Active Crowd Counting with Limited Supervision

Zhen Zhao[1⋆], Miaojing Shi[2⋆], Xiaoxiao Zhao[1], and Li Li[1,3]

[1] College of Electronic and Information Engineering, Tongji University
[2] King's College London
[3] Institute of Intelligent Science and Technology, Tongji University
zhenzhao0917@gmail.com; miaojing.shi@kcl.ac.uk; lili@tongji.edu.cn

Abstract. To learn a reliable people counter from crowd images, head center annotations are normally required. Annotating head centers is however a laborious and tedious process in dense crowds. In this paper, we present an active learning framework which enables accurate crowd counting with limited supervision: given a small labeling budget, instead of randomly selecting images to annotate, we first introduce an active labeling strategy to annotate the most informative images in the dataset and learn the counting model upon them. The process is repeated such that in every cycle we select the samples that are diverse in crowd density and dissimilar to previous selections. In the last cycle when the labeling budget is met, the large amount of unlabeled data are also utilized: a distribution classifier is introduced to align the labeled data with unlabeled data; furthermore, we propose to mix up the distribution labels and latent representations of data in the network to particularly improve the distribution alignment in-between training samples. We follow the popular density estimation pipeline for crowd counting. Extensive experiments are conducted on standard benchmarks i.e. ShanghaiTech, UCF_CC_50, MAll, TRANCOS, and DCC. By annotating limited number of images (e.g. 10% of the dataset), our method reaches levels of performance not far from the state of the art which utilize full annotations of the dataset.

1 Introduction

The task of crowd counting in computer vision is to automatically count people numbers in images/videos. With the rapid growth of world's population, crowd gathering becomes more frequent than ever. To help with crowd control and public safety, accurate crowd counting is demanded.

Early methods count crowds via the detection of individuals [49, 2, 34]. They suffer from heavy occlusions in dense crowds. More importantly, learning such people detectors normally requires bounding box or instance mask annotations for individuals, which often makes it undesirable in large-scale applications. Modern methods mainly conduct crowd counting via density estimation [32, 60,

⋆ Authors contributed equally.

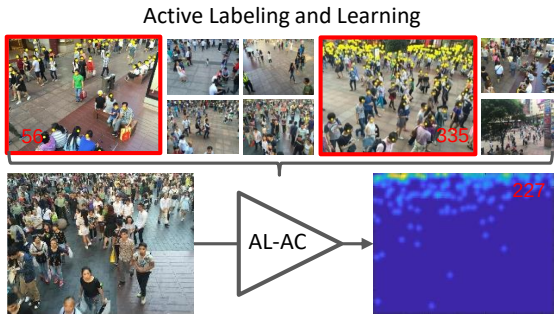

Fig. 1: Given a crowd counting dataset, we propose an active learning framework (AL-AC) which actively labels only a small proportion of the dataset and learns an accurate density estimation network using both labeled and unlabeled data.

44, 37, 26, 21, 20, 54]. Counting is realized by estimating a density map of an image whose integral over the image gives the total people count. Given a training image, its density map is obtained via Gaussian blurring at every head center. Head centers are the required annotations for training. Thanks to the powerful deep neural networks (DNNs) [17], density estimation based methods show a great success in recent progress [60, 39, 20, 35, 42, 54, 43, 25].

Despite above, annotating head centers in dense crowds is still a laborious and tedious process. For instance, it can take up to 10 minutes for our annotators to annotate a single image with 500 persons; while the popular counting dataset ShanghaiTech PartA [60] has 300 training images with an average of 501 persons per image! To substantially reduce the annotation cost, we study the crowd density estimation in a semi-supervised setting where only handful images are labeled while the rest are unlabeled. This setting has not been largely explored in crowd counting: [4, 61] propose to actively annotate the most informative video frames for semi-supervised crowd counting, yet the algorithms are not deep learning based and rely on frame consecutiveness. Recently, some deep learning works propose to leverage additional web data [24, 23] or synthetic data [51] for crowd counting; images in existing dataset are still assumed annotated, or at least many of them. The model transferability is also evaluated in some works [12, 54] where a network is trained on a source dataset with full annotations and tested on a target dataset with no/few annotations.

Given an existing dataset and a power DNN, we find that 1) learning from only a small subset, the performance can vary a lot depending on the subset selection; 2) for the specific subset that covers diverse crowd densities, the performance can be quite good (see results in Sec. 4.2). This motivates us to study crowd counting with very limited annotations yet producing very competitive precision. To achieve this goal, we propose an Active Learning framework for Accurate crowd Counting (AL-AC) as illustrated in Fig. 1: given a labeling budget, instead of randomly selecting images to annotate, we first introduce an active labelling strategy to iteratively annotate the most informative images in the

dataset and learn the counting model on them. In each cycle we select samples that cover different crowd densities and also dissimilar to previous selections. Eventually, the large amount of unlabeled data are also included into the network training: we design a classifier with gradient reversal layer [7] to align the intrinsic distributions of labeled and unlabeled data. Since all training samples contain the same object class, e.g. person, we propose to further align distributions in-between training samples by mixing up the latent representations and distribution labels among labeled and unlabeled data in the network. With very limited labeled data, our model produces very competitive counting result.

To summarize, several new elements are offered:

- We introduce an active learning framework for accurate crowd counting with limited supervision.
- We propose a partition-based sample selection with weights (PSSW) strategy to actively select and annotate both diverse and dissimilar samples for network training.
- We design a distribution alignment branch with latent MixUp to align the distribution between the labeled data and large amount of unlabeled data in the network.

