Towards Learning Multi-domain Crowd Counting

Zhaoyi Yan, Pengyu Li, Biao Wang, Dongwei Ren, *Member, IEEE* Wangmeng Zuo, *Senior Member, IEEE*

***Abstract*—Recently, deep learning-based crowd counting methods have achieved promising performance on test data with the same distribution as training set, while performance degradation usually occurs when testing on other or unseen domains. Due to the variations in scene contexts, crowd densities and head scales, it is a very challenging issue to tackle multi-domain crowd counting using one deep model. In this work, we propose a domain-guided channel attention network (DCANet) towards learning multi-domain crowd counting. In particular, our DCANet consists of feature extraction module, channel attention-guided multi-dilation (CAMD) module and density map prediction module. Given a testing image from a certain domain, channel attention is adopted to guide the extraction of domain-specific feature representation, and thus our DCANet can adaptively handle images from multiple domains. We further propose two domain-guided learning strategies, *i.e.*, dataset-level domain kernel (DDK) supervision and image-level domain kernel (IDK) supervision, by which channel attention in CAMD can be explicitly optimized to emphasize the channels corresponding to the domain of an input image. Furthermore, IDK can be adaptively updated when training DCANet, thereby improving the generalization ability to unseen scenes. Experimental results on benchmark datasets show that our DCANet performs favorably for handling multi-domain datasets using one single model. Moreover, our IDK training strategy can be applied to boost state-of-the-art methods on single domain dataset.**

***Index Terms*—Crowd counting, multi-domain learning.**

I. Introduction

Crowd counting, aiming at predicting pedestrian density map and counts from input image, has a wide range of applications in video surveillance, traffic monitoring [1], [2], *etc*. Recently, deep learning-based crowd counting methods [3]–[9] have achieved promising performance. For example, given a dataset such as ShanghaiTech [3], the trained crowd counting models can obtain satisfying accuracy on its corresponding testing set, but usually fail to handle other datasets, *e.g.*, UCF-QNRF [10] or UCF_CC_50 [11]. This is due to that there are noteworthy distribution discrepancy among multiple domains. As shown in Fig. 1, variations in scene contexts, crowd densities and head scales are common in different datasets or even within one dataset. Albeit promising performance on one specific domain, these crowd counting methods are limited in generalizing to other domains and perform worse for unseen scenery in practical applications, which can be validated from the performance of two state-of-the-art methods CANet [12] and DSSINet [13] listed in the first three lines in Table I and Table II.

Fig. 1. Diversity of crowd counting scenes. (a) and (b) are two samples from ShanghaiTech A, (c) and (d) are from ShanghaiTech B and UCF-QNRF respectively. It can been seen that diverse head scales and context variances exist across different datasets and even within one dataset, making it a very challenging task to tackle multi-domain crowd counting using one deep model.

Cross-domain learning for crowd counting is a possible solution to improve generalization ability on target domain. In particular, few shot learning [14]–[16] and domain adaption [17]–[19] can be exploited to improve the counting accuracy on target domain. However, cross-domain crowd counting ignores the performance degradation on source domain. Meanwhile, these deep models may overfit to one specific domain, considering small amount of images in crowd counting datasets. In [20], Marsden *et al*. proposed to relieve the forgetting problem on source domain by introducing a domain classifier, based on which corresponding counting branch is activated. However, it heavily relies on the classification accuracy and several counting branches also result in cumbersome network architecture. Thus, it is a challenging, valuable but unresolved task to simultaneously obtain good performance on multiple domains, *i.e.*, multi-domain crowd counting. An intuitive strategy is to directly train the model by combining all the training images of multiple domains. However, multi-domain crowd counting cannot be easily tackled by observing the incapability of consistent performance improvement from the last line in Table I and Table II.

In this paper, we propose a Domain-guided Channel Attention Network (DCANet) to address multi-domain crowd counting, where network architecture and learning strategies are specifically developed to exploit the knowledge from multiple domains. First, in terms of network architecture, our DCANet consists of three modules, *i.e.*, feature extraction,

This work was supported by National Natural Science Foundation of China under Grants No. U19A2073 and No. 62172127). We give special thanks to Alibaba Group for their support to this work.

Z. Yan, D. Ren and W. Zuo are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: yanzhaoyi@outlook.com, rendongweihit@gmail.com, cswmzuo@hit.edu.cn).

The e-mails of P. Li and B. Wang are lipengyu007@gmail.com and wangbiao225@foxmail.com, respectively.

Corresponding author: Dongwei Ren

channel attention-guided multi-dilation (CAMD) and density map prediction. As shown in Fig. 2, our DCANet is very concise. CAMD extracts features from multiple domains. Specifically, the multiple dilation rates in convolution kernels is beneficial to deal with the head scale variance. And according to [21], different channels in multi-dilation convolutions would response when receiving different scene contexts, crowd densities and head scales. To this end, the domain variance can be well modeled by CAMD. Therefore, as suggested in [22], the active parameters in deep models are usually sparse. And for crowd counting, we suggest that different channels in multi-dilation convolutions would response when receiving different scene contexts, crowd densities and head scales. Therefore, we introduce a simplified channel attention module to indicate the importance of deep features after multi-dilation convolutions. DCANet is expected to adapt the bias of domain distribution caused by the crowd density, illumination, perspective, *etc*.

Second, to better exploit the attention mechanism in CAMD module, we further propose two learning strategies to explicitly impose supervision on channel attention map. In particular, dataset-level domain kernel (DDK) and image-level domain kernel (IDK) are proposed to act as extra supervision signals when training DCANet. As for DDK, we assume that images from one dataset are with the same domain, and thus the importance of convolution channels can be computed based on a pre-trained DCANet model. Then DDK can be adopted to guide the channel attention in CAMD module. Furthermore, the manually split datasets still have variations, as shown in Fig. 1, based on which we propose IDK to adaptively update the channel attention supervision for individual image. By adopting IDK, our DCANet can obtain satisfying performance for multiple domains and generalize better to unseen domains.

Experimental results on benchmark datasets show the effectiveness of our DCANet. On unseen domains, our DCANet exhibits better generalization ability in comparison with single domain methods, and is comparable with cross-domain methods. On training domains, our DCANet can achieve comparable or better performance in comparison with single domain crowd counting methods specifically trained for the corresponding dataset.

To sum up, the main contributions of this work are as follows:

- We propose a domain-guided channel attention network (DCANet) to address multi-domain crowd counting. To the best of our knowledge, DCANet is the first work to tackle multi-domain crowd counting using one single deep model.
- Two novel learning objectives, *i.e*., dataset-level domain kernel (DDK) and image-level domain kernel (IDK), are proposed to guide the training of DCANet, making it effective in handling multiple domains and generalizing to unseen domains.
- Extensive experiments are conducted to validate the effectiveness of our DCANet against single domain and cross-domain crowd counting methods. Also with better re-split of different domain samples, the performance of DCANet can be further boosted.

The remainder of this paper is organized as follows. Section II briefly reviews relevant works. Section III presents the proposed method towards learning multi-domain crowd counting. Section IV conducts experiments along with analysis. Finally, Section V ends this paper with concluding remarks.

II. Related Work

In this section, we briefly review relevant works of CNN-based crowd counting and cross-domain crowd counting, as well as multi-domain learning for other computer vision tasks.

A. CNN-based Crowd Counting

Currently, various CNN-based methods have been suggested to train a better crowd estimator from various aspects, such as network architectures [3]–[6], [6], [7], [23]–[25], Graph Neural Network (GNN) [9], Neural architecture search (NAS) [26], loss functions [27]–[29], perspective information [30]–[34], *etc*. To alleviate the scale variance, Zhang *et al*. [3] presented a multi-column architecture for simple fusion of deep features with different sizes of receptive fields. CSRNet [35] stacks several dilated convolutions after truncated VGG [36] to enlarge receptive fields. Cao *et al*. [37] proposed scale aggregation modules to extract multi-scale features for accurate crowd count. Song *et al*. [7] adopted U-shape backbone to predict density maps from features of different scales and then combined them to estimate the final count. Luo *et al*. [9] suggested to use GNN to model scale variance and Hu *et al*. [26] proposed to adopt NAS technique to search the optimal architecture. Apart from exploring various architectures, Cheng *et al*. [27] aimed to replace Euclidean distance with maximum excess over pixels (MEP) loss and achieved promising performance. To avoid the intrinsic limitations of density maps, Ma *et al*. [28] proposed Bayesian loss which constructed a density contribution probability model from point annotations. And Wang *et al*. [29] proposed optimal transport loss to model the discrepancy between the estimated maps and ground-truth maps. Despite of different network architectures and training losses, these methods aim at tackling single-domain crowd counting, and fail to generalize to multiple domains.

B. Cross-domain Learning in Crowd Counting

Cross-domain learning mainly includes one/few shot learning [14]–[16], domain adaption [17]–[19], *etc*. [15] presented a meta-learning inspired approach to solve the few-shot scene adaptive crowd counting problem, and [14] further introduced one-shot scene-specific crowd counting. For domain adaption, CODA [17] performs adversarial training with pyramid patches from both source and target domains, so as to tackle different object scales and density distributions. Wang *et al*. [18] released a large synthetic dataset (GCC), and proposed SE CycleGAN to bridge the domain gap between the synthetic and real data. Gao *et al*. [19] proposed multi-level feature aware adaptation (MFA) and structured density map alignment (SDA) to extract domain invariant features and make density maps with a reasonable distribution on the real domain. However, domain adaption methods [17]–[19] focus

on improving performances on target domains and suffer from catastrophic forgetting [38], [39], giving rise to unsatisfying performances on source domains. To mitigate the catastrophic forgetting issue, Marsden *et al*. [20] proposed domain specific branches to process the backbone feature under the guidance of a classification network. Cross-domain methods cannot well address multiple domains and unseen domains.

