the network with a k-sigmoid layer to train the neural network. Similarly, CROSR [26] shared the idea of reconstruction-based representation, which models the class-belongingness by combining classification logits and latent representations. However, these supervised models only segment the infinite latent space for known classes, leaving no area for unknown classes.

Generative Methods. Apart from discriminative methods, generative models further take distributional information into consideration, which can be divided into two types. The first branch obtains a robust classifier by generating unknown instances. With the help of synthetic unknown instances, the classifier was able to learn a more explicit representation for partitioning known and unknown classes such as G-Openmax [12] and OSRCI [13]. However, the above methods were restricted to the quality and variety of the generated unknown instances. Another type of generative method solved the open-set recognition problem by modeling the inherent distribution of known classes. The generative models such as variational auto-encoder (VAE) were widely used in this type of method. For example, C2AE [14] trained a class conditional auto-encoder in two steps (closed-set training and open-set training), and the unknown samples were rejected by reconstruction errors based on the Extreme Value Theory. Additionally, a conditional Gaussian distribution learning (CGDL) method was proposed in [15], which modeled the bottleneck representation of the input samples by forcing latent features of each class to approximate a specific Gaussian model. Unknown samples were detected by both Gaussian probability and reconstruction errors. Similar to CGDL, CVAECapOSR [27] was also based on a CVAE framework and learned the latent features by capsule networks. [16] introduced a Gaussian mixture variational auto-encoder method for open-set recognition, which cooperatively learned reconstruction and performed class-based Gaussian mixture clustering in the latent space. However, these VAE-based architectures lack discriminative constraints to maintain the performance of known classification.

Hybrid Methods. Combining both generative and discriminative methods, several hybrid methods have been presented more recently. [17] proposed a framework called OpenHybrid, which included an encoder, a classifier, and a flow-based density estimator. In this architecture, the flow-based density estimator was used to detect whether a sample belongs to the unknown category. In [18], a convolutional prototype network (CPN) was proposed, which learned a CNN feature extractor and derivable prototypes for known classes representations end to end with the combination of generative and discriminative losses. [28] presented the concept of reciprocal points which potentially represent the extra-class space corresponding to each known category. Additionally, GFROR [29] trained a generative model to represent known classes and utilized a self-supervision method to separate the classes in the latent space. PROSER [30] learned data and classifier placeholders for the unknown classes. Nevertheless, these hybrid methods were heuristic and lacked a strong theoretical foundation for distribution modeling.

2.2. Novelty Detection

Unknown detection in open-set recognition is similar as the novelty detection task (also called outlier detection), which aims to detect outlier samples from the normal ones. Traditional novelty detection methods were mainly based on support vector machine (SVM) [31] or isolation forest [32]. In recent years, methods based on the deep neural network have been more applied for novelty detection tasks. The auto-encoder based approaches [33,34] were widely used, in which the auto-encoders were utilized to extract the common latent representations from normal samples, and novelties were then detected by reconstruction errors. Apart from auto-encoders, some studies used generative adversarial networks (GANs) for novelty detection. For example, [35] utilized GANs to generate the potential out-of-distribution samples which helped the model to learn robust discriminative boundaries. MANomaly [36] designed a mutual adversarial network for the network intrusion detection. For some discriminative methods, [37] first detected the novelty samples through the output softmax probabilities and demonstrated that the novelties tended to be related to a lower maximum softmax probability. ODIN [38] improved the novelty detection performance with the technique of temperature scaling and input perturbations. [6] learned the inlier distribution by adversarial training and used confidence estimation for more effective novelty detection. The existing novelty detection tasks only aimed to distinguish unknown novelty samples from known samples, with no need to classify known classes. But in open-set recognition, the capability of detecting unknown samples and discriminating known samples are both required.

2.3. Prototype Learning

Prototypes stand for the representative exemplars [39,40] or feature vectors of a category [41,42], representing the category distribution in the instance or feature space. The best-known prototype learning method is k-nearest-neighbor (KNN). To save storage space and improve computation efficiency, [39] proposed a better method called learning vector quantization (LVQ). In recent years, prototype learning was more combined with feature learning through neural network models. Under this framework, the prototypes can be regarded as learnable parameters which are optimized by self-defined loss functions. [41] proposed a center loss to learn centers for deep feature representation of each category and intra-class variance reduction. GCPL [42] designed several prototype losses to learn the representative and discriminative prototypes, which were meanwhile used as a regularization to improve the intra-class compactness. [43] extracted the prototypes at multiple levels of granularity with local optimization for the classification task. [44] proposed a graph prototypical contrastive learning approach for unsupervised graph representation learning. However, most of the existing methods set the prototypes in the form of feature vectors and only learned the prototypes by discriminative loss. In our proposed method, each prototype is treated as a Gaussian distribution rather than a feature vector. These Gaussian prototypes are learned in a probabilistic way with both generative and discriminative constraints to guarantee the representation ability of feature distribution in the latent space.

3. Preliminaries

3.1. Open-Set Recognition

Given training set $D_{train} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with n labeled instances and N known classes, $y_i \in \{1, \ldots, N\}$ is the label of data x_i. In the testing phase, there is a larger amount of test data, $D_{test} = \{(x_1, y_1), \ldots, (x_t, y_t)\}$. The label of the testing set belongs to $\{1, \ldots, N, N+1\}$, where the category $N+1$ is the label of potential unknown data. Note that the unknowns could include different classes and their specific classes are not focused in the open-set recognition task. Therefore all unknown classes are represented by the same category $N+1$.

3.2. Variational Auto-encoder

We briefly review the terminology and notation of VAE [45] before introducing the proposed method. We denote the observation as x and the latent variable as z. In VAE, the encoding process maps the input information into a constrained distribution in the latent space, while the decoding process tries to reconstruct the input observation.

From the perspective of Bayesian inference, the decoder models the probabilistic generative processes of x given the continuous representation z. The generation process first generates a latent variable z from the prior distribution $p_\theta(z)$, and then samples the data from the generative distribution $p_\theta(x|z)$, where θ denotes the model parameters. We approximate the posterior distribution $p_\theta(z|x)$ by another distribution $q_\phi(z|x)$, which is computed by the probabilistic encoder with parameter ϕ. The decoder $p_\theta(x|z)$ and encoder $q_\phi(z|x)$ are learned simultaneously by maximizing the evidence lower bound (ELBO):

$$L_{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL\left(q_\phi\left(z|x\right)\|p\left(z\right)\right), \tag{1}$$

in which the first term is the reconstruction of x by maximizing the log-likelihood $\log p_\theta(x|z)$ with sampling from $q_\phi(z|x)$, and the second term regularizes the latent variable z by minimizing the KL divergence between the approximated posterior and the prior of the latent variable.

A typical choice for the prior $p_\theta(z)$ is the standard Gaussian distribution $\mathcal{N}(0,1)$. The decoder $p_\theta(x|z)$ and encoder $q_\phi(z|x)$ are fit through two deep networks. And the KL divergence can be calculated as:

$$-KL\left(q_\phi(z|x)\|p_\theta(z)\right) = -\frac{1}{2}\sum_{i=1}^{d}\left(\mu_i^2 + \sigma_i^2 - \log\sigma_i^2 - 1\right), \tag{2}$$

where d is the dimension of the latent space, and μ_i, σ_i denote the ith dimension value of vector μ and σ, respectively.