Extensive experiments are conducted on standard counting benchmarks, i.e. ShanghaiTech [60], UCF_CC_50 [13], Mall [5],TRANCOS [9], and DCC [28]. Results demonstrate that, with a small number of labeled data, our AL-AC reaches levels of performance not far from state of the art fully-supervised methods.

2 Related works

In this section, we mainly survey deep learning based crowd counting methods and discuss semi-supervised learning and active learning in crowd counting.

2.1 Crowd counting

The prevailed crowd counting solution is to estimate a density map of a crowd image, whose integral of the density map gives the total person count of that image [60]. A density map encodes spatial information of an image, regressing it in a DNN is demonstrated to be more robust than simply regressing a global crowd count [58, 26]. Due to the commonly occurred heavy occlusions and perspective distortions in crowd images, multi-scale or multi-resolution architectures are often exploited in DNNs: Ranjan et al. [35] propose an iterative crowd counting network which produces the low-resolution density map and uses it to generate the high-resolution density map. Cao et al. [3] propose a novel encoder-decoder network, where the encoder extracts multi-scale features with scale aggregation modules and the decoder generates high-resolution density maps by using a set of transposed convolutions. Furthermore, Jiang et al. [15] develop a trellis encoder-decoder network that incorporates multiple decoding paths to hierarchically aggregate features at different encoding stages. In order to better utilize

multi-scale features in the network, the attention [21, 43], context [44, 22], or perspective [42, 55] information in crowd images is often leveraged into the network. Our work is a density estimation based approach.

2.2 Semi-supervised learning

Semi-supervised learning [29] refers to learning with a small amount of labeled data and a large amount of unlabeled data, and has been a popular paradigm in deep learning [52, 36, 18, 57]. It is traditionally studied for classification, where a label represents a class per image [19, 10, 36, 18]. In this work, we focus on semi-supervised learning in crowd counting, where the label of an image means the people count, with individual head points available in most cases. The common semi-supervised crowd counting solution is to leverage both labeled and unlabeled data into the learning procedure: Tan et al. [46] propose a semi-supervised elastic net regression method by utilizing sequential information between unlabeled samples and their temporally neighboring samples as a regularization term; Loy et al. [4] further improve it by utilizing both the spatial and temporal regularization in a semi-supervised kernel ridge regression problem; finally, in [61], graph Laplacian regularization and spatiotemporal constraints are incorporated into the semi-supervised regression. All these are not deep learning works and rely on temporal information among video frames.

Recently, Olmschenk et al. [30, 31] employ a generative adversarial network (GAN) in DNN to allow the usage of unlabeled data in crowd counting. Sam et al. [38] introduce an almost unsupervised learning method that only a tiny proportion of model parameters is trained with labeled data while vast parameters are trained with unlabeled data. Liu et al. [24, 23] propose to learn from unlabeled crowd data via a self-supervised ranking loss in the network. In [24, 23], they mainly assume the existence of a labeled dataset and add extra data from the web; in contrast, our AL-AC seeks a solution for accurate crowd counting with limited labeled data. Our method is also similar to [30, 31] in spirit of the distribution alignment between labeled and unlabeled data. While in [30, 31] they need to generate fake images to learn the discriminator in GAN which makes it hard to learn and converge. Our AL-AC instead mixes representations of labeled and unlabeled data in the network and learns the discriminator against them.

2.3 Active learning

Active learning defines a strategy determining data samples that, when added to the training set, improve a previously trained model most effectively [40]. Although it is not possible to obtain an universally good active learning strategy [6], there exist many heuristics [41], which have been proved to be effective in practice. Active learning has been explored in many applications such as image classification [45, 16] and object detection [8], while in this paper we focus on crowd counting. Methods in this context normally assumes the availability of the whole counting set and choose samples from it, which is the so-called pool-based

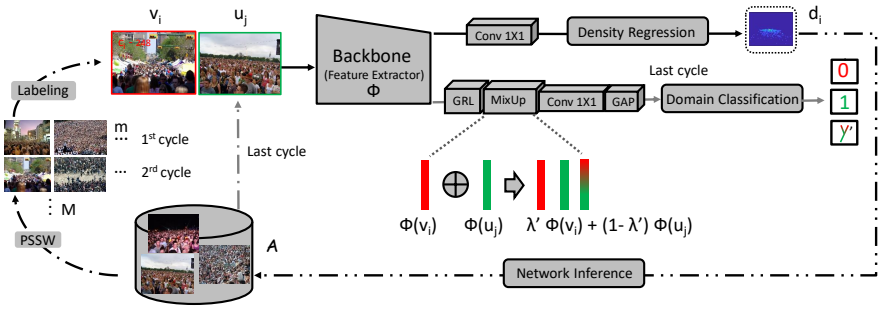

Fig. 2: Overview of our active learning framework for accurate crowd counting (AL-AC). GRL: gradient reversal layer; GAP: global average pooling. PSSW: Partition-based sample selection with weights; Conv 1×1: output channel is 1.

active learning [56]. [4] and [61] employ the graph-based approach to build adjacency matrix of all crowd images in the pool, sample selection is therefore cast as a matrix partitioning problem. Our work is also pool-based active learning.

Lately, Liu et al. [23] apply active learning in DNN where they measure the informativeness of unlabeled samples via mistakes made by the network on a self-supervised proxy task. The method is conducted iteratively and in each cycle it selects a group of images based their uncertainties to the model. The diversity of selected images is however not carefully taken care in their uncertainty measure, which might result in a biased selection within some specific count range. Our work instead interprets uncertainty from two perspectives: selected samples are diverse in crowd density and dissimilar to previous selection in each learning cycle. It should also be noted that [23] mainly focuses on adding extra unlabeled data to an existing labeled dataset, while our AL-AC seeks for the limited data to be labeled within a given dataset.

