

# Learning Posterior and Prior for Uncertainty Modeling in Person Re-Identification

Yan Zhang, Zhilin Zheng, Binyu He, Li Sun

East China Normal University

**Abstract.** Data uncertainty in practical person reid is ubiquitous, hence it requires not only learning the discriminative features, but also modeling the uncertainty based on the input. This paper proposes to learn the sample posterior and the class prior distribution in the latent space, so that not only representative features but also the **uncertainty can be built by the model**. The prior reflects the distribution of all data in the same class, and it is the trainable model parameters. While the posterior is the probability density of a single sample, so it is actually the feature defined on the input. We assume that both of them are in Gaussian form. To simultaneously model them, we put forward a **distribution loss**, which measures the KL divergence from the posterior to the priors in the manner of supervised learning. In addition, we assume that the posterior variance, which is essentially the uncertainty, is supposed to have the second-order characteristic. Therefore, a  $\Sigma$ -net is proposed to compute it by the high order representation from its input. Extensive experiments have been carried out on Market1501, DukeMTMC, MARS and noisy dataset as well.

**Keywords:** ReID, Uncertainty Modeling, Noisy Label

## 1 Introduction

Person re-identification (reid) is a classic pedestrian retrieval task that aims to find a particular person across non-overlapping camera views [40]. Given a query person, reid algorithms find out whether the same person has appeared in the gallery, which contains a large amount of candidates who have emerged at some time, in another place or under a different camera. Usually, the query can be an image [39,24], a short video sequence [28,38] and even a text description [33,16]. As reid has a wide application in the surveillance and security system, it has arisen researchers attention for years. In the early days, handcrafted features are mainly employed [36,19,8,41], while nowadays the deep learning based methods [40,21,12,37,7,32,2] dominate in this area. [40] is the simplest baseline model using deep features learned from the ResNet50 backbone. Many works [12,37,7,32,2] extend it and use it for fair comparison. Recently, [21] summarizes the commonly used tricks for training, and releases another strong baseline model.

In spite of the tremendous works with different loss functions or network structures, reid is still challenging, due to the low resolutions [18,30], various

poses [6,35] or occlusions [12], etc. These difficulties increase the uncertainty for the identification results and harm the performance to a certain extent. Most of previous works concentrate on learning deterministic discriminative features for reid [7,32,26], but are lack of **uncertainty modeling from the data itself**. Some works model the data distribution to account the uncertainty [27,34,25], while they do not explicitly distinguish the class priors and the sample posteriors.

This paper proposes to learn the sample posterior and the class priors simultaneously in reid task, so that it quantifies the uncertainty of an input image and its corresponding class. The key idea is ~~to model the feature representation and its uncertainty by two types of probability distribution~~. The prior models the latent distributions for all samples in the given class. In practice, it is often treated as a set of static parameters. However, our work optimizes them together with other model parameters by back propagation. Moreover, we assume the **single Gaussian prior** for each class in this paper, which makes the whole setting easily to be expressed in closed form equation. The posterior is the distribution for a single input sample given by the model. It is a special feature using not only for the classification, but also for the uncertainty evaluation.

Particularly, we measure the distance between the posterior and prior by **Kullback-Leibler (KL) divergence**, which serves as the upper bound of the negative log likelihood of the random code, drawn from the posterior, with respect to the given prior. Inspired by GM loss [27], we formulate the class conditional probability by normalizing the negative KL divergence between the posterior and the prior indicated by the label, and compute the cross entropy loss. An extra regularization term is added to ensure the posterior to be closer to the corresponding class prior than others. Furthermore, we propose a structure named  **$\Sigma$ -net** to use a high order feature of its input as the variance, so that the mean and variance of the posterior can be distinguishable. We perform extensive experiments on the image based reid dataset Market-1501 [39] and DukeMTMC-reID [24] and the video dataset MARS [38], and demonstrate the effectiveness of the proposed method.

Our contributions are summarized as follows:

- We propose to learn the posterior and the prior to model the uncertainty based on the individual sample and all samples in the same class, respectively.
- We propose a novel loss function, named distribution loss, to build the connection between the posterior and the prior distribution.
- We design the  $\Sigma$ -net structure to model the sample uncertainty by computing the high order feature as the variance in the posterior .
- We show the effectiveness of our method on several well-known image-based datasets and a video-based dataset.

## 2 Related Works

**High order discriminative feature learning.** To learn discriminative features, many works realize the value of high order features. Non-local attention

is one way to exploit the second-order feature, which is also known as the attention mechanism [29]. [31] proposes an inserted non-local attention module to utilize the second order information in the median layers. [5] proposes a spatial and channel attention from two different directions in the similar way. [23] automatically searches the network structure, and the final results show that the non-local attention module is important for improving the performance.

Apart from the non-local operation, bilinear pooling [20,1] is another choice to formulate the high order features. Original bilinear pooling methods have large computation costs. Given the input tensor  $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ , it computes the outer product  $\mathbf{f}^T \mathbf{f}$ , in which  $\mathbf{f} = \mathbf{F}(i, j) \in \mathbb{R}^C$ , at each spatial coordinate  $(i, j)$ . Then a sum pooling is followed to get the second-order feature  $\mathbf{f}^{(2)} = \sum_{i,j} \mathbf{f}^T \mathbf{f} \in \mathbb{R}^{C^2}$ . [14] proposes to use Hadamard product to implement it and thus increases the efficiency. In this work,  $\mathbf{F}$  is first reduced to two "thin" tensors  $\mathbf{F}_1$  and  $\mathbf{F}_2$  with only a few channels. Then elementwise product is computed between them to form the second order feature. Similar works like [13,3] also proves the effectiveness of Hadamard product. All the above mentioned works validate the high order features, but they are not designed for uncertainty modeling.

**Uncertainty modeling by posterior and prior.** In real reid application, there are inevitable noises on the data. These noises are either on the pixels (*e.g.* occlusions, blurs) or on the training labels (*e.g.* ID switch in the tracker). Modeling the data uncertainty mainly intends to deal with these noises. Since it improves the robustness of models greatly, it is an important task for its own sake, not only for reid. To deal with noisy labels, one type of works [9,22] aim to find the relation between the noisy and clean labels, and then give the estimated clean labels for training a robust model. Another type of works [34,25,27] model the data ambiguity through the probability distribution, either the sample posterior or the class prior. Besides noisy labels, these works also handle noisy images. The model in [34] outputs an auxiliary variance for each sample to increase model robustness when handling with noisy labels. [25] evaluates the distance between two images by considering the image-level noise. [34] and [25] both assume each sample to be the probability distribution, a conditional Gaussian posterior, in the latent space specified by the model parameters.

The work in [27] models the prior of the whole data as mixture Gaussians, and it uses the same number of independent Gaussians for each class. By maximizing the data likelihood in the corresponding prior distribution, it proposes a so called Gaussian Mixture (GM) loss, which performs better than other losses particularly when facing the adversarial samples. In GM loss, the extracted feature  $\mathbf{z}$  is assumed to follow a Gaussian mixture distribution as is expressed in Eq. 1.  $\boldsymbol{\mu}^{(k)}$  and  $\boldsymbol{\sigma}^{(k)}$  are the mean and the variance of class  $k$ , and  $p(k)$  is the prior probability of class  $k$ .  $K$  is the total class number. The class probability distribution  $p(y|\mathbf{z})$  can be expressed as Eq. 2.

$$p(\mathbf{z}) = \sum_{k=1}^K p(\mathbf{z}|k)p(k) = \sum_{k=1}^K \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{(k)})p(k) \quad (1)$$

$$p(y|\mathbf{z}) = \frac{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(y)}, \boldsymbol{\sigma}^{(y)})p(y)}{\sum_{k=1}^K \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{(k)})p(k)} \quad (2)$$

The above works show the feasibility for modeling posterior and prior distributions to cope with the data uncertainty. But the relation between them are still not considered.