3.3. Conditional Variational Auto-encoder

Directly derived from the VAE model, conditional variational auto-encoder (CVAE) aims to approximate the conditional distribution $p(x|y)$, where y is the class information of x. Similar to vanilla VAE, CVAE also consists of an encoder and a decoder, and while the encoder maps the input x and the class y to a prefixed distribution over the latent variable z with label information, the decoder aims to reconstruct the input samples x of a given category y by z. By adding the condition y to the vanilla VAE, the ELBO of CVAE can be obtained directly as:

$$L_{ELBO} = \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|z,y)] - KL\left(q_\phi\left(z|x,y\right)\|p\left(z|y\right)\right), \tag{3}$$

CVAE is not a specific architecture but a collection of methods that varies in the way of introducing the label information with different purposes, such as [46,47]. One simple approach is to introduce class information into the latent space, where a unique conditional prior is assigned to each class. To this end, $p(z|x)$ is constrained to a Gaussian prior $\mathcal{N}(\mu^y, \mathrm{I})$ by KL divergence, in which the mean μ^y is a variable and can be trained by the model. In this way, the KL divergence term of CVAE can be calculated as:

$$-KL(q_\phi(z|x,y)\|p_\theta(z|y)) = -\frac{1}{2}\sum_{i=1}^{d}\left[(\mu_i - \mu_i^y)^2 + \sigma_i^2 - \log\sigma_i^2 - 1\right]. \tag{4}$$

Our proposed method is based on the above theoretical framework, and the multiple Gaussian prototypes prior to effective open-set recognition is further introduced and thoroughly discussed in the following section.

4. Proposed Method

4.1. Overview

In this section, the proposed method is introduced and elaborated. In view of the problems that appeared in the existing open-set recognition methods, we propose *Multiple Gaussian Prototype Learning* (MGPL). As shown in Fig. 2, our proposed architecture includes an encoder, a decoder, and multiple Gaussian prototypes in the middle for training. Similar to CVAE, the encoder acts as a feature extractor that encodes the original distribution into the latent space, and the decoder samples from the latent space to generate samples in the original space. The Gaussian prototypes can be regarded as representations of each class in the latent space. Different from other existing methods, there is no additional fully-connected (FC) network classifier in our proposed architecture.

Generally, we consider a generalized negative log-likelihood objective to model the joint distribution of the known samples $p_\theta(\boldsymbol{x}, \boldsymbol{y})$ as follows:

$$\mathcal{L}_{\text{MGPL}} = \mathbb{E}_{x,y\sim\mathcal{D}}[-\log p_\theta(x,y)]. \tag{5}$$

Then, the log of the joint distribution $\log p_\theta(\boldsymbol{x}, \boldsymbol{y})$ is decomposed into:

$$\log p_\theta(x,y) = \lambda \log p_\theta(x|y)p(y) + (1-\lambda)\log p_\theta(y|x)p(x). \tag{6}$$

This decomposition naturally combines the two constrain terms into a hybrid objective, where the preceding generative constrain term concerns the generation task $\log p_\theta(\boldsymbol{x}|\boldsymbol{y})$ given the label $\boldsymbol{y}$, and the following discriminative constrain term concerns the classification

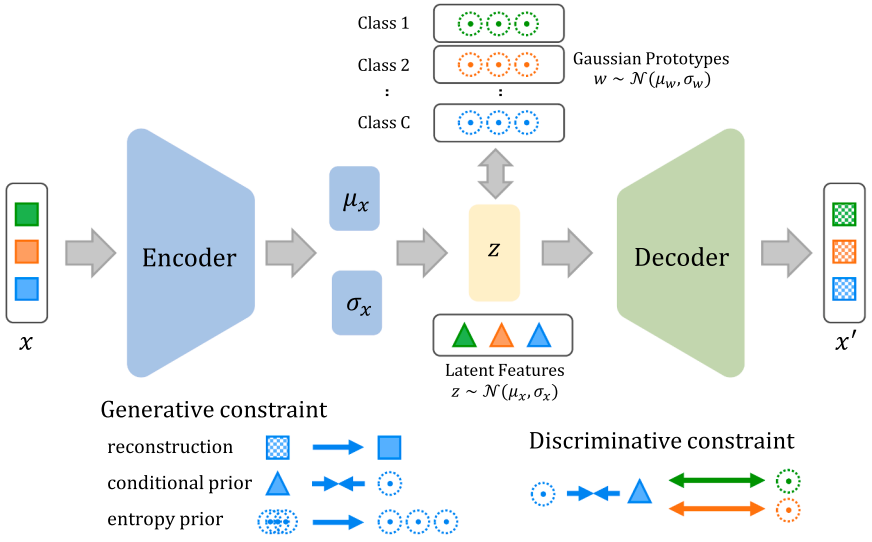


Fig. 2. Proposed MGPL model. The framework contains an encoder network, a decoder network, and multiple Gaussian prototypes that are learned to represent the class distribution in the latent feature space. The whole model is optimized under both generative and discriminative constraints derived by Bayesian inference. Under the generative constraint, the reconstruction term guides the generated sample to be similar to the input sample, the conditional prior term reduces the distance between the sample features and the Gaussian prototypes of the same class, and the entropy prior term prevents the multiple prototypes that represent the same class from collapsing. Meanwhile, the discriminative constraint drives the sample features away from the Gaussian prototypes of different classes and closer to the ones of the same class.

task $\log p_\theta(y|x)$. The hyper-parameter λ balances the influence between the two constraints.

Finally, by combining Eq. 5 and 6, we obtain the objective of our proposed method:

$$\mathscr{L}_{\text{MGPL}} \approx \lambda \underbrace{\mathbb{E}_{x,y\sim\mathscr{D}}[-\log p_\theta(x|y)]}_{\mathscr{L}_{\text{gen}}} + \left(1-\lambda\right) \underbrace{\mathbb{E}_{x,y\sim\mathscr{D}}[-\log q_\phi(y|x)]}_{\mathscr{L}_{\text{dis}}}. \tag{7}$$

Note that $\log p_\theta(x|y)$ is intractable to compute, so it is approximated with variational distribution $\log q_\phi(x|y)$. The $\log p(x)$ and $\log p(y)$ terms are also omitted because their expectations are constants with no model parameters. Detailed derivations of the generative and discriminative constraints are presented in the following subsections.

4.2. Gaussian Prototype

A Gaussian prototype w is a tuple (μ_w, σ_w, y) where y is the class label and μ_w, σ_w are the mean and deviation of the Gaussian distribution of the prototype defined as $\mathcal{N}(\mu_w, \sigma_w)$. The Gaussian prototypes can be regarded as representations of the distribution of each class in the latent space. Each class distribution is represented by K Gaussian prototypes, resulting in $K \times C$ prototypes in total, where C is the number of known classes. In the proposed method, the mean μ_w is a trainable parameter, the deviation σ_w is set to an identity matrix I for simplification, and the corresponding class y is predefined and fixed.

Previous methods [11,13,26] represent the distribution of known classes by calculating the mean activate vectors of each class after the training phase. Then the Euclidean distances between the test sample and the mean activate vectors are computed as the confidence score. However, the mean activate vectors might not be qualified representations of latent space distribution, for they are only suitable to estimate the Gaussian-like distribution, and the real distribution of the activate vectors of each class is not usually constrained to be a Gaussian distribution. Moreover, computing only one mean activate vector is insufficient to represent the complex and arbitrary feature space. Furthermore, the mean activate vector is merely a computed statistical value after the testing phase, with no reliable optimization constraints during the training.