C. Multi-domain Learning

Multi-domain learning aims at improving the performance over multiple domains. The initial idea of multi-domain learning is to train one model based on all the data from multiple sources [40]. To this end, domain-invariant features are extracted by the shared backbone, and domain-related representations are captured by the domain-specific branches. Similarly, cross-stitch network [41] proposes cross-stitch units that can learn an optimal combination of shared and domain-specific representations. Xiao *et al*. [42] argued that the domain-specific information can be further embedded into domain-related neurons and proposed domain guided dropout layer (DGD) for adaptive selection of effective neurons for each domain. Recently, Rebuffi *et al*. [43] proposed adapter residual modules to enable a high-degree of parameter sharing among domains. MDLCC [44] is proposed by introducing device-specific channel re-weighting module, which adopts the camera-specific characteristics to re-weight the common features. Apart from these methods, other works aim to learn domain-invariant representations while preserving the domain-specific representations. Liu *et al*. [45] proposed an adversarial multi-task learning to mitigate the shared and private latent spaces from interfering with each other via orthogonal regularization. Chen *et al*. [46] proposed to combine negative log-likelihood loss and the ℓ_2-norm loss with the adversarial loss.

Despite of extensive studies on multi-domain learning, multi-domain crowd counting remains not well addressed. Due to the large variances of the characteristics on multiple domains in crowd counting, it is not trivial to directly adopt existing multi-domain methods from other fields to tackle multi-domain crowd counting.

III. Proposed Method

Let us first define the multi-domain crowd counting problem. Given training images from M domains, the multi-domain dataset is constituted with multiple sub-datasets $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_M\}$. For each sub-dataset $\mathcal{D}_m$, it consists of N_m input images $\boldsymbol{I}_i$ and their corresponding ground-truth density maps $\boldsymbol{Y}_i^{gt}$, *i.e*., $\mathcal{D}_m = \{(\boldsymbol{I}_i, \boldsymbol{Y}_i^{gt})\}_{i=1}^{N_m}$. Multi-domain crowd counting aims to learn a single model to perform well on all the testing images from M domains. Also it is expected to generalize better to unseen domains than single domain methods. In this section, we first propose a concise network DCANet for multi-domain crowd counting, and then propose two domain-guided training strategies to explicitly exploit the knowledge of multiple domains for benefiting the training of DCANet.

A. Domain-guided Channel Attention Network

As shown in Fig. 2, DCANet $\mathcal{F}$ with parameters $\boldsymbol{\Theta}$ generally consists of three modules, *i.e*., feature extraction, channel attention-guided multi-dilation (CAMD), and density map prediction. For a given input image $\boldsymbol{I}$ with spatial size $H \times W$, DCANet can predict its crowd density map $\hat{\boldsymbol{Y}}$ with size $\frac{H}{2} \times \frac{W}{2}$ by forward pass through three modules.

1) Feature Extraction Module: Since the following two modules are with lightweight parameters, feature extraction module plays a crucial role for obtaining sufficient deep features for crowd counting [47]. Apart from the commonly-used VGG-based [36] architecture, several feature extraction modules have been developed for better crowd counting, such as encoder-decoder based [48], UNet-based [47], [49] and DenseNet-based [50] architectures, *etc*. In this work, we adopt a truncated HRNet-W40-C [51] (from *Stage1* to *Stage3*) with a convolutional layer (stacked behind *Stage3* to reduce the channel number to be C) as the feature extractor in our DCANet, resulting in deep features with size $\frac{H}{4} \times \frac{W}{4} \times C$, where C is channel number.

2) Channel Attention-guided Multi-dilation Module: Considering diverse crowd densities and head scales in multiple domains, multi-dilation convolutions are generally suggested to extract multi-scale features [52], [53]. Specifically, the multi-dilation network is comprised of 4 branches with dilation rates 2, 4, 6 and 8, respectively. These four features are concatenated as the final multi-scale features with dimension $\frac{H}{4} \times \frac{W}{4} \times 4C$[1], by which variations such as different head scales can be captured by different channels.

It is straightforward to directly predict crowd density map based on the multi-scale features. However, multiple domains have much more diverse contexts, and are very difficult to model only using multi-dilation convolutions. Motivated by the sparse features in deep models, we suggest that the convolution channel responses vary with different domains, and thus propose to further introduce channel attention map to guide the activation of deep features of multi-dilation convolutions. Specifically, we adopt a channel attention module to generate a map $\boldsymbol{m}$ with size $1 \times 1 \times 4C$ to indicate the importance of the multi-scale features. In comparison to the attention module in [54], our channel attention module is much more simple, in which only one convolutional layer with $4C$ output size is adopted. Then global average pooling and `sigmoid` are adopted to project it as a $1 \times 1 \times 4C$ channel attention map $\boldsymbol{m}$. The channel attention map can then be used to re-weight the features of multi-dilation convolutions, and it act like self-attention to measure the importance of deep features for different domains.

3) Density Map Prediction Module: The features from CAMD module are finally fed to a simple density map predictor, with only three convolutional layers, and we finally upsample the density map $2\times$ as the final estimated density map $\hat{\boldsymbol{Y}}$.

Training baseline DCANet. It is straightforward to adopt

[1]We omit the dimension of batchsize for simplicity.

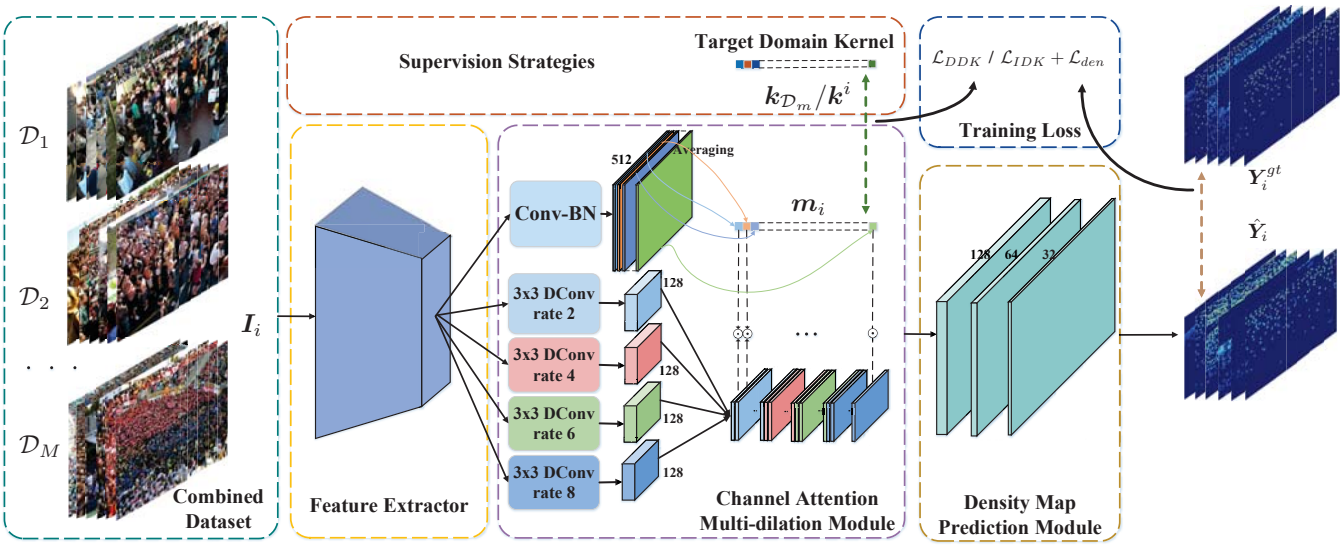


Fig. 2. Architecture of DCANet. Here, "$\odot$" indicates element-wise multiplication and "DConv" means dilated convolution. In the training stage, the feature extractor takes images from multi-domain dataset as the input. The output of the feature extractor serves as two roles. On the one hand, they are used to estimate the image-specific attention $\boldsymbol{m}_i$ for each image I_i. On the other hand, they serve as the input of CAMD module. The domain channel attention loss $\mathcal{L}_{DDK}$ / $\mathcal{L}_{IDK}$ encourages $\boldsymbol{m}_i$ close to domain kernel $\boldsymbol{k}_{\mathcal{D}_m}$ / $\boldsymbol{k}^i$. The output features of the CAMD module are re-weighted by $\boldsymbol{m}_i$. Finally, the density map predictor takes the re-weighted features as the input and outputs the estimated density maps.

MSE loss to train DCANet,

$$\mathcal{L}_{den} = \min_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|\hat{\boldsymbol{Y}}_i - \boldsymbol{Y}_i^{gt}\|^2, \tag{1}$$

where $\hat{\boldsymbol{Y}}_i = \mathcal{F}(\boldsymbol{I}_i; \boldsymbol{\Theta})$ is the predicted density map using DCANet, $\boldsymbol{Y}_i^{gt}$ is the ground-truth density map of input image $\boldsymbol{I}_i$. By the naive training strategy, proper channel attention is expected to be implicitly learned in CAMD module to guide the feature selection of multi-dilation convolutions. However, self-attention is still limited in learning the diverse knowledge from multiple domains.

B. Domain-guided Training Strategy

In order to guide the attention map $\boldsymbol{m}$ in CAMD module, we propose two supervision strategies, *i.e.*, dataset-level domain kernel (DDK) supervision and image-level domain kernel (IDK) supervision.