**Person re-identification.** The most common data types for person reid task are image and video. Image-based works mainly focus on mining local cues to improve the performance. [26] is one of the well-known works that explicitly tear features into parts to enforce the network to mine the discriminative features of different locations to a great extent. [32,12,7] also apply this idea into their works but in diverse ways. [32] concentrates on enlarging activated areas in CAM, while [12,7] feed images and features that have been random erased into a network to get rid of the risk of overfitting. To strengthen local information to tackle the overfitting has become a research trend, but few works pay attention to improve the robustness through modeling uncertainty existing in the data.

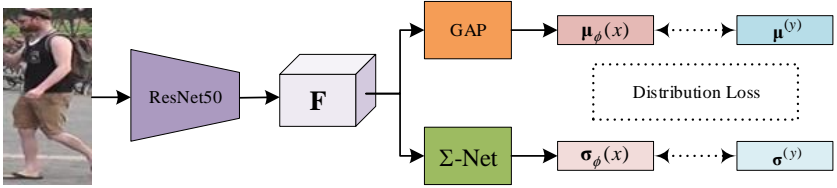
For video-based reid, one of the main issue is how to aggregate features in the temporal domain. [15] proposes to use spatiotemporal attention to aggregate features. [4] proposes a similarity aggregation and co-attentive embedding for video-reid. [37] applies attribute information to disentangle the feature embedding and aggregate the temporal features using attention based on the attribute confidence score. [11] makes a sufficient comparison on the effectiveness of different temporal aggregation methods. It propose to calculate the mean and the variance of the temporal features to model the aggregated feature’s probability density distribution. This is quite similar to our intuition but it does not model the class priors, and not to apply a network to produce the posterior variance.

### 3 Proposed Methods

#### 3.1 Overview Framework

Fig. 1 shows the overview framework of our method. The backbone is ResNet50, which is commonly used in reid task [40]. Its output feature, referred as  $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ , is a 3D tensor depending on the input image  $\mathbf{x}$ , with  $H$ ,  $W$  and  $C$  indicating the height, width and number of channels, respectively. In the upper branch,  $\boldsymbol{\mu}_\phi(\mathbf{x}) \in \mathbb{R}^C$  is directly obtained by the **global average pooling** (GAP), hence it is the mean of  $\mathbf{F}$  in statistics. Note that [40] directly gives the  $\boldsymbol{\mu}_\phi(\mathbf{x})$  into a classifier, expecting it to be discriminative, but high order details in  $\mathbf{F}$  are lost. Our work exploits  $\mathbf{F}$  in the lower branch, by feeding it into a  $\Sigma$ -net to compute the **high order feature**  $\boldsymbol{\sigma}_\phi(\mathbf{x})$ . The idea is to expect  $\boldsymbol{\sigma}_\phi(\mathbf{x})$  to reflect the uncertainty of the latent code  $\mathbf{z}$  on each corresponding dimension. Here we assume that  $\boldsymbol{\mu}_\phi(\mathbf{x})$  and  $\boldsymbol{\sigma}_\phi(\mathbf{x})$  together define the posterior for random variable  $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ , which is of the Gaussian so that  $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$ . Within this setting,  $q_\phi(\mathbf{z}|\mathbf{x})$  reflects the distribution of  $\mathbf{z}$  evaluated by the model  $q_\phi$  based on the input image  $\mathbf{x}$ . Hence the latent space of  $\mathbf{z}$  becomes probabilistic.

To complete the classification, we also define the **class prior** on  $\mathbf{z}$  and assume it to follow the Gaussian form, written as  $p(\mathbf{z}|y) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(y)}, \boldsymbol{\sigma}^{(y)})$ . Here the superscript  $y$  is the category label of input image  $\mathbf{x}$ . Note that all parameters  $\boldsymbol{\mu}^{(y)}$  and  $\boldsymbol{\sigma}^{(y)}$  in  $p(\mathbf{z}|y)$ , and  $\boldsymbol{\mu}_\phi(\mathbf{x})$  and  $\boldsymbol{\sigma}_\phi(\mathbf{x})$  in  $q_\phi(\mathbf{z}|\mathbf{x})$  are of the same dimension, and they are in  $\mathbb{R}^C$ . Different from  $\boldsymbol{\mu}_\phi(\mathbf{x})$  and  $\boldsymbol{\sigma}_\phi(\mathbf{x})$ ,  $\boldsymbol{\mu}^{(y)}$  and  $\boldsymbol{\sigma}^{(y)}$  are **trainable parameters**, rather than the output of a certain layer. Hence they are expected to evaluate the image posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ . Then we design a distribution loss to connect  $q_\phi(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{z}|y)$ , making the posterior from  $\mathbf{x}$  relatively close to the corresponding prior indicated by its class label  $y$ . Details about the  $\Sigma$ -net and the distribution loss are given in following two subsections.



**Fig. 1.** Overview framework of proposed method. It is built upon the backbone of ResNet50. The output  $\mathbf{F}$  is given to two branches. The above one models the mean  $\boldsymbol{\mu}_\phi(\mathbf{x})$  of the Gaussian posterior and the below one employs the  $\Sigma$ -net to output the variance  $\boldsymbol{\sigma}_\phi(\mathbf{x})$  of the same Gaussian. The posterior Gaussian  $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$  is constrained by the prior  $p(\mathbf{z}|y) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(y)}, \boldsymbol{\sigma}^{(y)})$  based on the distribution loss.

### 3.2 $\Sigma$ -Net

We now provide details about the designed structure, named  $\Sigma$ -net, for modeling the posterior variance  $\boldsymbol{\sigma}_\phi(\mathbf{x})$ . The basic idea is to adopt the high order features to make  $\boldsymbol{\sigma}_\phi(\mathbf{x})$  different from the first order feature  $\boldsymbol{\mu}_\phi(\mathbf{x})$ , since the  $\boldsymbol{\sigma}_\phi(\mathbf{x})$  itself is the second order parameter of a distribution. Actually, mining the high order features becomes common in the fine-grained image classification. Many structures seem to fulfill our requirement. However, these works focus on learning discriminative feature, rather than representing the uncertainty. Here we modify the **bilinear pooling structure** in [14,13], and change it to  $\Sigma$ -net so that it becomes appropriate for evaluating the variance of a posterior.