In our proposed method, the latent space is constrained to be composed of several Gaussian regions of different classes, with each region represented by multiple Gaussian prototypes. Different from the previous CVAE-like methods in which only one Gaussian prototype is assigned to each class, our proposed method utilizes multiple Gaussian prototypes to handle more complex data distributions. The mapping relationship between the classes and the Gaussian prototypes is one-to-many, that is, a Gaussian prototype represents a specific class, yet a class can be represented by more than one Gaussian prototype due to the complex distributions of the input data.

The Gaussian prototypes can also be regarded as the latent variables in Eq. 7 under both generative and discriminative constraints via the Bayesian inference, as demonstrated in Fig. 3. The discriminative constraint makes the Gaussian prototype directly available for classification tasks with no need for additional classifiers. In the implementation, each Gaussian prototype is formed by two d-dimensional vectors with parameters including μ_w and σ_w, where d is the dimension of the predefined latent space. These parameters are randomly initialized and then trained with the whole network architecture end-to-end, which are updated by optimization methods based on gradient descent.

4.3. Generative Constraint

The generative constraint aims to approximate $\log p_\theta(x|y)$ to get a more effective feature representation in a generative way. General CVAE models only embed each class into one specific Gaussian distribution. However, one Gaussian prototype for each class in the latent space may not be sufficient for input distributions and tasks with higher complexity. Therefore, our proposed method represents each class with a mixture of Gaussian distributions, i.e. multiple Gaussian prototypes. To this end, a modified variational inference method is proposed below.

Two types of variables are included in our method: observable variables and latent variables. Observable variables include input sample x and output category y. Different from the previous CVAE frameworks, there are two latent variables z and w in our proposed

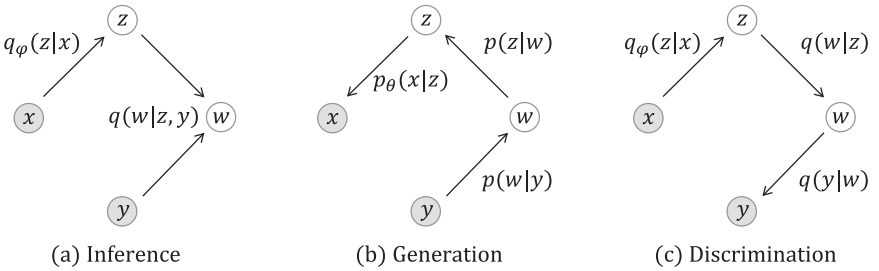


Fig. 3. Graphical models of Bayesian inference for MGPL. The generative constraint contains inference process in (a) and generation process in (b) that handles conditional generation. The discriminative constraint can be explained by the discrimination process in (c) that guides the classification.

method. z is a continuous variable representing the encoding feature vectors and w is a discrete variable representing the Gaussian prototype.

Similar to many VAE based methods, we approximate $p_\theta(x|y)$ with a variational Bayesian method and get the ELBO function as:

$$\begin{aligned}\log p_\theta(x|y) &= \log \sum_w \int p_\theta(x,z,w|y)dz \\ &= \log \sum_w \int q_\phi(z,w|x,y)\frac{p_\theta(x,z,w|y)}{q_\phi(z,w|x,y)}dz \\ &= \log \mathbb{E}_{q_\phi(z,w|x,y)}\left[\frac{p_\theta(x,z,w|y)}{q_\phi(z,w|x,y)}\right] \\ &\geqslant \mathbb{E}_{q_\phi(z,w|x,y)}\left[\log\frac{p_\theta(x,z,w|y)}{q_\phi(z,w|x,y)}\right] = L_{ELBO}\end{aligned} \tag{8}$$

where $q_\phi(z,w|x,y)$ is the inference process and $p_\theta(x,z,w|y)$ is the generation process.

As shown in Fig. 3(a)(b), the inference and generation process are further decomposed as:

$$\begin{aligned}q_\phi(z,w|x,y) &= q(w|z,y)q_\phi(z|x) \\ p_\theta(x,z,w|y) &= p(w|y)p(z|w)p_\theta(x|z).\end{aligned} \tag{9}$$

In the generation process, $p(w|y)$ is the prototype prior in a given class y, which is set to be uniform distribution equal to $1/K$ for each prototype for balance. $p(z|w)$ follows a Gaussian distribution $\mathcal{N}(z;\mu_w,I)$. Same as previous VAE frameworks, $p_\theta(x|z)$ is approximated by the decoder network. A sample of $p(x)$ can be generated through the following procedure. First, for a given category y, select a Gaussian prototype w of this category with probability $p(w|y)$. Then, draw z from this Gaussian prototype by $p(z|w)$. And finally, an original sample x can be generated from the decoder $q_\phi(x|z)$.

In the inference process, $p_\theta(z|x)$ is the latent feature that forms as a Gaussian distribution. This is approximated by the encoder network with the reparameterization trick. $q(w|z,y)$ is a categorical distribution which computes the assignment probability from z to the K Gaussian prototypes w of label y by distance, defined as:

$$q(w|z,y) := q(z \in w_{ij}|y) = \frac{e^{-\gamma d(z,w_j)}}{\sum_{l=1}^{K} e^{-\gamma d(z,w_{yl})}}, \tag{10}$$

where γ is the temperature parameter for smoothing the output probability and w_{yl} denotes the Gaussian prototypes with class label y. $d(\cdot)$ stands for the distance between z and w. Considering that both z and w are Gaussian distributions, we use KL divergence to measure the distance between z and w, defined as:

$$d(z,w) = KL(\mathcal{N}(\mu_x,\sigma_x)||\mathcal{N}(\mu_w,I)). \tag{11}$$

Then the ELBO of generative constraint can be given by:

$$\begin{aligned}L_{ELBO} &= \mathbb{E}_{q_\phi(z,w|x,y)}\left[\log\frac{p_\theta(x,z,w|y)}{q_\phi(z,w|x,y)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)q(w|z,y)}\left[\log\frac{p_\theta(x|z)p(z|w)p(w|y)}{q_\phi(z|x)q(w|z,y)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathbb{E}_{q(w|z,y)}\left[KL\left(q_\phi(z|x)\|p(z|w)\right)\right] - \mathbb{E}_{q_\phi(z|x)}[KL(q(w|z,y)\|p(w|y))],\end{aligned} \tag{12}$$

and the three terms of L_{ELBO} are explained below.

The first term, i.e. reconstruction term, is defined as:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] = \|\widehat{x}-x\|_2, \tag{13}$$

where $\widehat{x}$ is the reconstructed sample. This term is calculated by minimizing the reconstruction error of the generated sample. In this way, the latent variable z can keep more original information by generation to benefit open-set detection.

The second term, i.e. prototype conditional prior term, is defined as:

$$\begin{aligned}\mathbb{E}_{q(w|z,y)}\left[KL\left(q_\phi(z|x)\middle\|p(z|w)\right)\right] &= \mathbb{E}_{q(w|z,y)}[KL(\mathcal{N}(z;\mu_x,\sigma_x)\|\mathcal{N}(z;\mu_w,I))] \\ &= \mathbb{E}_{q(w|z,y)}\sum_{i=1}^{d}\frac{1}{2}\left[\left(\mu_i(x)-\mu_i^w\right)^2+\sigma_i^2-\log\sigma_i^2-1\right].\end{aligned} \tag{14}$$

This term enforces the latent variable z to fit its conditional Gaussian prototype prior by KL divergence, meaning that z should be close to and gather into clusters around the Gaussian prototypes of its category.