3 Method

3.1 Problem

We follow *crowd density estimation* in deep learning context where density maps are pixel-wise regressed in a DNN [60, 20]. A ground truth density map is generated by convolving Gaussian kernels at head centers in an image [60]. The network is optimized through a loss function minimizing the prediction error over the ground truth. In this paper, we place our problem in a *semi-supervised* setting where we only label several or few dozens of images while the rest large amount remains unlabeled. Both the labeled and unlabeled data will be exploited in model learning. Below, we introduce our active learning framework for accurate crowd counting (AL-AC).

3.2 Overview

Our algorithm follows an active learning pipeline in general. It is an iterative process where a model is learnt in each cycle and a set of samples is chosen to be labeled from a pool of unlabeled samples [41]. In classic setting, only one single sample is chosen in each cycle. This is however not feasible for DNNs because it is infeasible to train as many models as the number of samples since many practical problems of interest are very large-scale [40]. Hence, the commonly used strategy is batch mode selection [50, 23] where a subset is selected and labeled in each cycle. This subset is added into the labeled set to update the model and repeat the selection in next cycle. The procedure continues until a predefined criterion is met, e.g. a fixed budget.

Our method is illustrated in Fig. 2: given a dataset $\mathcal{A}$ with labeling budget M (number of images as in [38, 23]), we start by labeling m samples uniformly at random from $\mathcal{A}$. For each labeled sample v_i, we generate its count label c_i and density map d_i based on the annotated head points in v_i. We denote $\mathcal{V}^1 = \{v_i, c_i, d_i\}$ and $\mathcal{U}^1 = \{u_j\}$ as the labeled and unlabeled set in cycle 1, respectively. A DNN regressor R^1 is trained on $\mathcal{V}^1$ for crowd density estimation. Based on R^1's estimation of density maps on $\mathcal{U}^1$, we propose a partition-based sample selection with weights strategy to select and annotate m samples from $\mathcal{U}^1$. These samples are added to $\mathcal{V}^1$ so we have the updated labeled and unlabeled set $\mathcal{V}^2$ and $\mathcal{U}^2$ in 2^{rd} cycle. Model R^1 is further trained on $\mathcal{V}^2$ and updated as R^2. The prediction of R^2 is better than R^1 as it uses more labeled data, we use the new prediction on $\mathcal{U}^2$ to again select m samples and add them to $\mathcal{V}^2$. The process moves on until the labeling budget M is met. The unlabeled set $\mathcal{U}$ is also employed in network training through our proposed distribution alignment with latent MixUp. We only use $\mathcal{U}$ ($\mathcal{U}^T$) in the last learning cycle T as we observe that adding it in every cycle does not bring us accumulative benefits but rather additional training cost.

The backbone network is not specified in Fig. 2 as it can be any standard backbone. We will detail our selection of backbone, M, m and R in Sec. 4. Below we introduce our partition-based sample selection with weights and distribution alignment with latent MixUp. Overall loss function is given in this end.

3.3 Partition-based sample selection with weights (PSSW)

In each learning cycle, we want to annotate the most informative/uncertain samples and add them to the network. The *informativeness/uncertainty* of samples is evaluated from two perspectives: *diverse* in density and *dissimilar* to previous selections. It is observed that crowd data often forms a well structured manifold where different crowd densities normally distribute smoothly within the manifold space [4]; the diversity is to select crowd samples that cover different crowd densities in the manifold. This is realized by separating the unlabeled set into different density partitions for diverse selection. Within each partition, we want to select those samples that are dissimilar to previous labeled samples, such that the model has not seen them. The dissimilarity is measured considering both local

crowd density and global crowd count: we introduce a grid-based dissimilarity measure (GDSIM) for this purpose. Below, we formulate our partition-based sample selection with weights.

Formally, given the model R^t, unlabeled set $\mathcal{U}^t$ and labeled set $\mathcal{V}^t$ in t^{th} cycle, we denote by $\widetilde{c}_j$ the predicted crowd count by R^t for an unlabeled image u_j. The histogram of all $\widetilde{c}_j$ on $\mathcal{U}^t$ discloses the overall density distribution. For the sake of diversity, we want to partition the histogram into m parts and select one sample from each. Since the crowd counts are not evenly distributed (see Fig. 3: Left), sampling images evenly from the histogram can end up with a biased view of the original distribution. We therefore employ the Jenks natural breaks optimization [14] to partition the histogram. Jenks minimizes the variation within each range, so the partitions between ranges reflect the natural breaks of the histogram (Fig. 3).

Within each partition P_k, inspired by grid average mean absolute error (GAME) [9], we propose a grid-based dissimilarity from an unlabeled sample to labeled samples. Given an image i, GAME is originally introduced as an evaluation measure for density estimation,

$$\text{GAME}(L) = \sum_{l=1}^{4^L} |\widetilde{c_i^l} - c_i^l|, \tag{1}$$

where $\widetilde{c_i^l}$ is the estimated count in region l of image i. It can be obtained via the integration over the density $\widetilde{d_i^l}$ of that region l; c_i^l is the corresponding ground truth count. Given a specific level L, GAME(L) subdivides the image using a grid of 4^L non-overlaping regions which cover the full image (Fig. 3); the difference between the prediction and ground truth is the sum of the mean absolute error (MAE) in each of these regions. With different L, GAME indeed offers moderate ways to compute the dissimilarity between two density maps, taking care of both global counts and local details. Building on GAME, we introduce grid-based dissimilarity measure GDSIM as,

$$\underset{u_j \in \mathcal{P}_k}{\text{GDSIM}}(u_j, L_A) = \min_{i, v_i \in \mathcal{P}_k} \left(\sum_{L=0}^{L_\text{A}} \sum_{l=1}^{4^L} |\widetilde{c_j^l} - c_i^l| \right), \tag{2}$$