1) Dataset-level Domain Kernel: Motivated by Xiao *et al*. [42], we adopt the *impact score* to demonstrate the effective convolution kernels in multi-dilation convolutions of a baseline DCANet model. Given input $\boldsymbol{I}_i$, let $\boldsymbol{f}^c(\boldsymbol{I}_i)$ be the feature map of c-th channel from multi-dilation convolutions. The prediction count $\hat{P}_i$ and ground-truth count P_i^{gt} can be respectively obtained by integrating $\hat{\boldsymbol{Y}}_i$ and $\boldsymbol{Y}_i^{gt}$. Then, the metric MAE for the predictions in dataset $\mathcal{D}_m$ can be formulated as:

$$MAE = \frac{1}{N_m} \sum_{i=1}^{N_m} ||\hat{P}_i - P_i^{gt}||. \tag{2}$$

Here we further generalize the definition of MAE, and use it for a single image. Denote by $MAE^{\setminus c}(\hat{\boldsymbol{Y}}_i)$ the MAE obtained by setting $\boldsymbol{f}^c(\boldsymbol{I}_i)$ as zero. It can be calculated via Eqn. (2), where N_m is 1. For the c-th convolutional kernel of multi-dilation convolutions, we have

$$s_c(\boldsymbol{I}_i) = MAE^{\setminus c}(\hat{\boldsymbol{Y}}_i) - MAE(\hat{\boldsymbol{Y}}_i), \tag{3}$$

where $s_c(\boldsymbol{I})$ is the impact score of the c-th channel for image $\boldsymbol{I}_i$.

Then we suppress negative values in the impact score and then normalize the whole impact score, which can be formulated as

$$\tilde{s}_c(\boldsymbol{I}_i) = \texttt{softmax}(\max(s_c(\boldsymbol{I}_i), 0)). \tag{4}$$

Fig. 3 shows the t-SNE [55] visualization of the normalized impact score of the training images. It can be seen that the distribution of normalized impact score can be basically categorized into three subsets. Besides, the distribution of ShanghaiTech A [3] overlaps with that of UCF-QNRF [10], while being away from that of ShanghaiTech B [3], which accords with the fact that the images of ShanghaiTech A and UCF-QNRF are searched from the web, while the images of ShanghaiTech B are all captured from the street views. This inspires us to propose DDK based on the impact scores of each domain. Specifically, for the m-th domain $\mathcal{D}_m$, the averaged impact score of channel c is computed as

$$\bar{s}_c^m = \frac{1}{N_m} \sum_{i=1}^{N_m} s_c(\boldsymbol{I}_i), \quad s.t.\ I_i \in \mathcal{D}_m. \tag{5}$$

In this case, the higher value $\bar{s}_c^m$ is, the more vital the c-th convolution kernel is for domain $\mathcal{D}_m$. As the average *impact score* acts as an indicator of the importance of the convolutional kernels for a domain, we propose DDK $\boldsymbol{k}_{\mathcal{D}_m}$ for domain $\mathcal{D}_m$

$$\boldsymbol{k}_{\mathcal{D}_m} = \texttt{softmax}(\max(\bar{s}_c^m, 0)), \tag{6}$$

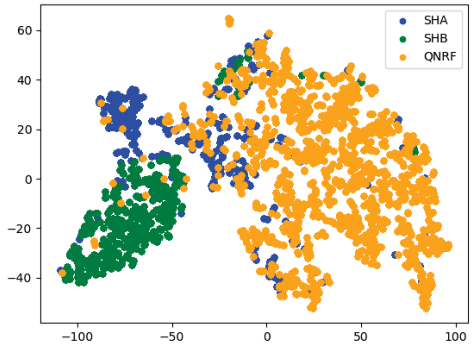


Fig. 3. The t-SNE [55] visualization of normalized impact scores of training samples. Since the images in ShanghaiTech B are captured from street views while the images from ShanghaiTech A and UCF-QNRF are searched from the web, the data distribution of ShanghaiTech A and UCF-QNRF should overlap with each other significantly and are expected away from that of ShanghaiTech B. The visualization basically accords with the prior distribution of three datasets.

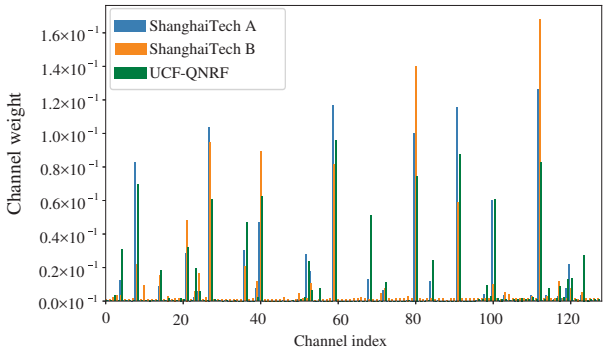


Fig. 4. Visualization of dataset-level domain kernels.

by which we suppress negative values in the impact score and then normalize the averaged impact score for domain $\mathcal{D}_m$.

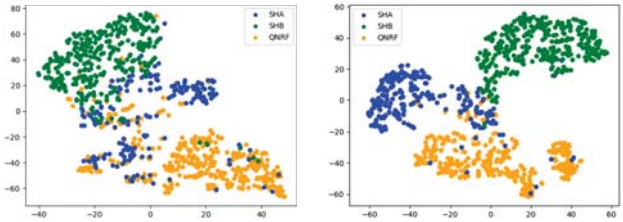
In Fig. 4, domain kernels are visualized for three datasets. It can be seen that the activation of domain kernels is sparse, and domain kernels are relevant with different datasets, which also supports our basic idea, *i.e.*, one single deep model is sufficient to tackle multi-domain crowd counting. In the DDK strategy, the images are categorized into different domains based the corresponding manually-split datasets. For each domain $\mathcal{D}_m$, the domain kernel $\boldsymbol{k}_{\mathcal{D}_m}$ is calculated via Eqn. (6). The domain kernel $\boldsymbol{k}_{\mathcal{D}_m}$ works as a fixed supervision signal in the whole training phase.

Then for any domain $\mathcal{D}_m$, its domain kernel $\boldsymbol{k}_{\mathcal{D}_m}$ can be computed, and can be used to act as the supervision signal for attention map $\boldsymbol{m}$ in CAMD module. Specifically, $\mathcal{L}_{DDK}$ is defined as:

$$\mathcal{L}_{DDK} = \min_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|\boldsymbol{m}_i - \boldsymbol{k}_{D_m}\|^2, \tag{7}$$

where $\boldsymbol{m}_i$ is the attention map of $\boldsymbol{I}_i$ in CAMD module. Overall, the learning objective can be defined as

$$\mathcal{L}_D = \mathcal{L}_{den} + \lambda_D \mathcal{L}_{DDK}, \tag{8}$$



(a) DCANet$_{base}$ (b) DCANet($\mathcal{L}_D$)

Fig. 5. The t-SNE visualization of predicted channel attention of DCANet$_{base}$ and DCANet ($\mathcal{L}_D$) on the test sets. It is seen that DDK aims to push all the predicted channel attentions forward to the fixed DDK, and thus DCANet ($\mathcal{L}_D$) is expected to achieve better performance over the multiple datasets.

λ_D is a hyper-parameter to balance $\mathcal{L}_{den}$ and $\mathcal{L}_{DDK}$. The attention supervision DDK will guide the attention module directly to find an optimal status and improves the interpretability of the outputs of the attention module.

2) Image-level Domain Kernel: Despite of the improvements of DDK training strategy, it suffers from limitations. The manual split of datasets cannot guarantee the samples within a dataset are from the same domain, which yields misleading supervision using the fixed DDK. Fig. 5 displays the t-SNE visualization of predicted channel attention of DCANet ($\mathcal{L}_D$) which is learned with DDK strategy. To address this issue, we further propose the IDK strategy, where image-specific domain kernels can be adaptively updated during training DCANet, instead of pushing the attention map to its corresponding fixed DDK.

We mark the IDK for image $\boldsymbol{I}_i$ as $\boldsymbol{k}_i$. Generally speaking, $\boldsymbol{k}_i$ is the linear combination of DDK from multiple domains with adaptive coefficients. The image-specific domain kernel $\boldsymbol{k}^i$ for the input image $\boldsymbol{I}_i$ is formulated as

$$\boldsymbol{k}_i = \sum_{m=1}^{M} \mu_m^i \boldsymbol{k}_{\mathcal{D}_m}, \tag{9}$$

where M is the number of domains, μ_m^i is the weight of domain kernel $\boldsymbol{k}_{\mathcal{D}_m}$ and is defined as the normalized cosine similarity between $\boldsymbol{m}_i$ and $\boldsymbol{k}_{\mathcal{D}_m}$

$$\mu_m^i = \frac{\exp(\tilde{\mu}_m^i)}{\sum_{m=1}^{M} \exp(\tilde{\mu}_m^i))}, s.t.\ \tilde{\mu}_m^i = \frac{\langle \boldsymbol{m}_i, \boldsymbol{k}_{\mathcal{D}_m} \rangle}{\|\boldsymbol{m}_i\|_2 \|\boldsymbol{k}_{\mathcal{D}_m}\|_2} \tag{10}$$

where $\langle \cdot, \cdot \rangle$ is the inner product operator.