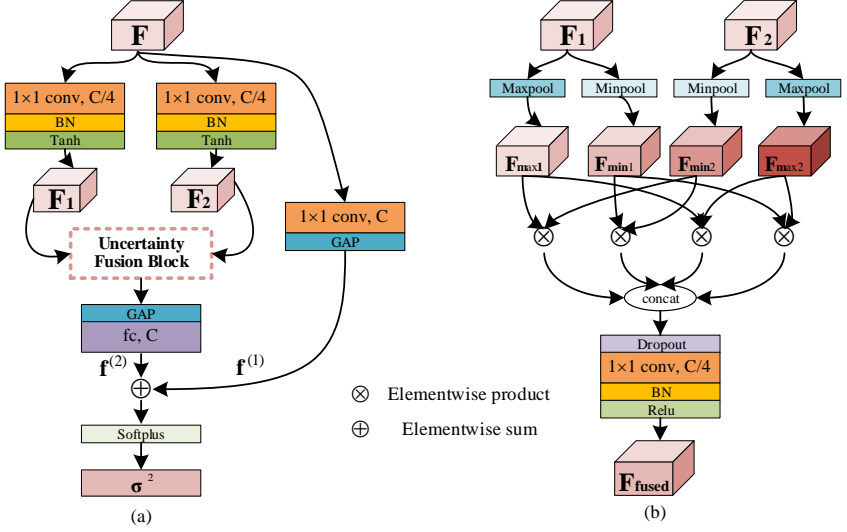
Fig. 2 shows the  $\Sigma$ -net which consists of three main branches. One of them provides the first order residuals  $\mathbf{f}^{(1)} \in \mathbb{R}^C$ , while the other two are multiplied together to form the second order features  $\mathbf{f}^{(2)} \in \mathbb{R}^C$ . Here we use the superscript to indicate feature order in the current module. Specifically, we first use the  $1 \times 1$  conv to construct a **linear shortcut mapping**, and then  $\mathbf{f}^{(1)}$  can be

computed simply by average pooling. To form the second order feature, instead of directly performing the element-wise multiplication between the feature  $\mathbf{F}_1$  and  $\mathbf{F}_2$  like the work in [14], we design an **uncertainty fusion block** which aims to mine the uncertainty in a better way. In this block, the **strided min and max pooling** on  $\mathbf{F}_1$  and  $\mathbf{F}_2$  is first carried out. So that the local min and max values, within the neighbourhood of a spatial location, are extracted. Hence, two sets of feature maps are obtained in each branch. One is the local min indicated by  $\mathbf{F}_{\min 1}$  and  $\mathbf{F}_{\min 2}$ , and the other is the local max  $\mathbf{F}_{\max 1}$  and  $\mathbf{F}_{\max 2}$ . They serve for quantifying the value range at each spatial coordinate. To take advantage of them, we make the **cross multiplication** between the two branches, that is  $\mathbf{F}_{\min 1}$  (or  $\mathbf{F}_{\max 1}$ ) are multiplied by the  $\mathbf{F}_{\min 2}$  (or  $\mathbf{F}_{\max 2}$ ) in another branch. All together, four groups of feature maps, which include  $\mathbf{F}_{\min 1} \otimes \mathbf{F}_{\min 2}$ ,  $\mathbf{F}_{\min 1} \otimes \mathbf{F}_{\max 2}$ ,  $\mathbf{F}_{\max 1} \otimes \mathbf{F}_{\min 2}$ , and  $\mathbf{F}_{\max 1} \otimes \mathbf{F}_{\max 2}$  can be obtained. They are concatenated together followed by the dropout operation, and  $1 \times 1$  conv-BN-ReLU to reduce the channel number to  $C/4$ , the same as its input. Finally  $\Sigma$ -net introduces GAP and another linear function to get the result  $\mathbf{f}^{(2)} \in \mathbb{R}^C$ , hence it is of the same dimension with the number of channels of  $\mathbf{F}$ . Note that the  $\Sigma$ -net adopts the **softplus function** as its activation to make sure that the final second-order feature is positive, and thus suitable for modeling the posterior variance.

### 3.3 Distribution Loss

To build a model reflecting the uncertainty from the data, we assume the network maps its input  $\mathbf{x}$  into a probabilistic embedding defined by the posterior  $q_\phi(\mathbf{z}|\mathbf{x})$ , which is in the Gaussian form  $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$ . Here  $\phi$  indicates the network parameters that generate the mean and variance of the posterior distribution. For the sake of simplicity, we use  $\phi$  to represent the parameters related to the posterior, either the mean or the variance. The label  $y$  for  $\mathbf{x}$  is also available during training, therefore  $q_\phi(\mathbf{z}, y|\mathbf{x})$  is known on training data given  $\phi$ . On the other hand, we also consider the prior  $p(\mathbf{z}|y)$  (or  $p(\mathbf{z}, y)$  given  $y$  is known on the training data) on  $\mathbf{z}$  given a specific label  $y$ . For simplicity,  $p(\mathbf{z}|y) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(y)}, \boldsymbol{\sigma}^{(y)})$  is assumed to be a **single Gaussian for each class**. Although, both  $q_\phi(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{z}|y)$  defines the probabilistic characteristic of  $\mathbf{z}$ . Their difference is still obvious. The former defines on a particular sample  $\mathbf{x}$ , while the latter evaluates whole samples in a class. Usually, the prior  $p(\mathbf{z}|y)$  is predefined and fixed during training. But our work regards  $\boldsymbol{\mu}^{(y)}$  and  $\boldsymbol{\sigma}^{(y)}$  as model parameters indicated by  $\psi$ , hence they are updated together with other parameters  $\phi$  by minimizing the proposed distribution loss. In this case, the prior becomes  $p_\psi(\mathbf{z}|y)$  (or  $p_\psi(\mathbf{z}, y)$ ).

Note that traditional deterministic embedding is a special case in this assumption. If  $\boldsymbol{\sigma}_\phi(\mathbf{x}) \rightarrow 0$ ,  $q_\phi(\mathbf{z}|\mathbf{x})$  becomes a Dirac delta distribution  $\delta(\mathbf{z} - \hat{\mathbf{z}})$ , where  $\hat{\mathbf{z}}$  is the mean of the output feature. This is supposed to happen if the model feels sure about  $\mathbf{x}$ . To derive the proposed loss, we need to connect  $q_\phi(\mathbf{z}|\mathbf{x})$  with  $p_\psi(\mathbf{z}|y)$  indicated by its label  $y$ . We choose the **KL divergence**  $D_{KL}(q_\phi||p_\psi)$  between them as the evaluation metric, which turns out to be the negative log likelihood between the extracted feature  $\hat{\mathbf{z}}$  and the assumed prior  $p_\psi(\mathbf{z}, k)$  when  $\boldsymbol{\sigma}_\phi(\mathbf{x}) \rightarrow 0$ . The relation between the likelihood  $\mathcal{N}(\hat{\mathbf{z}}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{(k)})$  and the



**Fig. 2.** A schematic of the proposed  $\Sigma$ -net. (a) shows the structure of the whole  $\Sigma$ -net. It outputs the variance of the posterior based on the second order feature  $\mathbf{f}^{(2)}$  and the first order residual  $\mathbf{f}^{(1)}$ . The uncertainty fusion block replaces the direct element-wise production in the bilinear pooling layer. (b) illustrates the details of the uncertainty fusion block. It has two inputs  $\mathbf{F}_1$  and  $\mathbf{F}_2$ , and gives the output  $\mathbf{F}_{fused}$  with the same size as the two inputs.