where u_j and v_i are from the unlabeled set $\mathcal{U}^t$ and labeled set $\mathcal{V}^t$, respectively; they both fall into the $\mathcal{P}_k$-th partition. $\widetilde{c_i^l}$ and c_i^l are crowd counts in region l as in formula (1) but for different images u_j and v_i (see Fig. 3: Right). Given the level L_A, unlike GAME, we compute the dissimilarity between u_j and v_i by traversing all levels from 0 to L_A (Fig. 3). In this way, the dissimilarity is computed based on both global count ($L = 0$) and local density ($L = L_A$) differences. Afterwards, instead of averaging the dissimilarity scores from u_j to all the v_i in $\mathcal{P}_k$, we use min to indicate if u_j is closer to any one of the labeled images, it is regarded as a familiar sample to the model. Ideally, we should choose the most dissimilar sample from each partition; nevertheless, the crowd count $\widetilde{c_j^l}$ in formula (2) is not

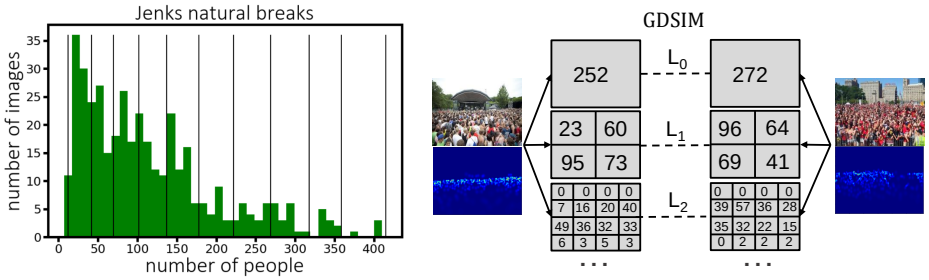

Fig. 3: Illustration of Jenks natural breaks (Left) and grid-based dissimilarity measure (GDSIM, Right). We take the histogram of crowd count on SHB.

ground truth. We convert the GDSIM scores to probabilities and adopt weighted random selection to label one sample from each partition.

3.4 Distribution alignment with latent MixUp

Since labeled data only represents partial crowd manifold, particularly when they are limited, distribution alignment with large amount of unlabeled data becomes necessary even within the same domain. In order for the model to learn a proper subspace representation of the entire set, we introduce distribution alignment with latent MixUp.

We assign labeled data with distribution labels 0 while unlabeled data with labels 1. A distribution classifier branched off from the deep extractor (ϕ in Fig. 2) is designed: it is composed of a gradient reversal layer (GRL) [7], 1×1 convolution layer and global average pooling (GAP) layer. The GRL multiplies the gradient by a certain negative constant (-1 in this paper) during the network back propagation; it enforces that the feature distributions over the labeled and unlabeled data are made as indistinguishable as possible for the distribution classifier, thus aligning them together.

The hard distribution labels create hard boundaries between labeled and unlabeled data. To further merge the distributions and particularly align in-between training samples, we adapt the idea from MixUp [59]. MixUp normally trains a model on random convex combinations of raw inputs and their corresponding labels. It encourages the model to behave linearly "between" training samples, as this linear behavior reduces the amount of undesirable oscillations when predicting outside the training samples. It has been popularly employed in several semi-supervised classification works [1, 47, 48, 59]. In this work, we integrate it into our distribution alignment branch for semi-supervised crowd counting. We find that mixing raw input images does not work for our problem. Instead we propose to mix their latent representations in the network: supposedly we have two images, x_1, x_2, and their distribution labels y_1, y_2, respectively. The latent representations of x_1 and x_2 are produced by the deep extractor ϕ as two tensors ($\phi(x_1)$ and $\phi(x_2)$) from the last convolutional layer of the backbone.

We mix up $(\phi(x_1), y_1)$, $(\phi(x_2), y_2)$ with a weight λ' as

$$\begin{aligned} z' &= \lambda'\phi(x_1) + (1-\lambda')\phi(x_2) \\ y' &= \lambda' \times y_1 + (1-\lambda') \times y_2. \end{aligned} \tag{3}$$

where (z', y') denotes the mixed latent representation and label. λ' is generated in the same way with [1]: $\lambda' = max(\lambda, 1-\lambda)$, $\lambda \sim \text{Beta}(\alpha, \alpha)$; α is a hyper-parameter set to 0.5. Both labeled and unlabeled data can be mixed. For two samples with the same label, their mixed label remains. We balance the number of labeled and unlabeled data with data augmentation (see Sec. 4.1) so a mixed pair can be composed of labeled or unlabeled data with (almost) the same probability. MixUp enriches the distribution in-between training samples. Together with GRL, it allows the network to elaborately knit the distributions of labeled and unlabeled data. The alignment is only carried out in the last active learning cycle as an efficient practice. The network training proceeds with a multi-task optimization that minimizes the density regression loss on labeled data and the distribution classification loss for all data including mixed ones, specified below.

3.5 Loss function

For density regression, we adopt the commonly used pixel-wise MSE loss $\mathcal{L}_{reg}$:

$$\mathcal{L}_{reg} = \frac{1}{2K}\sum_{k=1}^{K} \|d_k^e - d_k^g\|_2^2 \tag{4}$$

d_k^e and d_k^g denote the density map prediction and ground truth of image k, respectively. K is the number of labeled images. For the distribution classification, since distribution labels for mixed samples can be non-integers, we adopt the binary cross entropy with logits loss $\mathcal{L}_{dc}$, which combines a Sigmoid layer with the binary cross entropy loss. Given an image pair, $\mathcal{L}_{dc}$ is computed on each individual as well as their mixed representations (see Fig. 2). The overall multi-task loss function is given by

$$\mathcal{L} = \mathcal{L}_{reg} + \beta\mathcal{L}_{dc} \tag{5}$$

4 Experiments

We conduct our experiments on three counting datasets: ShanghaiTech [60], UCF_CC_50 [13], Mall [5]. In the supplementary material, we offer more results not only in the three datasets for people counting, but also in the TRANCOS [9] and DCC [28] datasets for vehicle and cell counting, respectively.