Analogously, we define IDK-based loss function as

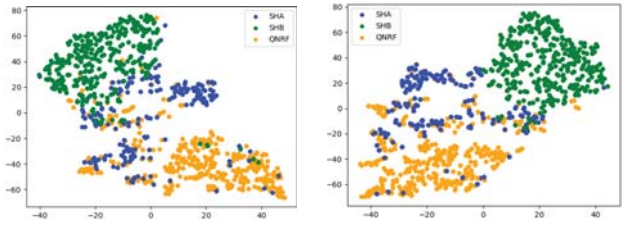
$$\mathcal{L}_{IDK} = \min_{\boldsymbol{\Theta}, \boldsymbol{k}_{\mathcal{D}_m}} \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|\boldsymbol{m}_i - \boldsymbol{k}_i\|^2. \tag{11}$$

We emphasize that one of the main differences between $\mathcal{L}_{IDK}$ and $\mathcal{L}_{DDK}$ lies in the adaptive $\boldsymbol{k}_i$, where the coefficients can be adaptively updated.

Finally, the overall loss function $\mathcal{L}_I$ is the combination of $\mathcal{L}_{den}$ and $\mathcal{L}_{IDK}$, defined as

$$\mathcal{L}_I = \mathcal{L}_{den} + \lambda_I \mathcal{L}_{IDK}. \tag{12}$$

Here are some notes on the merits of IDK. (i) Image-level domain kernel $\boldsymbol{k}^i$ is constructed for each input image $\boldsymbol{I}_i$ during



(a) DCANet$_{base}$ (b) DCANet($\mathcal{L}_I$)

Fig. 6. The t-SNE visualization of predicted channel attention of DCANet$_{base}$ and DCANet ($\mathcal{L}_I$) on the test sets. When adopting IDK training strategy, the predicted attention are moving towards to its adaptive instance-aware domain kernel, making the distribution space of IDK more flexible than that of DDK, which accords with higher performance of DCANet ($\mathcal{L}_I$) on the unseen datasets in Table III.

training. Thus, different images have their own specific image-level domain kernels, even though the images may come from the same dataset. (ii) In contrast to the model learned with $\mathcal{L}_D$, model trained with $\mathcal{L}_I$ performs better in multi-domain training and can generalize better to unseen images. Fig. 6 shows the t-SNE visualization of predicted channel attention of DCANet ($\mathcal{L}_I$) which is learned with IDK strategy.

3) Re-split of Muti-domain Datasets: Since the impact score indicates the significance of the convolutional kernels for the corresponding input image, it is intuitive to further investigate IDK strategy by performing clustering over the training images based on the corresponding impact scores. Concretely, we normalize the impact score (obtained on the baseline model of DCANet) of each image $\boldsymbol{I}_i$ via Eqn. (4). The normalized impact scores are then used to cluster via KMeans++ [56], by setting S subsets. Afterwards, we compute the domain kernels of each subset and perform IDK training, respectively. The trained model is termed as DCANet ($sub=S$).

To further exploit the potential of clustering, it is reasonable to obtain the impact scores of input images based on DCANet ($sub=S$) instead of DCANet$_{base}$ and then repeat the steps of computing domain kernels of each subset and perform IDK training on the model DCANet ($sub=S$). The trained model is denoted as DCANet ($sub=S$, $irs=1$), where irs stands for *iterative re-splitting*. For naming consistency, DCANet ($sub=S$) can also be re-written as DCANet ($sub=S$, $irs=0$) . Accordingly, we can obtain DCANet ($sub=S$, $irs=t+1$) when such a training loop is conducted based on DCANet ($sub=S$, $irs=t$), where t is the index of iterative re-splitting.

C. Pipeline of training full DCANet

First, on the training dataset from M domains, we train baseline DCANet model using Eqn. (1) for 400 epochs, based on which DDK can be accordingly computed for the subsequent training procedure. The baseline model is named as DCANet$_{base}$. Second, DCANet$_{base}$ can be further trained via DDK training loss Eqn. (8) for 100 epochs, resulting in DCANet ($\mathcal{L}_D$). Third, DCANet ($\mathcal{L}_D$) is further updated through IDK training loss Eqn. (12) for 100 epochs, resulting in DCANet ($\mathcal{L}_I$). For training DCANet ($sub=S$), we firstly explore the performances with different number of subsets to finally determine the suitable value of S. Then we further perform iterative re-splitting training with the determined number of subsets as stated in Sec. III-B3, and obtain DCANet ($sub=S$, $irs=t$) where t is the index of iterative re-splitting. The training epoch of DCANet ($sub=S$, $irs=t$) is also set as 100.

IV. Experiments

In this section, we first present details of implementation and datasets, then compare DCANet with the state-of-the-art crowd counting methods. Finally, ablation studies are conducted to validate the effectiveness of our proposed DDK and IDK in improving the performances on multi-domain dataset and even unseen dataset. The source code of DCANet is publicly available at https://github.com/Zhaoyi-Yan/DCANet.

A. Datasets

Multi-domain Dataset. To evaluate the performance of multi-domain crowd counting, we build our multi-domain dataset by setting $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ as ShanghaiTech A [3], ShanghaiTech B [3] and UCF-QNRF [10], respectively. Concretely, the training images of these three datasets are merged as the training data of DCANet$_{base}$. The trained model is then evaluated on testing images of the three datasets, respectively. These three datasets are taken as the observed domains. To further evaluate the generalization of the trained model, we evaluate the performance of the trained model on UCF_CC_50 and WorldExpo'10. These two datasets are employed as unseen domains in the paper. We adopt MAE and RMSE as the evaluation metrics for all datasets, which is consistent with [12], [35]. The detailed descriptions of these five datasets are listed as below.

ShanghaiTech [3] consists of 1,198 images, among which 482 images are collected from the web (Part A) and 716 images are captured from street views (Part B). Following the common settings [3], 300 images in Part A and 400 images in Part B are used for training, while the remaining images are used for testing. To make brief notations, Part A and Part B of ShanghaiTech are abbreviated as **SHA** and **SHB**, respectively.

UCF-QNRF [10] consists of 1,535 images, among which 1,201 images are used for training while the others for testing. This dataset is very challenging, since it contains 1,251,642 people with various head scales, densities and viewpoints. The name **UCF-QNRF** is abbreviated as **QNRF** in this paper.

UCF_CC_50 [11] only includes 50 images for testing. Due to the diverse scenes, this dataset is suitable to evaluate the performance of crowd counting for multiple domains. This dataset contains 50 images of diverse scenes. This dataset is only used as testing, so we evaluate our model on the whole dataset (*i.e.*, 50 images). **UCF_50** is the abbreviation of UCF_CC_50 if necessary.

WorldExpo'10 [57] comprises 3,980 annotated frames extracted from surveillance videos. Five testing scenes are available with a Region of Interest (ROI) in each scene. We only report the average MAE of five testing scenes due to the limited space. **WE'10** is adopted as the abbreviation of WorldExpo'10.

TABLE I
MAEs OF CANET FOR VARIOUS COMBINATIONS OF TRAINING AND TESTING SETTINGS. "⋆" INDICATES THE PERFORMANCE OF EVALUATION ON THE SINGLE DOMAIN.

Train \ Test	SHA	SHB	QNRF	UCF_50	WE'10
SHA	62.3⋆	29.5	167.0	335.9	30.4
SHB	138.7	7.8⋆	256.8	630.0	37.3
QNRF	78.6	35.9	107.0⋆	367.1	47.6
Multi-domain set	61.8	10.6	101.2	345.8	15.2

TABLE II
MAEs OF DSSINET FOR VARIOUS COMBINATIONS OF TRAINING AND TESTING SETTINGS. "⋆" INDICATES THE PERFORMANCE OF EVALUATION ON THE SINGLE DOMAIN.

Train \ Test	SHA	SHB	QNRF	UCF_50	WE'10
SHA	60.6⋆	21.7	199.9	425.6	39.9
SHB	148.9	6.9⋆	332.1	709.0	46.3
QNRF	65.8	12.5	99.1⋆	420.5	24.7
Multi-domain set	60.1	9.0	94.6	332.4	14.5

B. Implementation Details

We adopt fixed Gaussian kernel of size 15×15 to generate ground-truth density maps. We implement our DCANet with Pytorch [58], and use Adam [3] optimizer with fixed learning rate 0.0001. The batch size is 16. For images in ShanghaiTech, if the shorter side of the image is smaller than 416, we resize them to make the shorter side be 416. Besides, for those images that are too large in UCF-QNRF, we resize them and make the side length no larger than $2,048$. We do not change the aspect ratio when performing resizing operation. After resizing, random horizontal flipping and color jittering are also applied. Finally we perform the random cropping with patch size 400×400. For WorldExpo'10 and UCF_CC_50, we only use the corresponding images for evaluating the generalization ability of the model. We resize the images in UCF_CC_50 in the same way as ShanghaiTech does. For WorldExpo'10 whose images are with fixed size 576×720, following [35], each image and its dot maps are masked with ROI during preprocessing. Empirically, we find that DCANet by setting $C = 128$ shows satisfying performance and runtime. Besides, λ_D and λ_I are all set to 1.