$D_{KL}(q_\phi\|p_\psi)$  when  $\sigma_\phi(\mathbf{x}) \rightarrow 0$  is summarized in Eq. 3. The derivation can be found in the Appendix. Note that here  $k$  can specify any prior, not necessarily being the same as  $y$ .

$$\mathbb{E}_{q_\phi(\mathbf{z}, k|\mathbf{x})} \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{(k)}) p(k) = \exp(-D_{KL}(q_\phi(\mathbf{z}, k|\mathbf{x})\|p_\psi(\mathbf{z}, k)) + \text{Const.}) \quad (3)$$

Combining Eq. 3. with Eq. 2. from GM loss [27] and ignoring the Const., the conditional probability distribution  $p(y|\mathbf{z})$  under our assumption becomes Eq. 4. Moreover, the proposed distribution loss can be easily defined by the cross entropy between the  $p(y|\mathbf{z})$  and the one-hot label, as is expressed in Eq. 5.

$$p(y|\mathbf{z}) = \frac{\exp(-D_{KL}(q_\phi(\mathbf{z}, y|\mathbf{x})\|p_\psi(\mathbf{z}, y)))}{\sum_{k=1}^K \exp(-D_{KL}(q_\phi(\mathbf{z}, k|\mathbf{x})\|p_\psi(\mathbf{z}, k)))} \quad (4)$$

$$\begin{aligned} \mathcal{L}_{cls} &= -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \mathbb{I}(k == y) \log p(k|\mathbf{z}) \\ &= -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(-D_{KL}(q_\phi(\mathbf{z}, y|\mathbf{x}_i)\|p_\psi(\mathbf{z}, y)))}{\sum_{k=1}^K \exp(-D_{KL}(q_\phi(\mathbf{z}, k|\mathbf{x}_i)\|p_\psi(\mathbf{z}, k)))} \end{aligned} \quad (5)$$

$\mathcal{L}_{cls}$  only ensures the posterior is relatively closer to the correct prior than the other priors. Similar with GM loss, a **KL divergence regularization** can be employed to measure to what extent the posterior fits the assumed prior, as is shown in Eq. 6.

$$\mathcal{L}_{KL} = D_{KL}(q_\phi(\mathbf{z}, y|\mathbf{x})||p_\psi(\mathbf{z}, y)) \quad (6)$$

In summary, the distribution loss  $L_{dist}$  can be defined in Eq. 7, where  $\lambda$  is a non-negative hyper parameter. Note that, there is a closed form of KL divergence between two Gaussians, which can be found in the Appendix.

$$\mathcal{L}_{dist} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{KL} \quad (7)$$

### 3.4 Prior Guided Soft Labels

In traditional supervised classification, the label is usually represented by a one-hot vector with the number of entries equal to the total number of the classes. Each element is either 1 or 0, with the former lying at the ground truth classes. However, this setting ignores the fact that **the distances between different classes are not the same**, while these distances can help the classifier to understand the task. Now the key issue is how to model and compute the distance between any two classes. In practice, since it is difficult to evaluate the distance between two classes, the trick of **label smoothing** is widely used to prevent the classifier from over-fitting. This trick provides the soft label by cutting a small predefined value from the ground truth, and randomly assigning to other elements in the label vector. Indeed, it shows its advantages comparing with the one-hot label, but the class distances are still not considered.

Our algorithm tries to **build the Gaussian priors for each classes in the latent embedding**, which implies that the priors naturally lie in the latent space after training, and the distances among them reflect the class similarities. We argue that these Gaussians can be further exploited, and the distance between two of them can serve to form the soft label. Particularly, we employ the Wasserstein distance  $D_w$  to evaluate similarity between two class priors  $p_{\psi_1}(\mathbf{z}|y_1)$  and  $p_{\psi_2}(\mathbf{z}|y_2)$ . Note that  $D_w$  has the closed form solution given two Gaussians, and it can be computed as Eq. 8

$$D_w(p_{\psi_1}(\mathbf{z}|y_1); p_{\psi_2}(\mathbf{z}|y_2)) = \|\boldsymbol{\mu}^{(y_1)} - \boldsymbol{\mu}^{(y_2)}\|_2^2 + \|\sqrt{\boldsymbol{\sigma}^{(y_1)}} - \sqrt{\boldsymbol{\sigma}^{(y_2)}}\|_F^2 \quad (8)$$

Once we have the similarity matrix, we normalize each row with Eq. 9, hence the soft label depending on the learned prior can be generated.  $\tau$  is a temperature hyper-parameter, ranging from 0 to 1, for modulating the smoothness of the generated soft label. A higher hyper-parameter brings a softer label. In our experiments, we **fix the  $\tau$  as 0.17**, with which we can acquire a soft label with its maximum value close to 0.9.

$$\mathbf{y}_{soft} = \frac{\exp(-D_w/\tau)}{\sum_{k=1}^K \exp(-D_w/\tau)} \quad (9)$$



Note that, the soft label is not suitable for an end-to-end training manner as it is unstable due to the rapidly developing priors. We apply the soft label with a **two-stage training method**. At the first stage, label smoothing is utilized to ensure the model generalization and the training stability. At the second stage, the flexible soft label takes the place of the fixed smoothed label to help finetune the whole model. The evaluation results can be found in section 4.3.

## 4 Experiments

### 4.1 Datasets and Evaluation metrics

Both the image-based datasets and a video-based dataset are adopted for evaluation. The chosen image-based datasets include Market-1501 [39] and DukeMTMC-reID [24]. The selected video-based dataset is MARS [38]. The Cumulative Matching Characteristic (CMC) [10] and mean Average Precision (mAP) [39] are the performance metrics.

**Market-1501** contains 1501 identities captured by 6 cameras from different viewpoints. 12936 images of 751 identities are for training, 3368 query images and 19732 gallery images of the other 750 identities compose the test set. **DukeMTMC-reID** consists of 36411 images of 1404 identities, half of the identities are used for training, the 2228 query images and 17661 gallery images of the remained identities are applied for testing. **MARS** is a common dataset used for video-based person re-identification. It consists of 1261 different pedestrians and 20715 tracklets, and each of the identities is captured by at least 2 cameras. We follow the training/evaluation protocol used in [38], which selects 625 identities for training and the remaining for testing.

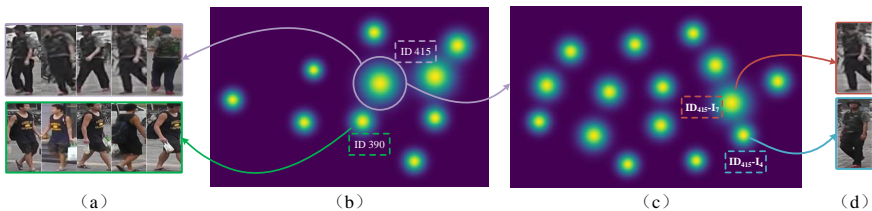
### 4.2 Implementation Details

We extend our experiments on the *reid strong baseline* framework [21]. The used tricks include *warmup*, *random erasing augmentaion*, *label smoothing* and *last stride=1*. Note that, for image-based datasets, we train our model only with the distribution loss, triplet loss is not added, so as to make fair comparison with [34]. While for video-based dataset, we add the triplet loss to match the related works' experiment settings. The total training epoch for MARS is 400 epochs. The base learning rate is  $3.5e-4$ , and it will decay by 3 at 70, 140, 210, 310 epoch. Besides, the number of sequences per batch is 16 and the sequence length is 4.

### 4.3 Visualization of the Probabilistic Embedding and the Analysis

In order to better understand the high dimensional space of the sample posterior and the class prior, we are interested in visualizing them, as is shown in Fig. 3. Since both the posterior and the prior are represented by high dimensional mean and variance vectors, the visualization is carried out based on the sampling and dimensionality reduction. The pipeline is listed as below.