4.1 Experimental Setup

Datasets. *ShanghaiTech* [60] consists of 1,198 annotated images with a total of 330,165 people with head center annotations. This dataset is split into SHA and

SHB. The average crowd counts are 123.6 and 501.4, respectively. Following [60], we use 300 images for training and 182 images for testing in SHA; 400 images for training and 316 images for testing in SHB. *UCF_CC_50* [13] has 50 images with 63,974 head center annotations in total. The head counts range between 94 and 4,543 per image. The small dataset size and large variance make this a very challenging counting dataset. We call it UCF for short. Following [13], we perform 5-fold cross validations to report the average test performance. *Mall* [5] contains 2000 frames collected in a shopping mall. Each frame on average has only 31 persons. The first 800 frames are used as the training set and the rest 1200 frames as the test set.

Implementation details. The backbone (ϕ) design follows [20]: VGGnet with 10 convolutional and 6 dilated convolutional layers, it is pretrained on ILSVRC classification task. We follow the setting in [20] to generate ground truth density maps. To have a strong baseline, the training set is augmented by randomly cropping patches of 1/4 size of each image. We set a reference number 1200, both labeled and unlabeled data in each dataset are augmented up to this number to have a balanced distribution. For instance, if we have 30 labeled images, we need to crop 40 patches from each image to augment it to 1200. We feed the network with a minibatch of two image patches each time. In order to have the same size of two patches, we further crop them to keep the shorter width and height of the two. We set the learning rate as 1e-7, momentum 0.95 and weight decay 5e-4. We train 100 epochs with SGD optimizer for each active learning cycle and before the last cycle, the network is trained with only labeled data. In the last cycle, it is trained with both labeled and unlabeled data. In all experiments, L_A is 3 for GDSIM (2) and β is 3 for loss weight (5).

Evaluation protocol. We evaluate the counting performance via the commonly used mean absolute error (MAE) and mean square error (MSE) [39, 44, 21] which measures the difference between the counts of ground truth and estimation. For active learning, we choose to label around 10% images of the entire set, which goes along with our setting of limited supervision. m is chosen not too small so that we can normally reach the labeling budget in about 2-4 active learning cycles. Sec. 5 gives a discussion on the time complexity. M and m are by default 30/40 and 10 on SHA and SHB, 10 and 3 on UCF (initial number is 4), 80 and 20 on Mall, respectively. We also evaluate different M and m to show the effectiveness of our method. The baseline is to randomly label M images and train a regression model using the same backbone with our AL-AC but without distribution alignment. As in [4, 61], taken the randomness into account, we repeat each experiment with 10 trials for both mean and standard deviation, to show the improvement of our method over baseline.

4.2 ShanghaiTech

Ablation study. The proposed partition-based sample selection with weights and distribution alignment with latent MixUp are ablated.

Labeling budget M ***and*** m**.** As mentioned in Sec. 4.1, we set $M = 30/40$ and $m = 10$ by default. Comparable experiments are offered in two ways. First,

Dataset	SHA		SHB	
Method	PSSW	RS	PSSW	RS
M= 10, m=10	121.2± 9.3	121.2± 9.3	20.5± 4.8	20.5± 4.8
M=20, m=10	96.7± 7.3	111.5± 7.4	17.0± 1.9	19.3± 2.2
M=30, m=10	93.5± 2.9	102.1± 7.0	15.7± 1.5	19.9± 3.1
M=40, m=10	**85.4**± **2.5**	**93.8**± **5.6**	**14.6**± **1.3**	**17.9**± **1.9**
M = 30, m = 5	92.6 ± 3.1	102.1 ± 7.0	15.1 ± 1.5	19.9 ± 3.1
M = 40, m = 5	**84.4**± **2.6**	**93.8**± **5.6**	**14.4**± **1.2**	**17.9**± **1.9**

M=40, m=10	SHA	SHB
RS (Baseline)	93.8	17.9
Even Partition	89.6	16.2
Global Diff	86.6	15.3
PSSW	**84.4**	**14.4**

Table 1: Ablation study of the proposed partition-based sample selection with weights (PSSW) strategy. Left: comparison against random selection (RS). Right: comparison to some variants of PSSW; Even Partition means evenly splitting on the histogram of crowd count; Global Diff refers to using global count difference for dissimilarity. MAE is reported on SHA and SHB.

keeping $m = 10$, we vary M from 10 to 40. The results are shown in Table 1. We compare our partition-based sample selection with weights (PSSW) with random selection (RS); distribution alignment is not added in this experiment. For PSSW, its MAE on SHA is gradually decreased from 121.2 with $M = 10$ to 85.4 with $M = 40$, the standard deviation is also decreased from 9.3 to 2.5. The MAE result is in general 10 points lower than RS. With different M, PSSW also produces lower MAE than RS on SHB. For example, with $M = 40$, PSSW yields an MAE of 14.6 v.s. 17.9 for RS.

Second, by keeping $M = 30/40$, we decrease m from 10 to 5 and repeat the experiment. Results show that having a small m indeed works slightly better: for instance, PSSW with $M = 30$ and $m = 5$ reduces MAE by 1.0 on SHA compared to PSSW with $M = 30$ and $m = 10$. On the other hand, m can not be too small as discussed in Sec. 3.2 and Sec. 5. In practice, we still keep $m = 10$ for both efficiency and effectiveness.