C. Difficulty of Multi-domain Crowd Counting

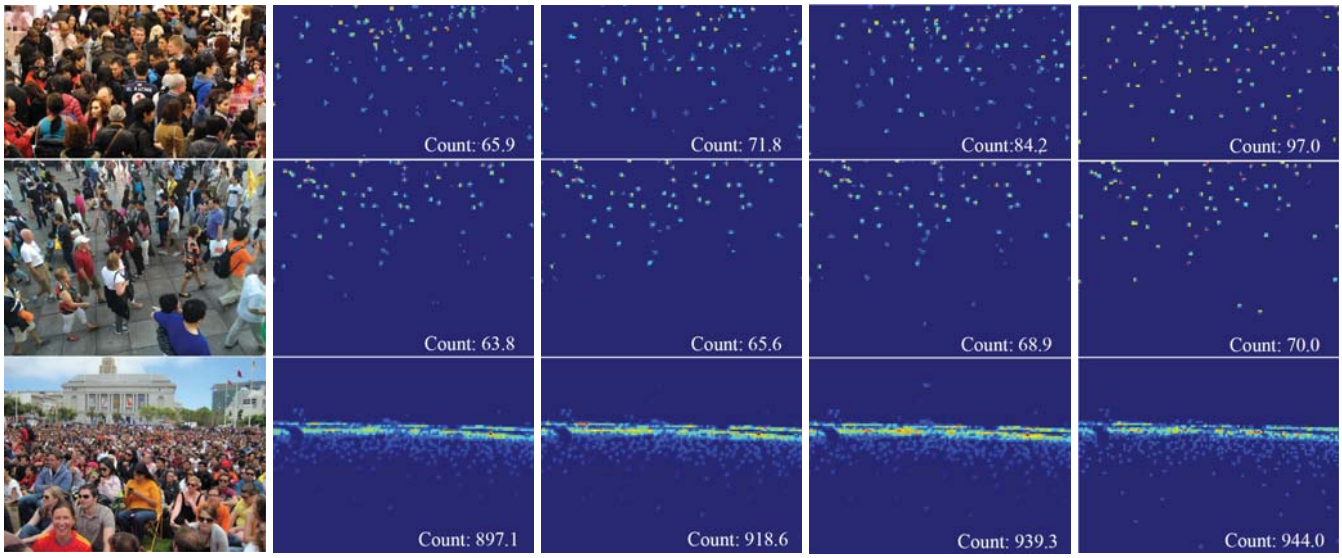
Before presenting comparison experiments, we first analyze the performance of two state-of-the-art single domain methods, *i.e.*, CANet [12] and DSSINet [13], when handling multi-domain datasets. In Table I and Table II, we report the performance of CANet [12] and DSSINet [13][2], when the model is trained on a certain dataset while testing on multiple datasets. Although they deliver high performance for each testing set with the same distribution as the training set, both two methods yield significant performance drops when distribution discrepancy exists between the training and testing sets. For example, compared with the setting that training and testing on ShanghaiTech B, CANet / DSSINet show $278.2\%/214.5\%$ increase on MAE when trained on ShanghaiTech A and tested on ShanghaiTech B. These single domain crowd counting methods are expected to perform worse on unseen domains, restricting their practical applications. Besides, when we train the model with the multi-domain dataset (*i.e.*, refer to IV-A for details), we notice that the model does not produce consistent performance gains across all testing sets, showing that multi-domain crowd counting learning is difficult to achieve merely by mixing all the training data. Such observations encourage us to tackle multi-domain crowd counting using one single deep model.

[2]Our reported results are based on the officially released training codes and pre-trained models.

D. Multi-domain Crowd Counting

1) Comparison with State-of-the-arts: To verify the effectiveness of our proposed DCANet, we compare our method with single-domain methods CSRNet [35] and CANet [13], cross-domain method DSM [20] and multi-domain method DGD [42]. Because the source code of DSM is not released and DGD for person re-identification cannot be applied to crowd counting, we make the following modifications. The major contribution of DSM [20] lies in constructing the domain-specific branch for each domain, and an extra domain classifier is also trained to classify the input to help select the appropriate branch. We build the domain-specific branches by tripling the CAMD module and removing all the attention prediction branches, besides an extra domain classifier is also constructed as DSM. The modified method is named as DSM*. DGD [42] is an approach to tackle multi-domain person re-identification, where the channel dropout mechanism is the essential part in tackling the multi-domain problem. We modify DCANet by replacing the attention mechanism with channel dropout to select the effective channels instead of re-weighting the features, which is termed as DGD*. Table III lists the results of these methods and our proposed DCANet.

By comparing our method with the competing approaches, we have the following three observations. (i) It can be seen that single domain methods (*i.e.*, CSRNet [35] and CANet [13]) only perform well on the observed domains, however, fail to generalize to unseen domains. (ii) DSM* performs worse than our method on the observed domains, mainly ascribing to the imperfect accuracy of the domain classifier. Besides, for unseen domain images, DSM* has to select an improper module via the domain classifier, resulting in large performance drops on UCF_CC_50 and WorldExpo'10. (iii) As for DGD*, it achieves generally satisfying performances in both observed and unseen domains. There is still much room for performance improvement. In fact, channel dropout in DGD* and image-specific channel attention in DCANet can be separately regarded as *hard* and *soft* ensemble learning. According to [59], [60], *soft* ensemble learning is superior to the *hard* one, which is in accord with the higher performances of DCANet ($\mathcal{L}_I$) on observed domains and even on unseen domains.

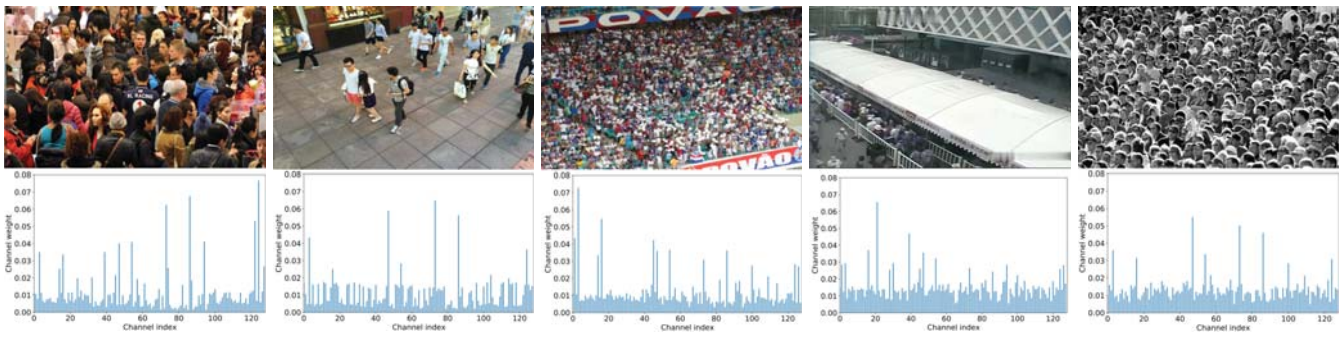


(a) Input images (b) DCANet (baseline) (c) DCANet ($\mathcal{L}_D$) (d) DCANet ($\mathcal{L}_I$) (e) Ground-truth

Fig. 7. Illustration of density maps predicted by different methods. (a) input images, (b), (c) and (d) are the density maps predicted by DCANet (baseline), DCANet ($\mathcal{L}_D$) and DCANet ($\mathcal{L}_I$), respectively. (e) ground-truth dot annotations.It is seen that DCANet ($\mathcal{L}_I$) outperforms the others.

TABLE III
COMPARISON WITH STATE-OF-THE-ARTS TRAINED ON THE MULTI-DOMAIN DATASET FROM SHA/SHB AND QNRF. **UCF_50 AND WE'10 WORK AS THE UNSEEN DOMAINS IN TESTING AND ARE USED FOR EVALUATING GENERALIZATION PERFORMANCE.**

Method	Observed domains						Unseen domains		
	SHA		SHB		QNRF		UCF_50		WE'10
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Single-domain Method									
CSRNet [35]	68.4	107.2	13.5	18.9	101.1	176.1	343.1	469.8	16.3
CANet [12]	66.4	103.1	12.8	20.6	104.2	170.5	340.9	486.3	15.1
Multi-domain Method									
DGD* [42]	59.2	101.4	8.4	13.5	94.6	167.8	326.3	446.4	14.8
Cross-domain Method									
DSM* [20]	59.8	100.8	8.5	14.3	94.1	166.7	376.3	452.8	19.7
Our Method									
DCANet$_{base}$	62.5	99.3	9.4	14.8	95.9	170.6	330.7	453.5	15.9
DCANet ($\mathcal{L}_D$)	59.0	**99.2**	7.9	12.9	93.8	164.9	345.1	460.5	17.3
DCANet ($\mathcal{L}_I$)	**58.3**	99.3	**7.2**	**11.8**	**88.9**	**160.2**	**309.6**	**431.4**	**12.4**
DM-Count$_{base}$	61.1	101.5	8.4	13.0	84.3	156.3	311.2	422.6	14.7
DM-Count ($\mathcal{L}_D$)	58.2	98.6	7.0	11.1	82.2	147.3	341.5	450.7	17.0
DM-Count ($\mathcal{L}_I$)	**56.8**	**97.4**	**6.1**	**10.3**	**80.4**	**144.1**	**296.6**	**412.5**	**11.5**



(a) ShanghaiTech A (b) ShanghaiTech B (c) UCF-QNRF (d) WorldExpo'10 (e) UCF_CC_50

Fig. 8. From top to bottom are inputs and the corresponding channel attentions predicted by DCANet ($\mathcal{L}_I$), respectively sampled from ShanghaiTech A / B, UCF-QNRF and unseen dataset WorldExpo'10, UCF_CC_50.