- Sample 2000 codes from each distribution.
- Use TSNE to project the high-dimensional codes into a 2D plane.
- Re-compute the mean and variance of the projected points from a certain distribution.
- Draw a 2D Gaussian distribution with the re-computed mean and variance.



**Fig. 3.** Visualization of the sample posterior and the class prior distribution. (a) The random selected images of ID-415 and ID-390 in Market1501 dataset. (b) The visualization of the prior from 10 randomly selected IDs. (c) The visualization of 15 posteriors in ID-415. (d) Two corresponding raw images of the selected posterior in (c). Radius of the visualized Gaussian distributions is proportional to the corresponding variance, thus larger radius indicates more uncertainty lying in the class or image. This figure is best viewed in color.

In our setting, we model the variance of the prior and the posterior to indicate the uncertainty of the class and the image, respectively. To clearly illustrate the effectiveness of the learned variances of the priors, we randomly choose 10 IDs in Market1501 to visualize the priors for them, which are ID 136, 390, 415, 442, 589, 792, 814, 982, 1007, 1260. Their corresponding priors are shown in Fig. 3 (b). It is obvious that some IDs (*e.g.* ID-415) **covering a larger area** than the others (*e.g.* ID-390), which implies that it has the larger uncertainty. Note that images from ID-390 are clear and the visual cues, such as the logo on the clothes and the bags in hand and on back, are evident, while images from ID-415 are blurry without many cues, hence it is more ambiguous for identification.

Besides the priors, we are also interested in visualizing the posterior, therefore, we use the same pipelines for visualization on the posteriors from ID-415. The Gaussians for 15 images are shown in Fig.3 (c). Note that these Gaussians are from images of the same ID-415. But still, we find similar phenomenon as it is on the priors. The covering areas are quite different. Fig.3 (d) shows two raw images ( $I_7$  and  $I_4$ ) of the corresponding Gaussians in Fig.3 (c). The above image  $I_7$  has the larger variance than the below one  $I_4$ , and its raw image also has relatively low image quality. It can be concluded that, the learned variances can indicate the uncertainty of a class or an image, and the uncertainty usually depends on the image quality, apparent attributes, etc., which fits well with our common sense.

#### 4.4 Quantitative Analysis and Ablation Study

The following experiments can be divided into three aspects. Firstly, we evaluate our propositions on common datasets without noises to see their performance on clean data. Secondly, evaluation on noisy data, consisting of label noise and image noise, is needed to validate the robustness of our model. And finally, we make an ablation study on the designed  $\Sigma$ -Net.

**Experiments on clean data** We separately conduct experiments on the baseline model, the class priors and the sample posterior variance, so as to intuitively demonstrate the effectiveness of our proposition. In the following subsections, **baseline** represents for the classic IDE [40] model constrained by cross entropy; **+ prior** means the class priors are added to the baseline and the corresponding loss function is GM loss [27]; **+variance** is our full model, as is shown in Fig. 1. **+ soft label** is considered to be a trick, which can be applied to any model that has defined the class priors. Here, we add the soft label trick to finetune our full model. **+ reranking** is a widely used trick for person reid, we only add it to the full model finetuned with soft label to see the best performance.

**Table 1.** The performance of different models evaluated on Market1501 and DukeMTMC-reID datasets

Methods		Market1501		DukeMTMC-reID	
		Rank-1 mAP		Rank-1	mAP
$\mathbf{G}$	AOS(CVPR18)[12]	86.5	79.4	79.2	62.1
	DistNet(ICC2019)[34]	87.3	70.8	74.7	56.0
	MLFN(CVPR18)[2]	90.0	74.3	81.0	62.8
	BFE(ICC2019)[7]	<b>95.3</b>	<b>86.7</b>	<b>89.0</b>	<b>76.0</b>
$\mathbf{L}(+\mathbf{G})$	HA-CNN(CVPR18)[17]	91.2	75.7	80.5	63.8
	PCB(ECCV2018)[26]	93.8	81.6	83.3	69.2
	CAMA(CVPR19)[32]	94.7	84.5	85.8	72.9
baseline		89.7	78.4	80.0	65.3
+prior		89.8	78.7	80.6	68.4
+variance		91.0	80.0	82.0	68.3
+soft label		<b>91.2</b>	<b>80.5</b>	<b>82.6</b>	<b>68.8</b>
+reranking		92.4	89.8	87.3	83.7

Our results on both image-based datasets and video-based dataset are listed in Table 1 and Table 2, respectively. For image-based reid, we divide the previous work into  $\mathbf{G}$  and  $\mathbf{L}(+\mathbf{G})$  group.  $\mathbf{G}$  indicates the work only utilizes the global features, while  $\mathbf{L}(+\mathbf{G})$  means local features are also employed during training. In our setting, we only use the global features without focusing on the local region. As is shown in Table 1, compared with the most of the  $\mathbf{G}$  methods, our approach has better results, except for BFE [7], which actually applies the drop blocks to enhance the local regions, hence improves the performance of the

**Table 2.** The performance of different models evaluated on MARS dataset

Method	Rank-1	mAP
DRSA[15]	82.3	65.8
Snippet[4]	86.3	76.1
ADTA[37]	<b>87.7</b>	78.2
mean[11]	82.9	76.2
mean + variance[11]	85.2	77.9
+ prior	85.4	79.3
+ $\Sigma$ -Net	85.4	<b>79.6</b>

model. In addition, through separating each proposed modules, it is clearly that each module makes contribution for improving the performance.

For the video-based work, [11] has proved that modeling sequence feature as a probability density distribution can reach competitive results with other temporal aggregation methods. We then add class priors learning on the GE model [11] by introducing GM loss. As is shown in Table 2, the participation of the class priors brings about an improvement. Furthermore, we exploit the variance of the sample posterior by the proposed  $\Sigma$ -Net, rather than directly utilize the variance in statistics like **mean + variance**. The result is listed  **$\Sigma$ -Net**, the proposed  $\Sigma$ -Net brings 0.3% increase in mAP. Before applying the  $\Sigma$ -Net, we need to first max pool the feature on the temporal space, so that we can get the aggregated feature to produce a variance that is correspond to the input sequence.

**Table 3.** The performance of different models on Market1501 with 10% random noise on label

Method	Rank-1	mAP
DistNet[34]	82.1	62.0
baseline	79.4	58.9
+prior	81.8	63.3
+variance	<b>83.1</b>	<b>65.8</b>

**Experiments on noisy data** Data uncertainty usually reflects on two aspects, "noises on image" or "noises on label". DistNet [34] has conducted sufficient experiment on modeling a posterior distribution to deal with noisy data. We follow the testing protocol proposed in [34] to evaluate our approach on Market1501 with 10% random noise. To be fair, we re-implement the DistNet net on this dataset, and the result listed in Table 3 is slightly higher than the results claimed in [34]. When the class priors are added to the baseline, the mAP has already exceed DistNet 1.3%. And then by adding the posterior variance, our approach

can still have 2% increase in Rank-1 and mAP. The results once again prove the effectiveness of our model.