The third term is a prototype entropy prior term. Note that $p(w|y) = 1/K$ is for all prototypes w of label y, so this term is defined as:

$$\begin{aligned}\mathbb{E}_{q_\phi(z|x)}[KL(q(w|z,y)\|p(w|y))] &= \mathbb{E}_{q_\phi(z|x)}\sum_{w\in W_y}\left[-q\left(w\middle|z,y\right)\log\frac{q(w|z,y)}{p(w|y)}\right], \\ &= \mathbb{E}_{q_\phi(z|x)}\sum_{w\in W_y}[-q(w|z,y)\log K(q(w|z,y))].\end{aligned} \tag{15}$$

This term aims to force the prototype probabilities of each class into a uniform distribution. In other words, it aims to maximize the entropy of the K prototypes that represent the same class, and prevent these multiple Gaussian prototypes from collapsing to one.

4.4. Discriminative Constraint

Intuitively, the generative constraint pushes the latent variables closer to the Gaussian prototypes of the corresponding category. To further improve the discrimination ability, the discriminative constraint is set to separate the feature and the Gaussian prototypes of different classes in the latent space. Same as the generation process, the discrimination process can also be explained by the Bayesian inference framework.

The purpose of the class discrimination task is to maximize the posterior probability from x to y on the correct class, which is defined as:

$$\begin{aligned}\log q_\phi(y|x) &= \log\sum_w\int q_\phi(y,w,z|x)dz \\ &= \log\sum_w\int q(y|w)q(w|z)q_\phi(z|x)dz,\end{aligned} \tag{16}$$

where $q_\phi(z|x)$ is the latent feature forms as a Gaussian distribution, same as the common VAEs, and is approximated by the encoder network and optimized by the reparameterization trick. $q(y|w)$ is a known and fixed probability because each Gaussian prototype w has its own corresponding class y. Specifically, like an indicator function, $q(y|w)$ equals to 1 for the corresponding class and 0 for the other classes.

As for $q(w|z)$, it can be regarded as the probability that assigns the latent variable z to the Gaussian prototype w. Therefore, it is natural that closer Gaussian prototypes obtain higher probabilities. Note that both the latent variable z and the Gaussian prototype w follow the Gaussian distribution, so it is simple to use KL divergence to measure the distances from a latent variable to the Gaussian prototypes. By normalizing these distances, $q(w|z)$ can be computed as:

$$q(w|z) := q(z\in w_{ij}) = \frac{e^{-\gamma d\left(z,w_{ij}\right)}}{\sum_{k=1}^{C}\sum_{l=1}^{K}e^{-\gamma d(z,w_{kl})}}, \tag{17}$$

where $d(\cdot)$ is the KL divergence and γ is the temperature parameter for smoothing the output probability. K is the number of Gaussian prototypes for each class and C is the total number of known classes. Note that $q(w|z)$ computes the assignment probability of all $K \times C$ Gaussian prototypes, but $q(w|z,y)$ in Eq. 10 only computes the assignment probability of the K Gaussian prototypes with the same label y.

Substitute Eq. 17 into Eq. 16, and the discriminative constraint can be obtained:

$$\log q_\phi(y|x) = \log\frac{\sum_{j=1}^{K}e^{-\gamma d\left(z,w_{yj}\right)}}{\sum_{k=1}^{C}\sum_{l=1}^{K}e^{-\gamma d(z,w_{kl})}}, \quad z\sim q_\phi(z|x). \tag{18}$$

In summary, minimizing discriminative constraints will decrease the distances from the instance features to prototypes with the corresponding category, and increase the distances from the instance features to prototypes with incorrect categories, which makes prototypes with different categories more separated from each other.

4.5. Testing

After trained by both generative and discriminative constraints, one encoder and decoder network that approximates $q_\phi(z|x)$ and $p_\theta(x|z)$ can be obtained. Meanwhile, the parameters of Gaussian prototypes w which represent the distribution of each class in the latent space are also obtained.

Unlike other methods that are in need of an additional classifier, in the proposed method, Gaussian prototypes can be regarded as a natural discriminator for finding the nearest Gaussian prototype. Specifically, given an instance x, its latent variable z is first obtained by the encoder network $q_\phi(z|x)$. Then the nearest Gaussian prototype w_* can be obtained by comparing the distances from z to all the Gaussian prototypes w_{ij}:

$$w_* = \text{argmin}_{w_{ij}} \{ d(z, w_{ij}) \}, \quad z \sim q_\phi(z|x). \tag{19}$$

Finally, the input sample will be recognized as unknown if the minimum distance is over a given threshold τ, otherwise, it will be classified to the class of the nearest Gaussian prototype w_*:

$$y_{pred} = \begin{cases} Unknown, & \text{if} d(z, w_*) > \tau, \\ Class\ of\ w_*, & \text{otherwise}, \end{cases} \tag{20}$$

where τ is decided by ensuring 95% of data in the validation set to be recognized as known [15,29]. Since the model can well constrain the feature distribution of known classes into a certain region represented by the multiple Gaussian prototypes, it is an effective way for threshold estimation by choosing a majority 95% of samples in the validation set as known.

5. Experiment

In this section, the performance of the proposed method on benchmark datasets is evaluated and compared with the state-of-the-art methods. Same as the recent works in this area [29,27], our experiment settings are followed by the protocol in [13]. We first report the performance on unknown detection and open-set recognition tasks. Then we visualize the latent space and the confidence score with the CIFAR10 dataset to analyze the benefit of the proposed method. Meanwhile, an ablation study is conducted to analyze the contribution of each part stage by stage. Hyper-parameters and the execution time of our proposed method are also analyzed. The results show that our proposed method achieves very competitive performance and is reliable and effective on open-set recognition tasks.

5.1. Implementation Details

In the proposed method, we set the same hyper-parameter settings for all datasets. We use the Adam optimizer with a learning rate of 0.001 and fix the batch size to 32 in all training phases. Each class is assigned with three Gaussian prototypes ($K = 3$), which are initialized by the random normal distribution with a fixed dimension of 128. The balance parameter λ of constraint terms is set to 0.005. For the network architecture, a modified ResNet-18 network with batch normalization is used as the encoder and the decoder is the mirror structure. Following [27], we add a skip connection with random dropout between the middle layer of the encoder and decoder network. In this way, the shallow layers will focus more on the detail reconstruction while the deep layers focus more on learning the latent space. Random center-cropped and random horizon flip are used as data augmentation except for MNIST. All experiments are conducted on a computer with Intel E5-2678 2.5 GHz CPU, 32 GB RAM, and Nvidia RTX 3090 GPU for computational acceleration.

5.2. Unknown Detection

During the experiment, we validate the ability of our proposed method on detecting unknown samples. For a fair comparison, our experimental setting follows the same protocol defined in [13], where only several classes are selected from a dataset to train the model, while the rest are considered as unknown samples. Several benchmark datasets are used for evaluation, including MNIST, SVHN, CIFAR10, and Tiny-ImageNet. MNIST, SVHN, and CIFAR10 datasets all contain 10 classes, and 6 classes are randomly selected as known samples to train the model, with the 4 remaining classes as unknown samples. Moreover, with 4 classes in the CIFAR10 dataset sampled as the known-set classes, 10 or 50 classes from the CIFAR100 dataset are randomly selected as open-set classes, which is reported as CIFAR + 10 or CIFAR + 50. For Tiny-ImageNet, which is a subset of ImageNet [48] with 200 classes, we randomly select 20 classes as known classes with the rest as unknowns. As is a binary recognition task, the Area Under the Receiver Operator Characteristic curve (AUROC) score is used to measure the performance of known and unknown detection.