2) Comparison of DCANet variants with different training strategies: As for our proposed DCANet, although it is concise, when embedded with a CAMD module, better results can be achieved than single domain methods. From the last few rows in Table III, when compared with baseline DCANet, it is seen that DCANet ($\mathcal{L}_D$) shows better performances on the observed domains, which is mainly due to the introduction of DDK supervision. However, DCANet ($\mathcal{L}_D$) delivers inferior performances on the unseen domains compared to baseline DCANet. This is mainly attributed to the limitations of static DDK when handling images from unseen domains. Finally, DCANet ($\mathcal{L}_I$) outperforms baseline DCANet with 4.2, 2.2, 7.0 MAE decreases on the observed domain ShanghaiTech A/B, UCF-QNRF, and even with 21.1 and 3.5 MAE decreases on the unseen domain UCF_CC_50 and WorldExpo'10, respectively. Such significant performance gain is mainly creditable to IDK which is adaptive learning during training and is in view of all the information of images in all the domains available, instead of only manually-divided independent domains in DDK. In Fig. 7, we show some testing samples evaluated on model DCANet$_{base}$, DCANet ($\mathcal{L}_D$) and DCANet ($\mathcal{L}_I$). It can be seen that DCANet ($\mathcal{L}_I$) estimates the most accurate density maps and counts.

Furthermore, we try to apply our learning strategy to state-of-the-art single-domain crowd counting method DM-Count [29], whose source code is publicly available. We add CAMD block before the last convolutional layer of DM-Count, and train the model on the multi-domain dataset to get DM-Count$_{base}$. It is noted that our DDK and IDK strategies only provide additional supervisions on the predicted channel attention in CAMD, and we keep the other settings consistent with the original networks. DM-Count ($\mathcal{L}_D$) and DM-Count ($\mathcal{L}_I$) can also be obtained by performing DDK and IDK training strategies. Considering the significant performance gains on both our concise DCANet and the state-of-the-art method DM-Count, our DDK and IDK training strategies are effective in tackling multi-domain crowd counting.

Fig. 8 shows some visualizations of estimated channel attention in DCANet ($\mathcal{L}_I$). In Fig. 8(a), the values of channel attention become larger as the channel index increases. It indicates that the network assigns higher weights to the feature maps with large receptive fields, which is consistent with large head scales of the input. In Fig. 8(b)(c), the head scales are smaller than Fig. 8(a). It can be seen that the predicted channel attention shows larger peak values for convolution channels with smaller dilation rates. Besides, even for unseen datasets WorldExpo'10 and UCF_CC_50 (*i.e.*, Fig. 8(d)(e)), we observe that estimated attention maps of Fig. 8(d) show high responses for relatively smaller indexes compared with Fig. 8(e), corresponding to the head scales of the two images.

3) Re-split of Multi-domain Datasets via Clustering: Table IV demonstrates the performance when the training images are clustered via the corresponding normalized impact scores and then conducted with IDK training, which is detailed illustrated in Sec. III-B3. The number of clusters are respectively set from 3 to 6 subsets, resulting in four models DCANet (sub=3), DCANet (sub=4), DCANet (sub=5) and DCANet (sub=6).

TABLE IV
MAEs OF DCANET WHEN RE-SPLITING.

Method	SHA	SHB	QNRF
DCANet$_{base}$	62.5	9.4	95.9
DCANet ($\mathcal{L}_D$)	59.0	7.9	93.8
DCANet ($\mathcal{L}_I$)	58.3	7.2	88.9
DCANet ($sub = 3$)	58.0	7.1	88.5
DCANet ($sub = 4$)	57.1	6.7	86.7
DCANet ($sub = 5$)	**56.8**	**6.6**	86.2
DCANet ($sub = 6$)	57.0	6.8	**86.1**

On the one hand, we observe that DCANet ($sub = 3$) outperforms DCANet ($\mathcal{L}_I$), which demonstrates the rationality of the re-split. On the other hand, when the number of subsets increases, the performance consistently improves until the number of domain kernels reaches 5. It is noted that DCANet (sub=5) delivers the best average MAE, surpassing DCANet ($\mathcal{L}_D$) with MAE 2.2, 1.3 and 7.6 over the datasets, which shows the effectiveness of re-split. Regarding the performances shown in Table. IV, we eventually decide that clustering with 5 subsets as the basic setting of re-splitting.

4) Iterative Re-split of Multi-domain Dataset via Clustering: To further exploit the the potential of clustering, we perform iterative re-split training procedure illustrated in Sec. III-B3. Table V lists the results. It can be seen that performance improves moderately as the number of re-split increases until irs hits 1. DCANet (sub=5, irs=2) does not appear to have significant performance gains over DCANet (sub=5, irs=2), and thus we adopt (sub=5, irs=1) as our final setting.

E. Single-domain Crowd Counting

Intuitively, DDK and IDK not only can be used in multi-domain learning, but also can be adopted to improve the performance in single domain crowd counting. Table VI lists the performance of our method and state-of-the-arts. The baseline DCANet in single-domain is represented as DCANet$_{base}$(single). Similarly, in single-domain problem, DCANet ($\mathcal{L}_D$, single) and DCANet ($\mathcal{L}_I$, single) can also be trained when the number of domains M degrades to 1, as described in Sec. III-B. It can be seen that our DCANet$_{base}$(single) is concise and is slightly inferior to several state-of-the-art methods. We emphasise that our training strategy is intended for multi-domain learning for crowd counting, rather than increasing the performance of a particular dataset. When applied with IDK strategy, DCANet ($\mathcal{L}_I$, single) attains performance gains of 2.2, 0.9, 9.1, 20.4 and 1.5 MAE decreases on these five datasets, respectively. We note that gains by introducing our IDK become larger with the more diverse scene variance and head scales, *etc*. This also indicates the rationale of our IDK in tackling large variances among the observed domains and even among the unseen domains. Besides, when we substitute DCANet as DM-Count [29], similar performance gains for DDK/IDK training strategies are observed, indicating the effectiveness and robustness of our method.

TABLE V
COMPARISON OF DIFFERENT MODELS OF ITERATIVE RE-SPLITTING. **UCF_50 AND WE'10 WORK AS THE UNSEEN DOMAINS IN TESTING AND ARE USED FOR EVALUATING GENERALIZATION PERFORMANCE.**

Method	Observed domains						Unseen domains		
	SHA		SHB		QNRF		UCF_50		WE'10
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
DCANet$_{base}$	62.5	99.3	9.4	14.8	95.9	170.6	330.7	453.5	15.9
DCANet ($\mathcal{L}_D$)	59.0	99.2	7.9	12.9	93.8	164.9	345.1	460.5	17.3
DCANet ($\mathcal{L}_I$)	58.3	99.3	7.2	11.8	88.9	160.2	309.6	431.4	12.4
DCANet ($sub=5$, $irs=0$)	56.8	97.6	6.6	10.5	86.2	153.3	295.0	413.9	11.1
DCANet ($sub=5$, $irs=1$)	56.2	97.0	**6.4**	**10.2**	**85.1**	**149.7**	**288.5**	400.2	**10.4**
DCANet ($sub=5$, $irs=2$)	**56.1**	**96.7**	6.6	10.7	85.7	151.1	293.6	**398.5**	10.7

TABLE VI
COMPARISON OF OUR METHOD WITH STATE-OF-THE-ARTS TRAINED ON **SINGLE** DATASET.

Method	SHA		SHB		QNRF		UCF_50		WE'10
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MCNN [3]	110.2	173.2	26.4	41.3	277	-	377.6	509.1	11.6
Switch-CNN [23]	90.4	135.0	21.6	33.4	228	445	318.1	439.2	9.4
CP-CNN [24]	73.6	112.0	20.1	30.1	-	-	298.8	320.9	8.9
CSRNet [35]	68.2	115.0	10.6	16.0	-	-	266.1	397.5	8.6
SANet [37]	67.0	104.5	8.4	13.6	-	-	258.4	334.9	8.2
PCC [61]	73.5	124.0	11.0	19.0	-	-	240.0	315.5	7.2
CANet [12]	62.3	100.0	7.8	12.2	107	183	212.2	243.7	7.2
ADCrowdNet [62]	63.2	98.9	8.2	15.7	-	-	266.4	358.0	7.3
SPN [63]	64.2	98.4	7.2	11.1	104.7	173.6	188.4	315.3	-
COBC [64]	62.8	102.0	8.6	16.4	118	192	239.6	322.2	8.2
MAN [65]	61.8	100.0	8.6	13.3	-	-	245.4	349.3	8.3
DSSINet [13]	60.6	96.0	6.9	10.3	99.1	159.2	216.9	302.4	6.7
RPNet [32]	61.2	96.9	8.1	11.6	-	-	-	-	8.2
Bayesian Loss [28]	62.8	101.8	7.7	12.7	88.7	154.8	229.3	308.2	-
LibraNet [66]	55.9	97.1	7.3	11.3	88.1	143.7	181.2	262.2	-
DM-Count [29]	59.7	95.7	7.4	11.8	85.6	148.3	211.0	291.5	-
ADSCNet [67]	55.4	97.7	6.4	11.3	71.3	132.5	198.4	267.3	-
DCANet$_{base}$ (single)	61.4	108.8	8.7	15.4	99.2	177.7	203.6	318.1	7.0
DCANet ($\mathcal{L}_D$, single)	60.6	101.0	8.1	13.9	93.3	168.4	191.2	303.7	6.7
DCANet ($\mathcal{L}_I$, single)	59.2	94.4	7.8	12.3	90.1	150.4	183.2	260.1	6.5
DM-Count$_{base}$ (single)	59.4	96.3	7.3	11.6	85.1	150.5	207.4	288.0	6.8
DM-Count ($\mathcal{L}_D$, single)	58.0	96.7	6.7	11.1	81.9	146.6	187.3	294.1	6.3
DM-Count ($\mathcal{L}_I$, single)	57.5	94.9	6.5	10.6	81.0	147.0	182.6	268.9	6.1

TABLE VII
MAES OF OUR MODEL WITH DIFFERENT MODULES AND TRAINING STRATEGIES.