Variants of PSSW. Our PSSW has two components: the Jenks-based partition for diversity, and the GDSIM for dissimilarity (Sec. 3). In order to show the effectiveness of each, we present two variants of PSSW: Even Partition and Global Diff. Even Partition means that Jenks-based partition is replaced by evenly splitting the ranges on the histogram of crowd count while GSDIM remains; Global Diff means that GDSIM is replaced by using the global count difference to measure the dissimilarity while Jenks-based partition remains. We report MAE on SHA and SHB in Table 1: Right. It can be seen that Even Partition produces MAE 89.6 on SHA and 16.2 on SHB, while Global Diff produces 86.6 and 15.3. Both are clearly inferior to PSSW (84.4 and 14.4). This suggests the importance of the proposed diversity and dissimilarity measure.

Distribution alignment with latent MixUp. Our proposed distribution alignment with latent MixUp is composed of two elements: distribution classifer with GRL and latent MixUp (Sec. 3.4). To demonstrate their effectiveness, we present the result of PSSW plus GRL classifer (denoted as PSSW + GRL), and latent MixUp (denoted as PSSW + GRL + MX) in Table 2. We take $M = 40$ as an example, adding GRL and MX to PSSW contributes to 5.0 points

Dataset	SHA		SHB	
M = 30, m =10	MAE	MSE	MAE	MSE
PSSW	93.5± 2.9	151.0± 15.1	15.7± 1.5	28.3± 3.4
PSSW+GRL	90.8± 2.7	144.9± 14.5	14.7± 1.3	27.8± 2.9
PSSW+GRL+MX	**87.9**± **2.3**	**139.5**± **12.7**	**13.9**± **1.2**	**26.2**± **2.5**
M = 40, m =10	MAE	MSE	MAE	MSE
PSSW	85.4± 2.5	144.7± 10.7	14.6± 1.3	24.6± 3.0
PSSW+GRL	82.7± 2.4	140.9± 11.3	13.7± 1.3	23.5± 2.2
PSSW+GRL+MX	**80.4** ± **2.4**	**138.8**± **10.1**	**12.7**± **1.1**	**20.4**± **2.1**

M=40, m=10	SHA	SHB
RS (Baseline)	93.8	17.9
RS+GRL+MX	87.3	15.1
PSSW	84.4	14.4
PSSW+GRL+MX	**80.4**	**12.7**

Table 2: Ablation study of the proposed distribution alignment with latent MixUp. Left: analysis on latent MixUp (MX) and gradient reversal layer (GRL). Right: comparison against RS plus GRL and MX. MAE is reported in the right table.

MAE decrease on SHA and 1.9 points decrease on SHB. Specifically, The MX contributes to 2.3 and 1.0 points decrease on SHA and SHB, respectively. The same observation goes for MSE: by adding GRL and MX, it decreases from 144.7 to 138.8 on SHA, from 24.6 to 20.4 on SHB.

To make a further comparison, we also add the proposed distribution alignment with latent MixUp to RS in Table 2: Right, where we achieve MAE 87.3 on SHA and 15.1 on SHB. Adding GRL+MX to RS also improves the baseline: the performance difference between PSSW and RS becomes smaller; yet, the absolute value of the difference is still big, which justifies our PSSW. Notice PSSW + GRL + MX is the final version of our AL-AC hereafter.

Comparison with fully-supervised methods. We compare our work with those prior arts [60, 39, 20, 35, 42, 43, 27]. All these approaches are fully-supervised methods which utilize annotations of the entire dataset (300 in SHA and 400 in SHB). While in our setting, we label only 30/40 images, 10% of the entire set. It can be seen that our method outperforms the representative methods [60, 39] a few years ago, and are not far from other recent arts, i.e. [20, 35, 42, 43, 27]. A direct comparison to ours is CSRNet [20], we share the same backbone. With about 10% labeled data, our AL-AC retains 85% accuracy on SHA (68.2 / 80.4), 83% accuracy on SHB (10.6 / 12.7). Compared to our baseline (denoted as RS in Table 1), AL-AC in general produces significantly lower MAE, e.g. 87.9 v.s. 102.1 on SHA with $M = 30$; 17.9 v.s. 12.7 on SHB with $M = 40$.

Despite that we only label 10% data, our distribution alignment with latent MixUp indeed enables us to make use of more unlabeled data across datasets: for instance, a simple implementation with M = 40 on SHA, if we add SHB as unlabeled data to AL-AC for distribution alignment, we obtain an even lower MAE 78.6 v.s. 80.4 in Table 3.

Comparison with semi-supervised methods. There are also some semi-supervised crowd counting methods [23, 38, 31][1]. For instance in [38, 31], with $M = 50$ they produce MAE 170.0 and 136.9 on SHA, respectively. These are much higher MAE than ours. Since [38, 31] use different architectures from AL-

[1] Results of [23, 38] can be estimated from their curve plots.

Dataset	SHA		SHB	
Measures	MAE	MSE	MAE	MSE
MCNN [60]	110.2	173.2	26.4	41.3
Switching CNN [39]	90.4	135.0	21.6	33.4
CSRNet [20]	68.2	115.0	10.6	16.0
ic-CNN [35]	68.5	116.2	10.7	16.0
PACNN [42]	**62.4**	102.0	7.6	11.8
CFF [43]	65.2	109.4	**7.2**	11.2
BAYESIAN+ [27]	62.8	**101.8**	7.7	**12.7**
Baseline (M = 30)	102.1	164.0	19.9	30.6
AL-AC (M = 30)	**87.9**	**139.5**	**13.9**	**26.2**
Baseline (M =40)	93.8	150.9	17.9	27.3
AL-AC (M =40)	**80.4**	**138.8**	**12.7**	**20.4**

Table 3: Comparison of AL-AC to the state of the art on SHA and SHB.