In the unknown detection task, for each dataset, 5 combinations of known and unknown classes splits are selected and the results are reported by the average of these different splits, where the value before $\pm$ represents the mean and the value after $\pm$ represents the standard deviation. However, as recently discussed in [29], performance across different splits varies significantly (e.g. AUROC on CIFAR10 varied from 0.77 to 0.87 across different splits). We consider the reason is that different class split brings different levels of difficulty for open-set recognition tasks, which leads to serious problems in reproducibility and comparison. Therefore, we report the results that are evaluated by the same known-unknown splits in [13] from [13,?,?,?]. For the work that uses different classes splits

[30], we re-run the released public code of the original paper with the same class splits in [13]. In this way, our proposed method is compared with the previous methods under the same evaluation settings to guarantee the fairness and reproducibility. As shown in Table 1, compared with previous works, our proposed method achieves the best performance on all datasets.

5.3. Open-Set Recognition

In open-set recognition, the classifier needs to not only reject unknown samples correctly but also have a good performance on known classification. Therefore, we present the comparison of closed-set accuracy between the plain CNN (closed-set classifier) and the proposed method MGPL in Table 2. For a fair comparison, the plain CNN in Table 2 has the same backbone architecture (ResNet-18) as the encoder of the proposed MGPL method. With this experimental setting, we aim to compare the closed-set classification accuracy between the plain CNN and the proposed method. For the plain CNN, a fully connected layer with softmax is added to the Resnet-18 feature extraction encoder for classification. As for the proposed method, the Gaussian prototypes are utilized directly to replace the fully-connected layer classifier for the classification task after the ResNet-18 encoder. Although MGPL is an open-set classification method, its discriminative ability is not decreased or even better than the closed-set classifier on some datasets.

Under the following experimental settings, the performance of our proposed method is validated with open-set recognition tasks including both known classification and unknown detection. The model is trained on all the training data of one dataset, but during the testing phase, the test sets are added with unknown samples from another dataset. The open-set recognition performance is measured by the macro-averaged F1-scores over all known classes and the unknown class, with the MNIST and CIFAR10 datasets used as known samples for training.

In the first experiment, MNIST is considered as the training dataset, which is a very common dataset consisting of handwritten digits of 0–9. Following the setting in [26], we choose three datasets, Omniglot [49], MNIST-Noise and Noise, as open-set samples. Omniglot is a dataset containing various alphabet characters with images of 28×28 grayscale similar to MNIST. MNIST-Noise is synthesized by adding noise to MNIST testing samples. Noise is synthesized by randomly sampling each pixel value independently from a uniform distribution on [0, 1]. Examples of these datasets are shown in Fig. 4. Same as MNIST, each open dataset contains 10,000 test samples, with a known-to-unknown ratio of 1:1. The results are shown in Table 3, and it can be observed that our proposed method achieves the best performance on all given datasets.

In the second experiment, samples of all classes in CIFAR10 are chosen as known samples, with images of vehicles and animals. Following the setting in [26], samples in ImageNet [48] and LSUN [50] dataset are selected as unknown samples. To keep the same size as known samples, unknown samples are resized or cropped to generate four open datasets including ImageNet-crop, ImageNet-resize, LSUN-crop, and LSUN-resize. Same as the test set of CIFAR10, each open dataset contains 10000 samples. In this way, the known-to-unknown ratio is set to 1:1. We evaluate the open-set recognition performance with macro-averaged F1-scores between 10 known classes and 1 unknown class, and the results are shown in Table 4. Although a small misperform can be observed in one dataset compared to the best method, our proposed method leads the best performance in the rest three datasets. Moreover, it can also be observed from Table 4 that the performance of LSUN is very close to the recent methods, but there is an obvious increase for ImageNet. We consider the reason is that ImageNet is a more complex dataset than LSUN and requires more of an algorithm's ability to learn effective representations for the open-set recognition task, which means there is still room for improvement.

5.4. Visualization

In this section, we conduct two types of visualization with the CIFAR10 dataset. The CIFAR10 dataset contains 10 classes, in which 6 animals classes (bird, cat, deer, dog, frog, and horse) are considered to be known classes and 4 vehicle classes (airplane, car, ship, and truck) to be open-set classes. The performance of a plain CNN network is compared with the proposed method.

First, in Fig. 5, we show the learned latent space of a plain CNN and the proposed method. The high-dimensional latent features are reduced to 2D by T-SNE for visualization. Fig. 5(a) shows that in the latent feature space of the plain CNN, features of open-set and

Table 1
Unknown detection performance in terms of the Area Under the Receiver Operator Characteristic Curve (AUROC) score. Results are averaged among 5 different splits of known and unknown classes. As discussed in Section 5.2, all the results are evaluated on the same class splits for a fair comparison. N.R. is used when there is no particular value in the original paper.

Methods	SVHN	CIFAR10	CIFAR + 10	CIFAR + 50	Tiny-ImageNet
Softmax	0.886	0.677	0.816	0.805	0.577
OpenMax [11]	0.894	0.695	0.817	0.796	0.576
G-OpenMax [12]	0.896	0.675	0.827	0.819	0.58
OSRCI [13]	0.91 ± 0.01	0.699 ± 0.038	0.838	0.827	0.586
CROSR [26]	0.899 ± 0.018	N.R.	N.R.	N.R.	0.589
C2AE [14]	0.892 ± 0.013	0.711 ± 0.008	0.810 ± 0.005	0.803 ± 0.000	0.581 ± 0.019
CGDL [15]	0.896 ± 0.023	0.681 ± 0.029	0.794 ± 0.013	0.794 ± 0.003	0.653 ± 0.002
PRL [28]	0.931 ± 0.014	0.784 ± 0.025	0.885 ± 0.019	0.881 ± 0.014	0.711 ± 0.026
GFROR [29]	0.955 ± 0.018	0.831 ± 0.039	0.915 ± 0.002	0.913 ± 0.002	0.647 ± 0.012
CVAECapOSR [27]	0.956 ± 0.012	0.835 ± 0.023	0.888 ± 0.019	0.889 ± 0.017	0.715 ± 0.018
PROSER [30]	0.944 ± 0.016	0.791 ± 0.045	0.862 ± 0.006	0.852 ± 0.015	0.662 ± 0.005
MGPL (ours)	**0.957 ± 0.011**	**0.840 ± 0.021**	**0.927 ± 0.010**	**0.918 ± 0.003**	**0.730 ± 0.031**

Table 2
Comparison of closed-set accuracy between the plain CNN (closed-set classifier) and the proposed method MGPL with the same network backbone. Although MGPL aims at classifying known samples as well as learning Gaussian prototypes, there is no significant degradation in closed-set accuracy.

Architecture	MNIST	SVHN	CIFAR10	Tiny-ImageNet
Plain CNN	0.996	0.964	0.929	0.546
MGPL (ours)	**0.996**	**0.967**	**0.932**	**0.547**

Fig. 4. Dataset example of original MNIST, Omniglot, MNIST-noise, and Noise.

Table 3
Open-set classification results on MNIST dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F1-scores in 11 classes (10 known classes and 1 unknown class).