Network component and training strategy			Evaluation dataset				
Multi-dilation Module	Channel-attention Module	Extra Supervision	**Observed domain**			**Unseen domain**	
			SHA	SHB	QNRF	UCF_50	WE'10
			63.4	10.8	97.1	347.6	16.3
✔			62.5	9.4	95.9	330.7	15.9
✔	Self-attention [54]		61.4	8.5	95.6	327.6	15.2
✔	Simple-attention		61.9	8.3	96.1	322.0	14.7
✔	Simple-attention	DDK	59.0	7.9	93.8	345.1	17.3
✔	Simple-attention	IDK	**58.3**	**7.2**	**88.9**	**309.6**	**12.4**

F. Ablation Study

1) Effectiveness of the modules in DCANet: In Table VI-I, different modules of DCANet and two proposed training strategies are respectively evaluated. We note that our CAMD module is comprised of a multi-dilation module and a simplified channel-attention module. From the results, it can be observed that the multi-dilation module improves the performance. From the 3rd and 4th rows in Table VII, it is seen that our simplified channel-attention module achieves similar performance with original attention (under the settings of SENet [54] with a ratio $r = 4$). Besides, both DDK and IDK training strategies outperform conventional attention /

TABLE VIII
PERFORMANCE OF SINGLE DILATION IN CHANNEL ATTENTION MODULE.

Network	Loss	SHA	SHB	QNRF
Single-dilation Channel Attention Module				
DCANet ($rate = 2$)	$\mathcal{L}_{den}$	63.3	10.3	98.5
	$\mathcal{L}_D$	60.4	9.4	96.9
	$\mathcal{L}_I$	60.0	8.6	93.6
DCANet ($rate = 4$)	$\mathcal{L}_{den}$	62.7	9.4	95.2
	$\mathcal{L}_D$	59.6	8.7	94.0
	$\mathcal{L}_I$	58.7	7.7	90.5
DCANet ($rate = 6$)	$\mathcal{L}_{den}$	63.0	9.1	97.0
	$\mathcal{L}_D$	60.3	8.0	95.3
	$\mathcal{L}_I$	59.5	7.4	91.2
DCANet ($rate = 8$)	$\mathcal{L}_{den}$	64.1	9.6	99.2
	$\mathcal{L}_D$	61.1	8.6	97.5
	$\mathcal{L}_I$	60.3	8.1	93.1
Multi-dilation Channel Attention Module				
DCANet	$\mathcal{L}_{den}$	62.5	9.4	95.9
	$\mathcal{L}_D$	59.0	7.9	93.8
	$\mathcal{L}_I$	**58.3**	**7.2**	**88.9**

TABLE IX
MAES OF IDK STRATEGY DIRECTLY TRAINING ON THE BASELINE MODEL OF DCANET.

Method	SHA	SHB	QNRF
DCANet$_{base}$	62.5	9.4	95.9
DCANet ($\mathcal{L}_D$)	59.0	7.9	93.8
DCANet ($\mathcal{L}_I$)	58.3	7.2	88.9
DCANet ($\mathcal{L}_I$ w/o DDK)	58.5	7.3	89.7

TABLE X
MAES OF IDK STRATEGY WITH RANDOM INITIALIZED DOMAIN KERNELS.

Method	SHA	SHB	QNRF
DCANet$_{base}$	62.5	9.4	95.9
DCANet ($\mathcal{L}_D$)	59.0	7.9	93.8
DCANet ($\mathcal{L}_I$)	58.3	7.2	88.9
DCANet ($k = 3$)	58.8	7.8	91.4
DCANet ($k = 4$)	58.1	7.2	88.5
DCANet ($k = 5$)	**57.7**	**6.9**	**87.7**
DCANet ($k = 6$)	57.9	**6.9**	88.2

TABLE XI
PERFORMANCES OF THE MODELS WHEN USING GCC-SMALL DATASET.

Method	SHA	SHB	QNRF	GCC-small
DCANet$_{base}$ (single)	61.4	8.7	99.2	58.4
DCANet$_{base}$	66.2	12.8	110.5	63.6
DCANet ($\mathcal{L}_D$)	63.2	9.1	103.6	56.6
DCANet ($\mathcal{L}_I$)	**60.1**	**8.2**	**93.0**	**54.8**

simple-attention module in observed domains, which indicates the superiority of our training strategies. For unseen domains, DDK strategy narrows the overall distribution of attention estimation, leading to performance degradation in unseen domain UCF_50 and WE'10. However, our IDK predicts image-specific channel attention for each input, thus delivering better performance in both observed domains and unseen domains.

2) Multi-dilation v.s. Single-dilation Channel Attention Module in DCANet: To enhance the modeling of scale variances, we adopt multi-dilation channel attention module in our DCANet, instead of single-dilation channel attention module. We compare DCANet with DCANet ($rate = r$), where DCANet ($rate = r$) represents sharing the same dilation rate r across all the branches in CAMD. The comparison results are presented in Table VIII. It is seen that when adopting single-dilation setting, the performance degrades moderately, which is mainly due to the lower capacity in modeling the scale variances.

3) IDK Training on the Baseline Model of DCANet: Sunce domain kernels are learnable in IDK training, it is intuitive to exploit the possibility of directly performing IDK training on DCANet$_{base}$. Table IX demonstrates the corresponding performance which is termed as DCANet ($\mathcal{L}_I$ w/o DDK). It is seen that the performance of DCANet ($\mathcal{L}_I$ w/o DDK) only degrades slightly in comparison to DCANet ($\mathcal{L}_I$) with only 0.2, 0.1 and 0.8 MAE increases on the datasets. Such observation validates the flexibility of IDK training and motivates us to directly perform IDK training in conducting iterative re-splitting in Table V.

4) Initialization of IDK: From Eqn. (9), we know that $\boldsymbol{k}_i$ is initialized as the linear combination of all the dataset domain kernels $\boldsymbol{k}_{\mathcal{D}_m}$. Due to the fact that $\boldsymbol{k}_{\mathcal{D}_m}$ is learnable in IDK strategy, it is natural to come up with this thought: can the domain kernels $\boldsymbol{k}_{\mathcal{D}_m}$ be initialized randomly in IDK training? To this end, we conduct the experiments by continuing training baseline model DCANet$_{base}$ by IDK strategy with random initialized domain kernels, as shown in Table X. In order to make comparison with Table IV in Sec. IV-D3, the number of domain kernels is also set to 3, 4, 5, 6, respectively. DCANet ($k{=}m$) indicates that there are m randomly initialized domain kernels when performing IDK training. When $m{=}3$, DCANet ($k = 3$) outperforms DCANet ($\mathcal{L}_D$), indicating the effectiveness of IDK strategy. It is observed that when $m > 3$, the performances of DCANet ($k{=}m$) surpass DCANet ($\mathcal{L}_I$), but are still inferior to the counterparts of "DCANet ($sub = m$)" shown in Table IV. This reveals that a better initialization (*e.g.*, clustering by impact scores) further improves the performance. Even with random initialized domain kernels, we emphasize that DCANet ($k = 4$) still achieves significant performance gains with MAE 4.8, 2.5 and 8.2 decreases, comparing with DCANet$_{base}$ on the datasets.

5) Evaluation of IDK on Larger Domain Gaps: To further evaluate the robustness of IDK strategy on different domains, we select a synthetic dataset GCC [18] which contains up to $15,212$ images. In order to make training more efficient, we sample the GCC-small dataset from GCC dataset. Specifically, we randomly select $1,489$ images from the whole dataset, and split them into two parts: $1,191$ training images and 298 testing images. We first train and evaluate DCANet$_{base}$ (single) on GCC-small dataset. Then the training images of GCC-small are merged into the multi-domain dataset, and train DCANet$_{base}$, DCANet ($\mathcal{L}_D$) and DCANet ($\mathcal{L}_I$), as illustrated in Sec. III-C. It is noted that the new multi-domain dataset consists of 300 ShanghaiTech A images, 400 ShanghaiTech

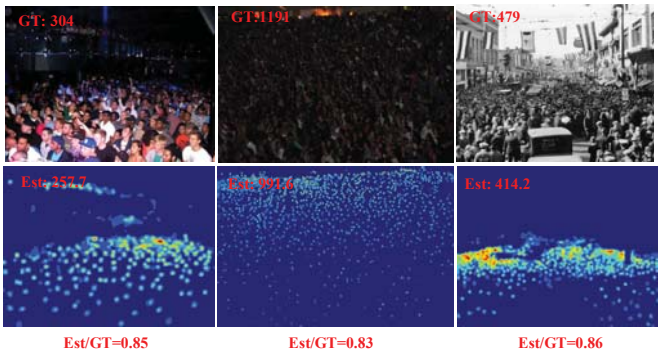


Fig. 9. Failure cases: samples with top-3 descending ΔP values.

B images, $1,201$ QNRF images and $1,191$ synthetic images. In this case, there ar larger domain gaps among these four datasets, since ShanghaiTech A prefers congested scenes, ShanghaiTech B images are sparse and are street views, UCF-QNRF images tend to be highly-congested and with complicated backgrounds, and GCC-small images are synthetic.

In Table XI, it is seen that DCANet$_{base}$ shows inferior performance to DCANet$_{base}$ (single) over all the datasets. Such performance drop is attributed to the large portion of synthetic images and large domain gaps between the datasets. This indicates that it is not trivial to obtain satisfying result for multiple domains by directly training on the multi-domain data. However, DCANet ($\mathcal{L}_D$) and DCANet ($\mathcal{L}_I$) outperform DCANet$_{base}$ consistently. Specifically, DCANet ($\mathcal{L}_I$) delivers the best performance with 60.1, 8.2, 93.0, 54.8 MAE on ShanghaiTech A, ShanghaiTech B, UCF-QNRF and GCC-small, with the decreases of 9.2%, 35.9%, 15.8%, 13.8% MAE on these datasets. Such observations indicate the effectiveness of our proposed training strategies.