To evaluate the robustness against the image-level noise, we design a new testing protocol. As person reid is a quite easy to overfit, at the training stage, most works prefer to apply random crop, erasing, flip, etc. to strengthen the generalization of their models. Therefore, at the test stage, the noise from the augmentation will not be a challenge for the models. To avoid the influence from the data augmentation, we choose to add Gaussian blur with different kernel size onto the raw query images. And then the blurred query images will be used for retrieving among the clean gallery images. We have evaluated our model on four different degrees of blur, and the evaluation results are listed in Table 4. Note that, all the models to be evaluated are trained on the clean data. It can be seen that the learned class priors make few contribution against the blurry attack, and the sample posterior plays a key role for strengthening model robustness. DistNet also models the sample posterior, but poor performance may be due to a certain amount of information loss caused by the sampling operation.

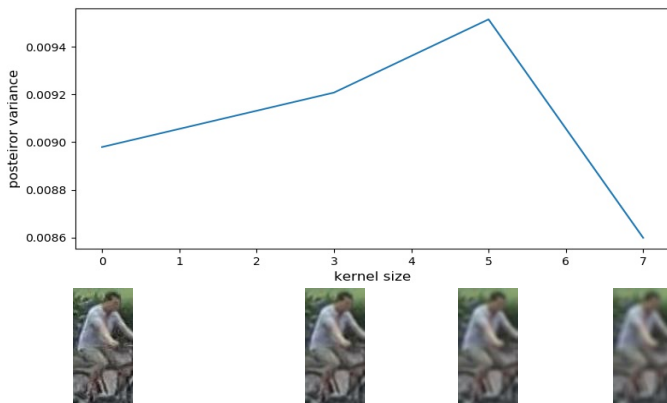
**Table 4.** The performance of different models on Market1501 with four Gaussian blur of different kernel size

Method	$0 \times 0$		$3 \times 3$		$5 \times 5$		$7 \times 7$	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
DistNet [34]	87.6	72.8	82.4	66.9	62.4	48.1	35.7	27.8
baseline	89.7	78.4	85.7	73.8	62.7	50.6	38.3	30.3
+ prior	89.8	78.7	84.4	72.5	59.8	48.7	29.7	24.6
+ variance	<b>91.0</b>	<b>80.0</b>	<b>86.8</b>	<b>74.9</b>	<b>69.6</b>	<b>56.8</b>	<b>43.6</b>	<b>35.1</b>

In Fig. 4, we list four blurry examples and the mean value of their corresponding posterior variance on different dimensions. As the degree of ambiguity deepens, the output posterior variance will gradually increase. But after a certain level, it will drop rapidly. We consider it is because the sever ambiguity has successfully attack the model, and the posterior distribution starts to be away from the previous center.

**Table 5.** The ablation study on sigma net. The performance is evaluated on Market1501 dataset

Method	clean data		noisy label		noisy image (3)		noisy image (5)		noisy image (7)	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
MLP	90.5	80.0	82.2	61.8	84.5	70.5	62.4	49.1	35.8	27.9
BM	<b>90.8</b>	<b>80.4</b>	<b>84.2</b>	63.2	85.0	71.2	63.5	50.8	37.4	29.9
$\Sigma$ -Net	91.0	80.0	83.1	<b>65.8</b>	<b>86.8</b>	<b>74.9</b>	<b>69.6</b>	<b>56.8</b>	<b>43.6</b>	<b>35.1</b>



**Fig. 4.** Examples of blurred query images and the mean value of their corresponding posterior variances. From left to the right, the blurred kernel size is  $0 \times 0$  (*i.e.* raw images),  $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$ . Best viewed in color.

**Ablation study on  $\Sigma$ -Net.** The proposed  $\Sigma$ -Net has gone through three development period. The first one is an **MLP**, which only generates the first-order feature. Then we consider to use the bilinear model (**BM**), which refers to the  $\Sigma$ -Net without the uncertainty fusion block, to introduce the second-order feature. And in the last period, we add the **Uncertainty Fusion Block** to better grasp the feature uncertainty. Table 4.4 lists the comparison results on the three structures. When evaluated on clean data, the three structures have the similar performance, and **BM** seems to be the best. However, when we apply them to noisy data,  $\Sigma$ -Net has the robustest characteristic.

## 5 Conclusion

In this paper, we propose to learn the posterior and prior simultaneously with a novel distribution loss, which builds the connection between the posterior and the prior and makes the whole training process able to follow the end-to-end training trend. Furthermore, we propose a  $\Sigma$ -Net module to output a second-order feature as the posterior variance to maintain the mathematical significance of the variance, which is an omitted point of the previous work. In addition, we visualize the learned posteriors and class priors, illustrating that the learned distributions can reflect the data characteristics to some extent. We also list the results of comparison with other works and ours ablation studies. The experiment results meet our expectation and prove the effectiveness of our approach.

## A Mathematical proofs

We would like to set up the relation between the proposed distribution loss and the GM loss [27]. It is easy to prove that the KL divergence, between the posterior  $q_\phi(\mathbf{z}|\mathbf{x})$  and the prior  $p(\mathbf{z}|y)$ , degenerates to the log-likelihood of the prior  $p(\mathbf{z}|y)$ , when the posterior variance  $\sigma_\phi(\mathbf{x}) \rightarrow 0$ . On the other hand, the we assume both  $q_\phi(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{z}|y)$  are Gaussians, therefore, the KL divergence can be computed in the closed-form solution. These details are given in A.1 and A.2, respectively.

### A.1 Relation between KL divergence and log-likelihood

Assume that the posterior  $q_\phi(\mathbf{z}|\mathbf{x})$  is a Dirac delta distribution.

$$q_\phi(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - \hat{\mathbf{z}})$$

If we assume conditional independence between  $\mathbf{z}$  and the class  $y$ , then we have

$$q_\phi(\mathbf{z}, y|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x})p(y|\mathbf{x})$$

$p(y|\mathbf{x})$  is the one-hot encoding of label.

$$\begin{aligned} D_{KL}(q_\phi(\mathbf{z}, y|\mathbf{x})||p(\mathbf{z}, y)) &= D_{KL}[\delta(\mathbf{z} - \hat{\mathbf{z}})p(y|\mathbf{x})||p(\mathbf{z}|y)p(y)] \\ &= -\sum_k \int \delta(\mathbf{z} - \hat{\mathbf{z}})p(k|\mathbf{x}) \log \frac{p(\mathbf{z}|k)p(k)}{\delta(\mathbf{z} - \hat{\mathbf{z}})p(k|\mathbf{x})} d\mathbf{z} \\ &= -\sum_k \int \delta(\mathbf{z} - \hat{\mathbf{z}})p(k|\mathbf{x}) \log p(\mathbf{z}|k) d\mathbf{z} - \sum_k \int \delta(\mathbf{z} - \hat{\mathbf{z}})p(k|\mathbf{x}) \log \frac{p(k)}{p(k|\mathbf{x})} d\mathbf{z} \\ &\quad + \sum_k \int \delta(\mathbf{z} - \hat{\mathbf{z}})p(k|\mathbf{x}) \log \delta(\mathbf{z} - \hat{\mathbf{z}}) d\mathbf{z} \\ &= -\sum_k p(k|\mathbf{x}) \log p(\hat{\mathbf{z}}|k) - \sum_k p(k|\mathbf{x}) \log \frac{p(k)}{p(k|\mathbf{x})} + \int \delta(\mathbf{z} - \hat{\mathbf{z}}) \log \delta(\mathbf{z} - \hat{\mathbf{z}}) d\mathbf{z} \end{aligned}$$