Methods	Omniglot	MNIST-noise	Noise
Softmax	0.595	0.801	0.829
OpenMax [11]	0.780	0.816	0.826
CROSR [26]	0.793	0.827	0.826
CGDL [15]	0.850	0.887	0.859
PROSER [30]	0.862	0.874	0.882
MGPL (ours)	**0.981**	**0.978**	**0.981**

Table 4
Open-set recognition results on CIFAR10 with various outliers added to the test set as unknowns. The performance is evaluated by macro F1-score in 11 classes (10 known classes and 1 unknown class).

Methods	ImageNet-crop	ImageNet-resize	LSUN-crop	LSUN-resize
Softmax	0.639	0.653	0.642	0.647
Openmax [11]	0.660	0.684	0.657	0.668
OSRCI [13]	0.636	0.635	0.650	0.648
CROSR [26]	0.721	0.735	0.720	0.749
C2AE [14]	0.837	0.826	0.783	0.801
CGDL [15]	0.840	0.832	0.806	0.812
RPL [28]	0.811	0.810	0.846	0.820
GFROSR [29]	0.821	0.777	0.843	0.784
CVAECapOSR [27]	0.857	0.834	0.868	**0.882**
PROSER [30]	0.848	0.824	0.867	0.856
MGPL (ours)	**0.862**	**0.862**	**0.869**	0.868

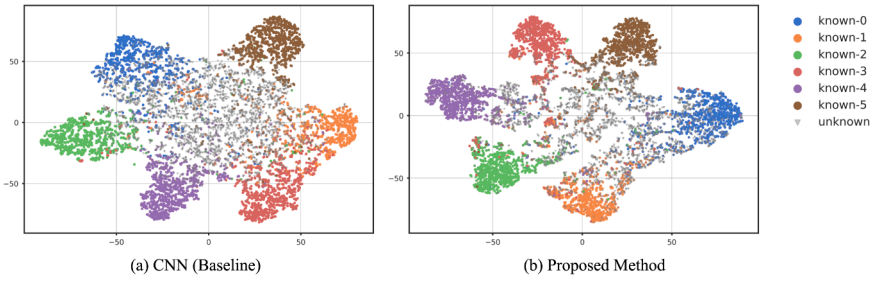


Fig. 5. Visualization of the latent feature space of (a) plain CNN and for (b) the proposed method. The proposed method learns a more compact and separatable latent space.

known-set samples are mixed together, which is difficult to detect open-set samples. However, the latent space of the proposed method shown in Fig. 5(b) indicates that with the help of multiple Gaussian prototypes, the open-set and known-set samples have fewer overlaps, and the known-set classes are more compact into clusters.

Then, we visualize the confidence score histograms of open-set and known-set samples generated by a plain CNN and the proposed method, as shown in Fig. 6(a)(b) respectively. For baseline CNN, we define the maximum output logit before the Softmax layer as the confidence score in the traditional way. For our proposed method, the maximum negative distance to the Gaussian prototypes is defined as the confidence score. We can infer from Fig. 6 that the confidence scores for the known-set samples gather more tightly in the proposed method. Therefore, the known-set and open-set samples are more separated in the proposed method. Meanwhile, the tightly-gathered known-set confidence score makes selection of the reject threshold much easier.

Furthermore, in Fig. 7, we compare the receiver operating characteristic (ROC) curve and the precision-recall (PR) curve between the plain CNN and the proposed method. It can be observed that the proposed method exhibits much better performance than the plain CNN at various threshold settings including low and very low false alarm regimes, indicating that the proposed method is able to detect unknown samples effectively and robustly.

In summary, the proposed method achieves a more compact and separated latent space for open-set recognition. The score distribution of the known-set samples is tighter and has fewer overlaps with that of the open-set samples, leading to better open-set rejection performance.

5.5. Ablation Study

In the ablation study, the contribution of each part is analyzed stage by stage from the baseline CNN to our proposed method. The ablation study is also conducted using the CIFAR10 dataset with the classes splits the same as in Section 5.4. The following cases are considered.

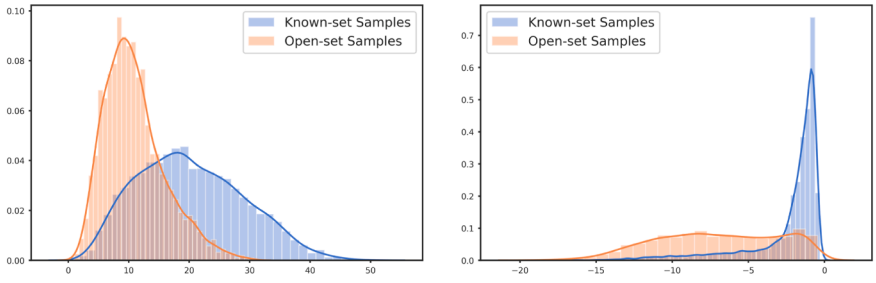
- **CNN + FC** uses a plain CNN as the feature extractor and a fully-connected (FC) network as the classifier. In the testing phase, the open-set scores are calculated through the maximization of the output logits of all known classes.
- **CVAE + FC** uses a conditional auto-encoder as the latent feature extractor under the generative constraint mentioned in Section 4.3. The latent features are then output to a fully-connected network for classification. During the testing phase, distances from the learned features to the center of all known classes are calculated, and the open-set score is defined by the minimum distance.
- **MGPL (Single Prototype)** replaces the FC classifier with the Gaussian prototype classifier, where each class is presented by one Gaussian prototype. The testing procedure is similar to "CVAE + FC" where the mean of the prototypes is regarded as the center of the corresponding class.
- **MGPL (Proposed)** extends the single prototype to multiple prototypes for a better representation of the latent feature space.

It can be observed from Table 5 that using CVAE as the generative constraint improves the open-set recognition ability of the plain CNN, but slightly decreases the classification performance. Meanwhile, the discrimination ability is maintained by replacing the FC classifier with the Gaussian prototype classifier. Moreover, utilizing more Gaussian prototypes to represent the latent space can further enhance the performance of closed-set classification and open-set rejection. The ablation study validates that each part of the proposed method contributes to the improvement of performance.

5.6. Hyper-Parameters Analysis

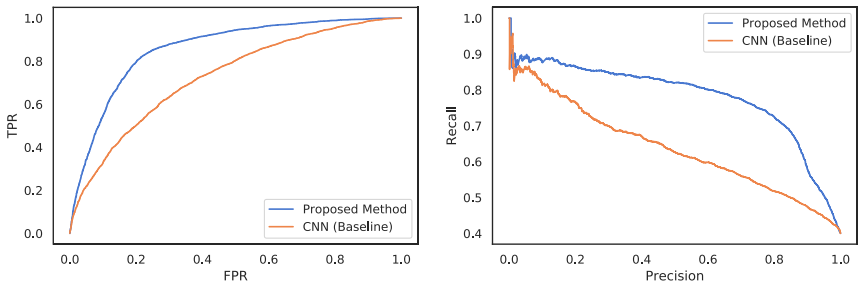
In this section, the influences of hyper-parameters are analyzed. There are two key hyper-parameters in the proposed method. The first one is parameter λ, the trade-off between the generative and discriminative terms in the loss function. The second one is parameter K, the number of the Gaussian prototypes of the same class. The hyper-parameters analysis is conducted using the CIFAR10 dataset with the classes splits the same as in Section 5.4, and the performance is measured by the AUROC and accuracy.