Counting	UCF	
Measures	MAE	MSE
MCNN [60]	377.6	509.1
Switching CNN [39]	318.1	439.2
CP-CNN[44]	295.8	320.9
CSRNet [20]	266.1	397.5
ic-CNN [35]	260.0	365.5
PACNN [42]	241.7	320.7
BAYESIAN+ [27]	**229.3**	**308.2**
Baseline (M=10, m=3)	444.7± 25.9	600.3± 32.7
AL-AC (M=10, m=3)	351.4± 19.2	448.1± 24.5
Baseline (M=20, m=10)	417.2± 29.8	550.1± 25.5
AL-AC (M=20, m=10)	**318.7**± 23.0	**421.6**± 24.1

Table 4: Comparison of AL-AC with state of the art on UCF.

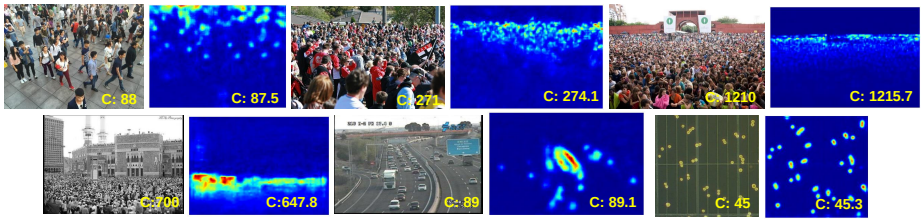

Fig. 4: Examples of AL-AC on SHA, SHB, UCF, TRANCOS, and DCC. Ground truth counts are in the original images while predicted counts in the estimated density maps.

AC, they are not straightforward comparisons. For [23], it uses about 50% labeled data on SHA (Fig.7 in [23]) to reach the similar performance of our AL-AC with 10% labeled data. We both adopt the VGGnet yet [23] utilizes extra web data for ranking loss while we only use unlabeled data within SHA, we use dilated convolutions while [23] does not. To make them more comparable, we instead use the same backbone of [23] and repeat AL-AC on SHA (implementation details still follow Sec. 4.1), the mean MAE with M=30, m=10 on SHA becomes 91.4 (v.s. 87.9 in Table 3), which is still much better than that of [23].

In the supplementary material, we also provide the result by gradually increasing M till 280 on SHA, where we show that by labelling about 80-100 labeled data (nearly 30% of the dataset), AL-AC already reaches the performance close to the fully-supervised method, as in [20] (Table 3).

4.3 UCF_CC_50

It has 40 training images in total. We show in Table 4 that, labeling ten of them ($M = 10, m = 3$) already produces a very competitive result: the MAE is 351.4 while the MSE is 448.1. The MAE and MSE are significantly lower (93.3 and

Mall	Baseline	AL-AC*	Count Forest [33]	ConvLSTM [53]	DecideNet [21]	E3D [62]	SAAN [11]
MAE	5.9± 0.9	3.8± 0.5	4.4	2.1	1.5	1.6	1.3
MSE	6.3± 1.1	5.4± 0.8	2.4	7.6	1.9	2.1	1.7

Table 5: Comparison of AL-AC with state of the art on Mall (M=80, m=20).

152.2 points) than baseline. We analyzed the result and found that our AL-AC is able to select those hard samples with thousands of persons and label them for training, while this is not guaranteed in random selection. Compared to fully supervised method, e.g. [20], our MAE is not far. We also present the result of $M = 20, m = 10$: MAE/MSE is further reduced.

4.4 Mall

Different from ShanghaiTech and UCF datasets, Mall contains images with much sparser crowds, 31 persons on average per image. Following our setup, we label 80 out of 800 images and compare our AL-AC with both baseline and other fully-supervised methods [33, 53, 21, 62, 11] in Table 5. With 10% labeled data, we achieve MAE 3.8 superior to the baseline and [33], MSE 5.4 superior to the baseline and [53]. This shows the effectiveness of our method on sparse crowds.

5 Discussion

We present an active learning framework for accurate crowd counting with limited supervision. Given a counting dataset, instead of annotating every image, we introduce a partition-based sample selection with weights to label only a few most informative images and learn a crowd regression network upon them. This process is iterated till the labeling budget is reached. Next, rather than learning from only labeled data, the abundant unlabeled data are also exploited: we introduce a distribution alignment branch with latent MixUp in the network. Experiments conducted on standard benchmarks show that labeling only 10% of the entire set, our method already performs close to recent state-of-the-art.

By choosing an appropriate m, we normally reach the labeling budget in three active learning cycles. In our setting, training data in each dataset are augmented to a fixed number. We run our experiments with GPU GTX1080. It takes around three hours to complete each active learning cycle. The total training hours are more or less the same to fully-supervised training, as in each learning cycle we train much fewer epochs with limited number of labeled data. More importantly, compared to the annotation cost for an entire dataset (see Sec. 1 for an estimation on SHA), ours is substantially reduced !

Acknowledgement: This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61828602 and 51475334; as well as National Key Research and Development Program of Science and Technology of China under Grant No. 2018YFB1305304, Shanghai Science and Technology Pilot Project under Grant No. 19511132100.