The last two terms are constants, therefore,

$$\begin{aligned} D_{KL}(q_\phi(\mathbf{z}, y|\mathbf{x})||p(\mathbf{z}, y)) &= -\sum_k p(k|\mathbf{x}) \log p(\hat{\mathbf{z}}|k) + \text{Const.} \\ &= -\sum_k \mathbb{I}(k == y) \log p(\hat{\mathbf{z}}|k) + \text{Const.} \\ &= -\sum_k \mathbb{I}(k == y) \log \mathcal{N}(\hat{\mathbf{z}}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{(k)}) + \text{Const.} \\ &= -\log \mathcal{N}(\hat{\mathbf{z}}; \boldsymbol{\mu}^{(y)}, \boldsymbol{\sigma}^{(y)}) + \text{Const.} \end{aligned}$$

Hence,

$$\mathcal{N}(\hat{\mathbf{z}}; \boldsymbol{\mu}^{(y)}, \boldsymbol{\sigma}^{(y)}) = \exp(-D_{KL}(q_\phi(\mathbf{z}, y|\mathbf{x})\|p(\mathbf{z}, y) + \text{Const.}))$$

Note that, this equation is equivalent to Eq. 3 in the main body, as the expectation in Eq. 3 can be omitted due to  $p(k)$  is a one-hot label.

## A.2 KL divergence of two multivariate Gaussians

$$D_{KL}(f\|g) = \int f(x) \ln \frac{f(x)}{g(x)} dx$$

For two Gaussians  $f$  and  $g$  defined as the  $d$  dimensional probability density functions, the KL divergence has a closed formed expression, where  $\boldsymbol{\mu}_g$ ,  $\boldsymbol{\Sigma}_g$  and  $\boldsymbol{\mu}_f$ ,  $\boldsymbol{\Sigma}_f$  are the mean vectors and co-variance matrices for  $g$  and  $f$ , respectively.

$$D(f\|g) = \frac{1}{2} \left[ \ln \frac{|\boldsymbol{\Sigma}_g|}{|\boldsymbol{\Sigma}_f|} + \text{Tr}(\boldsymbol{\Sigma}_g^{-1} \boldsymbol{\Sigma}_f) - d + (\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)^\top \boldsymbol{\Sigma}_g^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_g) \right]$$

Hence, in our setting,

$$\begin{aligned} & D(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|y)) \\ &= \frac{1}{2} \left[ \ln \frac{|\boldsymbol{\sigma}^{(y)}|}{|\boldsymbol{\sigma}_\phi(\mathbf{x})|} + \text{Tr}(\frac{\boldsymbol{\sigma}_\phi(\mathbf{x})}{\boldsymbol{\sigma}^{(y)}}) + \frac{(\boldsymbol{\mu}_\phi(\mathbf{x}) - \boldsymbol{\mu}^{(y)})^\top (\boldsymbol{\mu}_\phi(\mathbf{x}) - \boldsymbol{\mu}^{(y)})}{\boldsymbol{\sigma}^{(y)}} \right] + \text{Const.} \end{aligned}$$

Note that  $\boldsymbol{\mu}_\phi(\mathbf{x})$ ,  $\boldsymbol{\sigma}_\phi(\mathbf{x})$ ,  $\boldsymbol{\mu}^{(y)}$  and  $\boldsymbol{\sigma}^{(y)}$  are diagonal matrices.

## B Supplementary Experiments

To further validate our propositions' robustness against image-level noise, we conduct the following experiments. We manually make up three different types of noises, which are the simulations of motion blurs, low resolutions and occlusions, to degrade the image quality of the query set in the Market1501 [39]. The degraded images are shown in Fig. 6 and Fig. 7, respectively. Note that to demonstrate the robustness of model for unseen noises, the degraded images do not incorporate into the training set by the data augmentations.

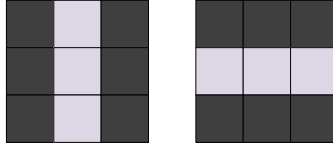
**Table 6.** The performances of different models against the motion blur on Market1501.

Method	0 × 0		5 × 5		10 × 10		15 × 15	
	Rank-1 mAP		Rank-1 mAP		Rank-1 mAP		Rank-1 mAP	
DistNet [34]	87.6	72.8	76.8	60.3	46.3	35.5	21.1	17.2
baseline	89.7	78.4	79.7	67.2	49.6	40.7	20.0	19.4
+ prior	89.8	78.7	79.6	67.1	46.2	37.8	19.6	16.9
+ variance	<b>91.0</b>	<b>80.0</b>	<b>83.0</b>	<b>70.7</b>	<b>52.8</b>	<b>43.4</b>	<b>23.6</b>	<b>21.1</b>



**Table 7.** The performance of different models against the interpolation noise on Market1501.

Method	1.0		0.75		0.50		0.25	
	Rank-1 mAP		Rank-1 mAP		Rank-1 mAP		Rank-1 mAP	
DistNet [34]	87.6	72.8	83.2	67.1	78.4	61.9	56.8	43.1
baseline	89.7	78.4	87.9	75.7	83.0	69.2	56.9	45.8
+ prior	89.8	78.7	87.1	75.9	81.9	69.7	54.1	43.7
+ variance	<b>91.0</b>	<b>80.0</b>	<b>88.7</b>	<b>77.8</b>	<b>85.2</b>	<b>72.8</b>	<b>62.9</b>	<b>51.5</b>

**Fig. 5.** Examples of  $3 \times 3$  motion blur kernel. The left kernel is for generating vertical motion blur, and the right is for horizontal motion blur. The value of the black element is 0, and the value of the purple one is  $1/\text{kernel size}$ .

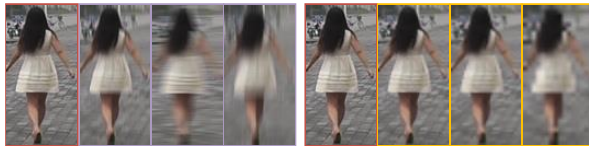
Motion blur often happens in reid due to the targets' fast moving. In Table 6, the kernel sizes of  $5 \times 5$ ,  $10 \times 10$ , and  $15 \times 15$  indicate the intensity of the blurring. The kernel are essentially of two types, as is shown in Fig. 5, to simulate the blur caused by the vertical and horizontal motions, respectively. The probability of applying them are 50% and 50%.

The second type of degradation is caused by the limited resolution of target. To simulate this case, we first downsize the original image, then apply the bi-linear interpolation to recover it. The test results are listed in Table 7, **0.75**, **0.50**, **0.25** represent for the different downsize ratios.

It can be seen from Table 6 and Table 7, our full model achieves the best performances in both cases. Here **baseline**, **+prior** and **+variance** are the three ablation models, which are introduced in experimental section of the paper. **DistNet** is the model from [34].