Fig. 8 shows the impact of λ. Naturally, increasing the value of λ raises the effect of the generative constraint, making the model



(a) CNN (Baseline) (b) Proposed Method

Fig. 6. Score histograms for open-set and known-set samples by (a) the plain CNN and (b) the proposed method. The score distribution of the proposed method are less overlapping and the known-set scores are tighter.

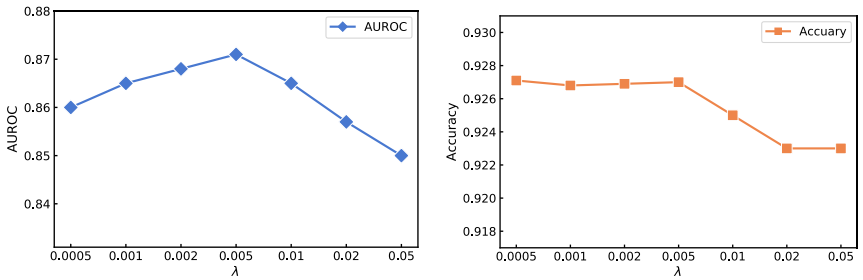


(a) ROC Curve (b) PR Curve

Fig. 7. Comparison between the plain CNN (orange line) and the proposed method (blue line) by visualizing ROC and PR curve. It is shown that the proposed method performs better than the plain CNN.

Table 5
Ablation study on the model architecture. We report the classification performance (accuracy) and open-set rejection performance (AUROC) with different model architectures.

Architecture	Classification Accuracy	Open-set Rejection (AUROC)
CNN + FC	0.924	0.818
CVAE + FC	0.917	0.854
MGPL (Single Prototype)	0.924	0.863
MGPL (Proposed)	**0.927**	**0.871**



(a) λ vs AUROC (b) λ vs Accuracy

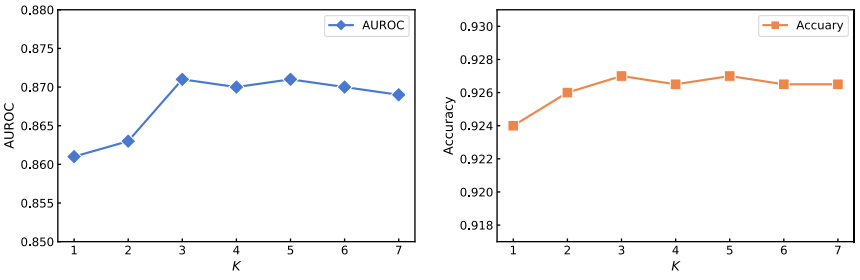
Fig. 8. The sensitivity analysis with λ in terms of (a) AUROC and (b) Accuracy.

focus more on the generation task rather than the classification task. The balance parameter λ is tuned from 0.0005 to 0.05. Raising λ in a small range from zero increases the AUROC score, which means the generative constraint will improve the unknown detection task. However, a high λ is most likely to harm the classification task and decrease the unknown detection performance. Therefore, the final suitable value of λ is chosen as 0.005 to bring the best performance for both known classification and unknown detection.

Fig. 9 shows the impact of K, which is tuned from 1 to 7. It can be observed that using multiple Gaussian prototypes rather than only one prototype increases the performance of both closed-set classification and open-set detection. However, the performance stops to improve continually and tends to stabilize when the number of Gaussian prototypes K passes 3. In addition, the powerful representation capabilities of deep neural networks also help with feature learning, so it is robust to tune the number of prototypes. Therefore, in consideration of improving the performance as well as reducing model complexity, the value of K is set to 3. More prototypes may be beneficial when the data distribution is too complex or hard to approximate.

5.7. Execution Time Analysis

To analyze the execution time, we compare the proposed method with related works in terms of the training time with the CIFAR10 dataset. It can be observed in Fig. 10 that although the proposed method is a hybrid model under both generative and discriminative constraints, our execution time is at the same level as the discriminative methods Softmax, Openmax [11] and the hybrid method Proser [30]. Compared to the generative methods OSRCI [13] and GFROSR [29], our proposed method consumes less time. As the most time-consuming method, OSRCI needs to first generate counterfactual images as novelties and then train the model with the initial dataset and the generated open-set instances. As the second time-consuming method, GFROSR feeds the reconstructed images to an



(a) K vs AUROC (b) K vs Accuracy

Fig. 9. The sensitivity analysis with K in terms of (a) AUROC and (b) Accuracy.

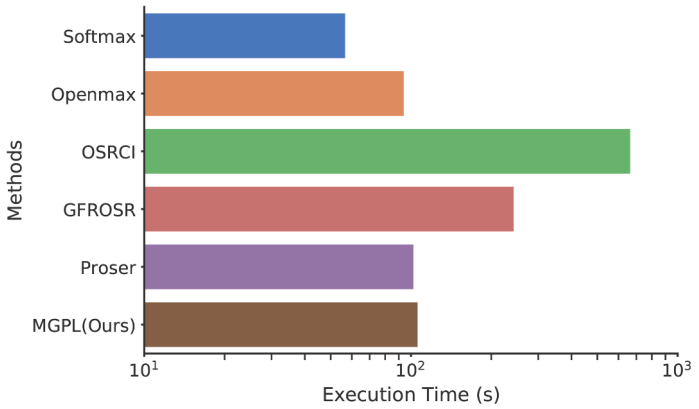


Fig. 10. Execution time for different methods on CIFAR10 dataset. The x-axis is in the logarithmic scale.

extra classification model. In contrast, our framework needs no extra training since both the generative and discriminative constraints are integrated into an entire framework and trained in a one-step way. As a result, our proposed method not only utilizes hybrid constraints for performance improvement of open-set recognition, but also holds an efficient training process.

6. Conclusion and Future Works

In this paper, we present a novel method named MGPL for open-set recognition, which aims to handle the unknown classes that do not exist in the training phase. In our proposed method, multiple Gaussian prototypes are learned to better represent the complex classes distribution in both generative and discriminative ways. The generative constraint guides the latent variables to compact around the Gaussian prototypes of the corresponding class, and the discriminative constraint separates the prototypes of different classes to improve the discrimination capability. More importantly, the entire model can be derived by Bayesian inference, providing theoretical support for open-set recognition tasks. Variable experiments on several benchmark datasets are conducted and satisfactory results are achieved. The experimental results prove that the proposed method can learn more effective latent representation with higher reliability and efficiency in open-set recognition tasks.

In future works, we will explore the means to discover and cluster the detected unknown samples into different categories, since all the unknown samples are grouped into the same novel class in the current method. In this way, not only can the known classes be classified, but also the newly appeared unknown classes can be detected and clustered. Furthermore, in the infinite data stream, to incrementally learn these novel class samples after discovering them might be an important focus for future research. In our proposed method, the class distribution can be well represented by the multiple Gaussian prototypes. Intuitively, these multiple Gaussian prototypes can be added or updated easily, which provides a good starting point for further research on open-set recognition in the evolving data stream.

CRediT authorship contribution statement

Jiaming Liu: Writing - original draft, Conceptualization, Methodology, Validation, Software. **Jun Tian:** Writing - review & editing, Validation. **Wei Han:** Visualization, Software. **Zhili Qin:** Visualization, Software. **Yulu Fan:** Writing - review & editing. **Junming Shao:** Writing - review & editing, Conceptualization, Supervision.