References

1. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249 (2019)
2. Brostow, G.J., Cipolla, R.: Unsupervised bayesian detection of independent motion in crowds. In: CVPR (2006)
3. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale aggregation network for accurate and efficient crowd counting. In: ECCV (2018)
4. Change Loy, C., Gong, S., Xiang, T.: From semi-supervised to transfer counting of crowds. In: CVPR (2013)
5. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: BMVC (2012)
6. Dasgupta, S.: Analysis of a greedy active learning strategy. In: NIPS (2005)
7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: JMLR (2015)
8. Gonzalez-Garcia, A., Vezhnevets, A., Ferrari, V.: An active search strategy for efficient object class detection. In: CVPR (2015)
9. Guerrero-Gómez-Olmedo, R., Torre-Jiménez, B., López-Sastre, R., Maldonado-Bascón, S., Onoro-Rubio, D.: Extremely overlapping vehicle counting. In: Iberian Conference on Pattern Recognition and Image Analysis (2015)
10. Hoffer, E., Ailon, N.: Semi-supervised deep learning by metric embedding. arXiv preprint arXiv:1611.01449 (2016)
11. Hossain, M., Hosseinzadeh, M., Chanda, O., Wang, Y.: Crowd counting using scale-aware attention networks. In: WACV (2019)
12. Hossain, M.A., Kumar, M., Hosseinzadeh, M., Chanda, O., Wang, Y.: One-shot scene-specific crowd counting. In: BMVC (2019)
13. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: CVPR (2013)
14. Jenks, G.F.: The data model concept in statistical mapping. International yearbook of cartography **7**, 186–190 (1967)
15. Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., Shao, L.: Crowd counting and density estimation by trellis encoder-decoder networks. In: CVPR (2019)
16. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: CVPR (2009)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
18. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: ICLR (2016)
19. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: ICMLW (2013)
20. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: CVPR (2018)
21. Liu, J., Gao, C., Meng, D., G. Hauptmann, A.: Decidenet: Counting varying density crowds through attention guided detection and density estimation. In: CVPR (2018)
22. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: CVPR (2019)
23. Liu, X., Van De Weijer, J., Bagdanov, A.D.: Exploiting unlabeled data in cnns by self-supervised learning to rank. IEEE transactions on pattern analysis and machine intelligence (2019)

24. Liu, X., Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. In: CVPR (2018)
25. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: CVPR (2019)
26. Lu, Z., Shi, M., Chen, Q.: Crowd counting via scale-adaptive convolutional neural network. In: WACV (2018)
27. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: ICCV (2019)
28. Marsden, M., McGuinness, K., Little, S., Keogh, C.E., O'Connor, N.E.: People, penguins and petri dishes: adapting object counting models to new visual domains and object types without forgetting. In: CVPR (2018)
29. Olivier, C., Bernhard, S., Alexander, Z.: Semi-supervised learning. In: IEEE Transactions on Neural Networks, vol. 20, pp. 542–542 (2006)
30. Olmschenk, G., Tang, H., Zhu, Z.: Crowd counting with minimal data using generative adversarial networks for multiple target regression. In: WACV (2018)
31. Olmschenk, G., Zhu, Z., Tang, H.: Generalizing semi-supervised generative adversarial networks to regression using feature contrasting. Computer Vision and Image Understanding (2019)
32. Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: ECCV (2016)
33. Pham, V.Q., Kozakaya, T., Yamaguchi, O., Okada, R.: Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In: ICCV (2015)
34. Rabaud, V., Belongie, S.: Counting crowded moving objects. In: CVPR (2006)
35. Ranjan, V., Le, H., Hoai, M.: Iterative crowd counting. In: ECCV (2018)
36. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: NIPS (2015)
37. Sam, D.B., Babu, R.V.: Top-down feedback for crowd counting convolutional neural network. In: AAAI (2018)
38. Sam, D.B., Sajjan, N.N., Maurya, H., Babu, R.V.: Almost unsupervised learning for dense crowd counting. In: AAAI (2019)
39. Sam, D.B., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. In: CVPR (2017)
40. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In: ICLR (2018)
41. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
42. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. In: CVPR (2019)
43. Shi, Z., Mettes, P., Snoek, C.G.: Counting with focus for free. In: ICCV (2019)
44. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid cnns. In: ICCV (2017)
45. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: ICCV (2019)
46. Tan, B., Zhang, J., Wang, L.: Semi-supervised elastic net for pedestrian counting. Pattern Recognition **44**(10-11), 2297–2304 (2011)
47. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: ICML (2019)
48. Verma, V., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825 (2019)

49. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. IJCV **63**(2), 153–161 (2003)
50. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology **27**(12), 2591–2600 (2016)
51. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: CVPR (2019)
52. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade, pp. 639–655. Springer (2012)
53. Xiong, F., Shi, X., Yeung, D.Y.: Spatiotemporal modeling for crowd counting in videos. In: ICCV (2017)
54. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to scale: Generating multipolar normalized density map for crowd counting. In: ICCV (2019)
55. Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-guided convolution networks for crowd counting. In: ICCV (2019)
56. Yang, Y., Ma, Z., Nie, F., Chang, X., Hauptmann, A.G.: Multi-class active learning by uncertainty sampling with diversity maximization. International Journal of Computer Vision **113**(2), 113–127 (2015)
57. Yang, Z., Shi, M., Avrithis, Y., Xu, C., Ferrari, V.: Training object detectors from few weakly-labeled and many unlabeled images. arXiv preprint arXiv:1912.00384 (2019)
58. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: CVPR (2015)
59. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. In: ICLR (2018)
60. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: CVPR (2016)
61. Zhou, Q., Zhang, J., Che, L., Shan, H., Wang, J.Z.: Crowd counting with limited labeling through submodular frame selection. IEEE Transactions on Intelligent Transportation Systems **20**(5), 1728–1738 (2018)
62. Zou, Z., Shao, H., Qu, X., Wei, W., Zhou, P.: Enhanced 3d convolutional networks for crowd counting. In: BMVC (2019)