**Table 8.** The performance of different models against the random erasing on Market1501.

Method	0.		0.1		0.2		0.3		0.4	
	Rank-1 mAP		Rank-1 mAP		Rank-1 mAP		Rank-1 mAP		Rank-1 mAP	
DistNet [34]	83.3	65.9	41.5	32.8	20.0	17.0	9.2	8.7	3.8	3.8
baseline	86.7	72.4	47.6	39.1	25.9	21.0	9.8	9.6	3.6	4.0
+ prior	<b>88.9</b>	<b>74.9</b>	<b>58.7</b>	<b>47.6</b>	30.6	26.5	12.2	12.0	4.1	4.8
+ variance	87.9	73.8	55.4	45.1	<b>31.2</b>	<b>26.8</b>	<b>14.9</b>	<b>13.9</b>	<b>5.7</b>	<b>6.4</b>



**Fig. 6.** Degraded images from the motion blur (the left four) and the interpolation noise (the right four). Red boxes are the raw images. Purple boxes, from left to the right, correspond to noise levels with  $5 \times 5$ ,  $10 \times 10$ ,  $15 \times 15$  kernel size. Similarly, images in yellow boxes, from left to right, are the decayed results of the bi-linear resizing from the downsize at 0.75, 0.5 and 0.25 of the original resolution. Best viewed in color.



**Fig. 7.** Samples of randomly erased images with different area sizes. From the left to the right, there are four groups of images. Group red contains images with 10% of the randomly erased area. Group orange, green and blue are for 20%, 30% and 40%, respectively.

The third type of noise we are considering is the occlusion. And most classic works prefer to use random erasing as an augmentation in the training phase. To fairly evaluate the robustness against the occlusion, we retrain all the four compared models without the random erasing augmentation, and apply it only on the query set during the evaluation. As is shown in Fig. 7, we control the erasing area as the noise level.

Table 8 lists the comparison results of random erasing test. When the occluded area is small, the  $+$  **prior** model shows the best performance. While as it becomes larger, our full model shows its potential against the occlusion attack.

## References

1. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: European Conference on Computer Vision. pp. 430–443. Springer (2012)

2. Chang, X., Hospedales, T.M., Xiang, T.: Multi-level factorisation net for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2109–2118 (2018)
3. Chen, B., Deng, W., Hu, J.: Mixed high-order attention network for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 371–381 (2019)
4. Chen, D., Li, H., Xiao, T., Yi, S., Wang, X.: Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1169–1178 (2018)
5. Chen, T., Ding, S., Xie, J., Yuan, Y., Chen, W., Yang, Y., Ren, Z., Wang, Z.: Abdnnet: Attentive but diverse person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8351–8361 (2019)
6. Cho, Y.J., Yoon, K.J.: Improving person re-identification via pose-aware multi-shot matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1354–1362 (2016)
7. Dai, Z., Chen, M., Gu, X., Zhu, S., Tan, P.: Batch dropblock network for person re-identification and beyond. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3691–3701 (2019)
8. Das, A., Chakraborty, A., Roy-Chowdhury, A.K.: Consistent re-identification in a camera network. In: European conference on computer vision. pp. 330–345. Springer (2014)
9. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer (2016)
10. Gray, D., Brennan, S., Tao, H.: Evaluating appearance models for recognition, reacquisition, and tracking. In: Proc. IEEE international workshop on performance evaluation for tracking and surveillance (PETS). vol. 3, pp. 1–7. Citeseer (2007)
11. Hu, T.Y., Hauptmann, A.G.: Multi-shot person re-identification through set distance with visual distributional representation. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 262–270 (2019)
12. Huang, H., Li, D., Zhang, Z., Chen, X., Huang, K.: Adversarially occluded samples for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5098–5107 (2018)
13. Jacob, P., Picard, D., Histace, A., Klein, E.: Metric learning with horde: High-order regularizer for deep embeddings. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6539–6548 (2019)
14. Kim, J.H., On, K.W., Lim, W., Kim, J., Ha, J.W., Zhang, B.T.: Hadamard product for low-rank bilinear pooling. arXiv preprint arXiv:1610.04325 (2016)
15. Li, S., Bak, S., Carr, P., Wang, X.: Diversity regularized spatiotemporal attention for video-based person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 369–378 (2018)
16. Li, S., Xiao, T., Li, H., Yang, W., Wang, X.: Identity-aware textual-visual matching with latent co-attention. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1890–1899 (2017)
17. Li, W., Zhu, X., Gong, S.: Harmonious attention network for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2285–2294 (2018)
18. Li, X., Zheng, W.S., Wang, X., Xiang, T., Gong, S.: Multi-scale learning for low-resolution person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3765–3773 (2015)

19. Li, Z., Chang, S., Liang, F., Huang, T.S., Cao, L., Smith, J.R.: Learning locally-adaptive decision functions for person verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3610–3617 (2013)
20. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1449–1457 (2015)
21. Luo, H., Gu, Y., Liao, X., Lai, S., Jiang, W.: Bag of tricks and a strong baseline for deep person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
22. Northcutt, C.G., Jiang, L., Chuang, I.L.: Confident learning: Estimating uncertainty in dataset labels. arXiv preprint arXiv:1911.00068 (2019)
23. Quan, R., Dong, X., Wu, Y., Zhu, L., Yang, Y.: Auto-reid: Searching for a part-aware convnet for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3750–3759 (2019)
24. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision. pp. 17–35. Springer (2016)
25. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6902–6911 (2019)
26. Sun, Y., Zheng, L., Yang, Y., Tian, Q., Wang, S.: Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 480–496 (2018)
27. Wan, W., Zhong, Y., Li, T., Chen, J.: Rethinking feature distribution for loss functions in image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9117–9126 (2018)
28. Wang, T., Gong, S., Zhu, X., Wang, S.: Person re-identification by video ranking. In: European conference on computer vision. pp. 688–703. Springer (2014)
29. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
30. Wang, Y., Wang, L., You, Y., Zou, X., Chen, V., Li, S., Huang, G., Hariharan, B., Weinberger, K.Q.: Resource aware person re-identification across multiple resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8042–8051 (2018)
31. Xia, B.N., Gong, Y., Zhang, Y., Poellabauer, C.: Second-order non-local attention networks for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3760–3769 (2019)
32. Yang, W., Huang, H., Zhang, Z., Chen, X., Huang, K., Zhang, S.: Towards rich feature discovery with class activation maps augmentation for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1389–1398 (2019)
33. Ye, M., Liang, C., Wang, Z., Leng, Q., Chen, J., Liu, J.: Specific person retrieval via incomplete text description. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 547–550 (2015)
34. Yu, T., Li, D., Yang, Y., Hospedales, T.M., Xiang, T.: Robust person re-identification by modelling feature uncertainty. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 552–561 (2019)
35. Zhao, H., Tian, M., Sun, S., Shao, J., Yan, J., Yi, S., Wang, X., Tang, X.: Spindle net: Person re-identification with human body region guided feature decomposition

- and fusion. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1077–1085 (2017)
36. Zhao, R., Ouyang, W., Wang, X.: Unsupervised salience learning for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3586–3593 (2013)
  37. Zhao, Y., Shen, X., Jin, Z., Lu, H., Hua, X.s.: Attribute-driven feature disentangling and temporal aggregation for video person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4913–4922 (2019)
  38. Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., Tian, Q.: Mars: A video benchmark for large-scale person re-identification. In: European Conference on Computer Vision. pp. 868–884. Springer (2016)
  39. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: Proceedings of the IEEE international conference on computer vision. pp. 1116–1124 (2015)
  40. Zheng, L., Yang, Y., Hauptmann, A.G.: Person re-identification: Past, present and future. arXiv preprint arXiv:1610.02984 (2016)
  41. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.S.: Hierarchical gaussianization for image classification. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 1971–1977. IEEE (2009)