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.

Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities (ZYGX2019Z014), Sichuan Key Research Program (22ZDYF3388), National Natural Science Foundation of China (61976044, 52079026), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062), National Key Research and Development Program (2016YFB0502300) and Sichuan Science and Technology Program (2022YFG0260).

References

[1] M.I. Razzak, S. Naz, A. Zaib, Deep learning for medical image processing: Overview, challenges and the future, Classification in BioApps (2018) 323–350.
[2] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, J.A. Benediktsson, Deep learning for hyperspectral image classification: An overview, IEEE Transactions on Geoscience and Remote Sensing 57 (9) (2019) 6690–6709.
[3] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, Data mining and knowledge discovery 33 (4) (2019) 917–963.
[4] S.U. Din, J. Shao, Exploiting evolving micro-clusters for data stream classification with emerging class detection, Information Sciences 507 (2020) 404–420.
[5] W.J. Scheirer, A. de Rezende Rocha, A. Sapkota, T.E. Boult, Toward open set recognition, IEEE transactions on pattern analysis and machine intelligence 35 (7) (2012) 1757–1772.
[6] Y. Zhang, B. Zhou, X. Ding, J. Ouyang, X. Cai, J. Gao, X. Yuan, Adversarially learned one-class novelty detection with confidence estimation, Information Sciences 552 (2021) 48–64.
[7] S. Yang, W. Zhang, R. Tang, M. Zhang, Z. Huang, Approximate inferring with confidence predicting based on uncertain knowledge graph embedding, Information Sciences 609 (2022) 679–690.
[8] W. Hu, C. Chen, F. Ye, Z. Zheng, Y. Du, Learning deep discriminative representations with pseudo supervision for image clustering, Information Sciences 568 (2021) 199–215.
[9] F. Ye, A.G. Bors, Learning joint latent representations based on information maximization, Information Sciences 567 (2021) 216–236.
[10] M. Hein, M. Andriushchenko, J. Bitterwolf, Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 41–50.
[11] A. Bendale, T.E. Boult, Towards open set deep networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1563–1572.
[12] Z. Ge, S. Demyanov, Z. Chen, R. Garnavi, Generative openmax for multi-class open set classification, arXiv preprint arXiv:1707.07418.
[13] L. Neal, M. Olson, X. Fern, W.-K. Wong, F. Li, Open set learning with counterfactual images, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 613–628.
[14] P. Oza, V.M. Patel, C2ae: Class conditioned auto-encoder for open-set recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2307–2316.
[15] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, G. Peng, Conditional gaussian distribution learning for open set recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13480–13489.
[16] A. Cao, Y. Luo, D. Klabjan, Open-set recognition with gaussian mixture variational autoencoders, arXiv preprint arXiv:2006.02003.
[17] H. Zhang, A. Li, J. Guo, Y. Guo, Hybrid models for open set recognition, in: European Conference on Computer Vision, Springer, 2020, pp. 102–117.
[18] H.-M. Yang, X.-Y. Zhang, F. Yin, Q. Yang, C.-L. Liu, Convolutional prototype network for open set recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence.
[19] W.J. Scheirer, L.P. Jain, T.E. Boult, Probability models for open set recognition, IEEE transactions on pattern analysis and machine intelligence 36 (11) (2014) 2317–2324.
[20] L.P. Jain, W.J. Scheirer, T.E. Boult, Multi-class open set recognition using probability of inclusion, in: European Conference on Computer Vision, Springer, 2014, pp. 393–409.
[21] E.M. Rudd, L.P. Jain, W.J. Scheirer, T.E. Boult, The extreme value machine, IEEE transactions on pattern analysis and machine intelligence 40 (3) (2017) 762–768.
[22] H. Zhang, V.M. Patel, Sparse representation-based open set recognition, IEEE transactions on pattern analysis and machine intelligence 39 (8) (2016) 1690–1696.
[23] P.R.M. Júnior, R.M. De Souza, R.D.O. Werneck, B.V. Stein, D.V. Pazinato, W.R. de Almeida, O.A. Penatti, R.d.S. Torres, A. Rocha, Nearest neighbors distance ratio open-set classifier, Machine Learning 106 (3) (2017) 359–386.
[24] A. Shah, N. Azam, B. Ali, M.T. Khan, J. Yao, A three-way clustering approach for novelty detection, Information Sciences 569 (2021) 650–668.
[25] L. Shu, H. Xu, B. Liu, Doc: Deep open classification of text documents, arXiv preprint arXiv:1709.08716.
[26] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, T. Naemura, Classification-reconstruction learning for open-set recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4016–4025.
[27] Y. Guo, G. Camporese, W. Yang, A. Sperduti, L. Ballan, Conditional variational capsule network for open set recognition, arXiv preprint arXiv:2104.09159.
[28] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, Y. Tian, Learning open set network with discriminative reciprocal points, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, Springer, 2020, pp. 507–522.
[29] P. Perera, V.I. Morariu, R. Jain, V. Manjunatha, C. Wigington, V. Ordonez, V.M. Patel, Generative-discriminative feature representations for open-set recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11814–11823.
[30] D.-W. Zhou, H.-J. Ye, D.-C. Zhan, Learning placeholders for open-set recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4401–4410.
[31] M. Amer, M. Goldstein, S. Abdennadher, Enhancing one-class support vector machines for unsupervised anomaly detection, in: Proceedings of the ACM SIGKDD workshop on outlier detection and description, 2013, pp. 8–15.
[32] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (1) (2012) 1–39.
[33] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 665–674.
[34] X. Du, J. Yu, Z. Chu, L. Jin, J. Chen, Graph autoencoder-based unsupervised outlier detection, Information Sciences 608 (2022) 532–550.
[35] K. Lee, H. Lee, K. Lee, J. Shin, Training confidence-calibrated classifiers for detecting out-of-distribution samples, International Conference on Learning Representations.
[36] L. Zhang, X. Xie, K. Xiao, W. Bai, K. Liu, P. Dong, Manomaly: Mutual adversarial networks for semi-supervised anomaly detection, Information Sciences 611 (2022) 65–80.

[37] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, Proceedings of International Conference on Learning Representations.
[38] S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, International Conference on Learning Representations.
[39] T. Kohonen, Learning vector quantization, in: Self-organizing maps, Springer, 1995, pp. 175–189.
[40] D. Chen, Q. Yang, J. Liu, Z. Zeng, Selective prototype-based learning on concept-drifting data streams, Information Sciences 516 (2020) 20–32.
[41] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European conference on computer vision, Springer, 2016, pp. 499–515.
[42] H.-M. Yang, X.-Y. Zhang, F. Yin, C.-L. Liu, Robust classification with convolutional prototype learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3474–3482.
[43] X. Gu, M. Li, A multi-granularity locally optimal prototype-based approach for classification, Information Sciences 569 (2021) 157–183.
[44] M. Peng, X. Juan, Z. Li, Graph prototypical contrastive learning, Information Sciences 612 (2022) 816–834.
[45] D.P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.
[46] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, Advances in neural information processing systems 28 (2015) 3483–3491.
[47] Y. Li, Q. Pan, S. Wang, H. Peng, T. Yang, E. Cambria, Disentangled variational auto-encoder for semi-supervised learning, Information Sciences 482 (2019) 73–85.
[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (3) (2015) 211–252.
[49] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.
[50] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, J. Xiao, Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, arXiv preprint arXiv:1506.03365.