6) Failure Cases: For each image I, we define a new metric named as improvement rate $\Delta P = \frac{|\hat{P}_{IDK} - P^{gt}|}{|\hat{P}_{base} - P^{gt}| + \epsilon}$, where P^{gt} is the ground-truth count, $\hat{P}_{base}$ denotes the predicted count by DCANet$_{base}$, $\hat{P}_{IDK}$ represents the estimated count by DCANet ($\mathcal{L}_I$). The parameter is set as $\epsilon = 1e^{-3}$. We rank the ΔP in the descending order and select the top-3 images as failure cases. Fig. 9 shows the results. It is seen that most of the images are in low illumination or grey images, which apparently act as "outliers" of the data distribution. Our method can also give satisfying estimated density maps and counts for these failure cases, which indicates the effectiveness of our method.

V. Conclusion

In this paper, we proposed a novel domain-guided channel attention network (DCANet) for multi-domain crowd counting, where dataset-level domain kernel (DDK) supervision and image-level domain kernel (IDK) supervision are proposed. Our DCANet is a very concise framework for multi-domain crowd counting. DDK strategy aims at predicting channel attention in view of the overall domain representation (*i.e.*, DDK), which boosts the performance in multiple domains. To improve the robustness of DCANet, IDK strategy further extends static DDK to adaptive IDK and encourages DCANet to predict image-specific channel attention. Experiments show that both DDK and IDK strategies improve the performance in multiple domains, while IDK further boosts the generalization and robustness of DCANet in unseen domains.

References

[1] X. Liu, W. Liu, T. Mei, and H. Ma, "Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 645–658, 2017.

[2] G. Wang, B. Li, Y. Zhang, and J. Yang, "Background modeling and referencing for moving cameras-captured surveillance video coding in hevc," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 2921–2934, 2018.

[3] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[4] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network," in *2018 IEEE Winter Conference on Applications of Computer Vision*, 2018, pp. 1113–1121.

[5] U. Sajid, H. Sajid, H. Wang, and G. Wang, "Zoomcount: A zooming mechanism for crowd counting in static images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3499–3512, 2020.

[6] M. Zhao, C. Zhang, J. Zhang, F. Porikli, B. Ni, and W. Zhang, "Scale-aware crowd counting via depth-embedded convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3651–3662, 2019.

[7] Q. Song, C. Wang, Y. Wang, Y. Tai, C. Wang, J. Li, J. Wu, and J. Ma, "To choose or to fuse? scale selection for crowd counting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 2576–2583.

[8] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang, and Y. Pang, "Attention scaling for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4706–4715.

[9] A. Luo, F. Yang, X. Li, D. Nie, Z. Jiao, S. Zhou, and H. Cheng, "Hybrid graph neural networks for crowd counting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11 693–11 700.

[10] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 532–546.

[11] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

[12] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.

[13] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1774–1783.

[14] M. A. Hossain, M. Kumar, M. Hosseinzadeh, O. Chanda, and Y. Wang, "One-shot scene-specific crowd counting." in *British Machine Vision Conference*, 2019, p. 217.

[15] M. K. K. Reddy, M. Hossain, M. Rochan, and Y. Wang, "Few-shot scene adaptive crowd counting using meta-learning," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2814–2823.

[16] Q. Wang, T. Han, J. Gao, and Y. Yuan, "Neuron linear transformation: modeling the domain shift for crowd counting," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[17] W. Li, L. Yongbo, and X. Xiangyang, "Coda: Counting objects via scale-aware adversarial density adaption," in *IEEE International Conference on Multimedia and Expo*, 2019, pp. 193–198.

[18] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8198–8207.

[19] J. Gao, Y. Yuan, and Q. Wang, "Feature-aware adaptation and density alignment for crowd counting in video surveillance," *IEEE Transactions on Cybernetics*, 2020.

[20] M. Marsden, K. McGuinness, S. Little, C. E. Keogh, and N. E. O'Connor, "People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8070–8079.

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[22] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 2074–2082.

[23] D. Babu Sam, S. Surya, and R. Venkatesh Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5744–5752.

[24] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1879–1888.

[25] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5245–5254.

[26] Y. Hu, X. Jiang, X. Liu, B. Zhang, J. Han, X. Cao, and D. Doermann, "Nas-count: Counting-by-density with neural architecture search," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 747–766.

[27] Z.-Q. Cheng, J.-X. Li, Q. Dai, X. Wu, and A. G. Hauptmann, "Learning spatial awareness to improve crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6152–6161.

[28] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6142–6151.

[29] B. Wang, H. Liu, D. Samaras, and M. Hoai, "Distribution matching for crowd counting," in *Advances in Neural Information Processing Systems*, 2020, pp. 1595–1607.

[30] M. Shi, Z. Yang, C. Xu, and Q. Chen, "Revisiting perspective information for efficient crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7279–7288.

[31] Z. Yan, Y. Yuan, W. Zuo, X. Tan, Y. Wang, S. Wen, and E. Ding, "Perspective-guided convolution networks for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 952–961.

[32] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe, "Reverse perspective network for perspective-aware object counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4374–4383.

[33] Y. Yang, G. Li, D. Du, Q. Huang, and N. Sebe, "Embedding perspective analysis into multi-column convolutional neural network for crowd counting," *IEEE Transactions on Image Processing*, vol. 30, pp. 1395–1407, 2020.

[34] Z. Yan, R. Zhang, H. Zhang, Q. Zhang, and W. Zuo, "Crowd counting via perspective-guided fractional-dilation convolution," *IEEE Transactions on Multimedia*, 2021.

[35] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[37] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

[38] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.

[39] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychological Review*, vol. 97, no. 2, p. 285, 1990.

[40] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[41] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[42] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1249–1258.

[43] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems*, 2017, pp. 506–516.

[44] J. Xiao, S. Gu, and L. Zhang, "Multi-domain learning for accurate and few-shot color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3258–3267.

[45] P. Liu, X. Qiu, and X.-J. Huang, "Adversarial multi-task learning for text classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1–10.

[46] X. Chen and C. Cardie, "Multinomial adversarial networks for multi-domain text classification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1226–1240.

[47] L. Zhu, Z. Zhao, C. Lu, Y. Lin, Y. Peng, and T. Yao, "Dual path multi-scale fusion networks with attention for crowd counting," *arXiv preprint arXiv:1902.01115*, 2019.

[48] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6133–6142.

[49] V. K. Valloli and K. Mehta, "W-net: Reinforced u-net for density map estimation," *arXiv preprint arXiv:1903.11249*, 2019.

[50] F. Dai, H. Liu, Y. Ma, J. Cao, Q. Zhao, and Y. Zhang, "Dense scale network for crowd counting," *arXiv preprint arXiv:1906.09707*, 2019.

[51] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[52] W. Xu, D. Liang, Y. Zheng, and Z. Ma, "Dilated-scale-aware attention convnet for multi-class object counting," *arXiv preprint arXiv:2012.08149*, 2020.

[53] M. Wang, H. Cai, J. Zhou, and M. Gong, "Interlayer and intralayer scale aggregation for scale-invariant crowd counting," *Neurocomputing*, vol. 441, pp. 128–137, 2021.

[54] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[55] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[56] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Tech. Rep., 2006.

[57] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 833–841.

[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8026–8037.

[59] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001.

[60] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[61] J. Gao, Q. Wang, and X. Li, "Pcc net: Perspective crowd counting via spatial convolutional network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3486–3498, 2019.

[62] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3225–3234.

[63] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, and X. Bai, "Learn to scale: Generating multipolar normalized density maps for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8382–8390.

[64] L. Liu, H. Lu, H. Xiong, K. Xian, Z. Cao, and C. Shen, "Counting objects by blockwise classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3513–3527, 2019.

[65] S. Jiang, X. Lu, Y. Lei, and L. Liu, "Mask-aware networks for crowd counting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 3119–3129, 2019.

[66] L. Liu, H. Lu, H. Zou, H. Xiong, Z. Cao, and C. Shen, "Weighing counts: Sequential crowd counting by reinforcement learning," in *European Conference on Computer Vision*. Springer, 2020, pp. 164–181.
[67] S. Bai, Z. He, Y. Qiao, H. Hu, W. Wu, and J. Yan, "Adaptive dilated network with self-correction supervision for counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Zhaoyi Yan received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2021. His research interests include deep learning, classification and crowd counting.

Pengyu Li received the Master degree from the Beijing University of the Posts and Telecommunications, Beijing, China, in 2015. His research interests include face recognition, crowd counting, unsupervised/semi-supervised learning, and deep learning with limited computational resources.

Biao Wang received the Ph.D. degree in the Department of Electronic Engineering from Tsinghua University, Beijing, China, in 2013. He has published over 20 papers in top tier academic conferences and journals. His current research interest include image classification, object detection, action recognition and unsupervised learning.

Dongwei Ren received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2017. From 2018 to 2021, he was an Assistant Professor with the College of Intelligence and Computing, Tianjin University. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include computer vision and deep learning.

Wangmeng Zuo (M'09-SM'14) received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. He is currently a Professor in the School of Computer Science and Technology, Harbin Institute of Technology. His current research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published over 100 papers in top tier academic journals and conferences. According to the statistics by Google scholar, his publications have been cited more than 20,000 times in literature. He has served as an Associate Editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *IEEE Transactions on Image Processing*.