A Generalized Loss Function for Crowd Counting and Localization

Jia Wan Ziquan Liu Antoni B. Chan
Department of Computer Science, City University of Hong Kong
jiawan1998@gmail.com, ziquanliu2-c@my.cityu.edu.hk, abchan@cityu.edu.hk

Abstract

Previous work [40] shows that a better density map representation can improve the performance of crowd counting. In this paper, we investigate learning the density map representation through an unbalanced optimal transport problem, and propose a generalized loss function to learn density maps for crowd counting and localization. We prove that pixel-wise L2 loss and Bayesian loss [29] are special cases and suboptimal solutions to our proposed loss function. A perspective-guided transport cost function is further proposed to better handle the perspective transformation in crowd images. Since the predicted density will be pushed toward annotation positions, the density map prediction will be sparse and can naturally be used for localization. Finally, the proposed loss outperforms other losses on four large-scale datasets for counting, and achieves the best localization performance on NWPU-Crowd and UCF-QNRF.

1. Introduction

Crowd counting and localization draw increasing attention recently because of its practical usage in surveillance, transport management and business. Most of the algorithms predict a density map from a crowd image, where the summation of the density map is the crowd count [41, 4]. A density map (a smooth heat map) is an *intermediate representation* of the crowd – one popular method to generate the ground-truth density map is to place a Gaussian kernel on each person's dot annotation. The density map estimator is then trained as a standard pixel-wise regression problem using L2 loss [12, 40] (see Fig. 1a). In contrast to pixel-wise L2 loss, Bayesian loss (BL) [29] generates an aggregated dot prediction from the density map prediction, and uses a point-wise loss function between the ground-truth dot annotations and the aggregated dot prediction (see Fig. 1b).

Both L2 and BL assume a fixed ground-truth representation, either Gaussian density kernels for L2 or Gaussian likelihoods for BL. Recent works [40, 43] have shown that the intermediate density map representation affects the counting performance, and a better density map representation can be learned in an end-to-end manner from the dot-

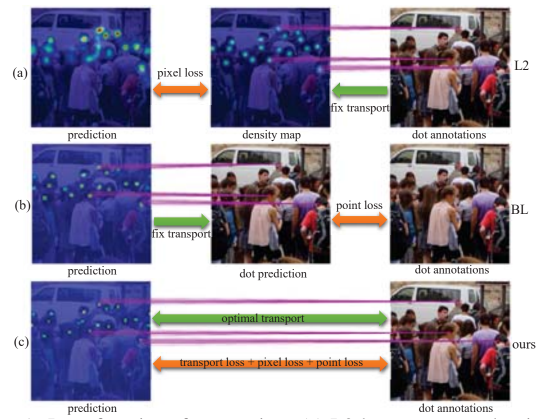


Figure 1: Loss functions for counting: (a) L2 loss generates density map as the supervision and uses a pixel-wise loss function. (b) Bayesian loss (BL) [29] computes an aggregated dot prediction and uses a point-wise loss function. We show that L2 and BL are related to an optimal transport problem using a suboptimal transport matrix. (c) Our proposed loss is based on unbalanced optimal transport, where the transport cost is fully minimized and both the pixel-wise and point-wise losses are considered.

annotations. However, [40, 43] still use L2 loss for training, which is not appropriate in suppressing background and improving localization. In particular, with L2 loss, a unit change in density in background regions (which is a large localization error) is equivalent to a unit change in density near a dot annotation (which is a small localization error). Thus the L2 loss function is not ideal for localization or generating compact density maps, and we should prefer a loss function that has an increased penalty for errors far from the annotations, so as to improve localization and compactness.

Considering both motivations of learning the density map representation and using localization-sensitive loss, we propose a generalized loss function based on an unbalanced optimal transport (OT) framework, which measures the transport cost between the predicted density map and the ground-truth dot annotations (see Fig. 1c). We show that the transport matrix, which is optimized to minimize the loss, is related to the intermediate density map representation. To better handle perspective changes in the image, we propose a perspective-guided transport cost function to better separate the density around people who are close together due to the camera perspective. The proposed loss function decomposes into four terms: 1) a transport loss that pushes the predicted density toward annotations; 2) a transport regularization term that prevents collapse onto a single annotation; 3) a pixel-wise loss that measures the difference between the predicted density map and the constructed density map (from the transport matrix); 4) a point-wise loss that ensures that all annotations are accounted for in the predicted density map. We further show that our proposed loss is a generalization of the traditional L2 loss with Gaussian density kernel and BL, i.e., they are special cases and suboptimal solutions to the unbalanced OT in our proposed loss.

Compared to previous losses, our proposed loss function has four advantages: 1) the density map representation is learned via the optimized transport matrix; 2) it does not require any special design for background regions (such as [29, 42]), and naturally pushes predicted density away from the background and towards the annotations; 3) it produces compact density maps that can be naturally used for localization; 4) it is less sensitive to the blur factor hyperparameter (which is equivalent to the Gaussian kernel variance). In summary, the contributions of the paper are four-fold:

1. We propose a generalized loss function, motivated by unbalanced optimal transport theory, for crowd counting and localization. We prove L2 and BL are special cases and suboptimal solutions of our loss function.
2. To handle perspective effects in crowd images, we propose a perspective-guided transport cost, which increases transport costs of density far from the camera, thus making densities in those regions more compact.
3. In extensive experiments on crowd counting, using our loss achieves better performance than traditional loss functions on three large-scale datasets, NWPU-Crowd, JHU-CROWD++, and UCF-QNRF.
4. Our low-resolution predicted density maps (1/8 image size) achieve the best localization performance on two large benchmarks NWPU-Crowd and UCF-QNRF.

2. Related Works

Traditional crowd counting Traditional methods count the number of people in an image by detecting human bodies [18] or body parts [20], which does not work well for images with high crowd density. Thus, direct regression methods are proposed based on low-level features [4, 5, 12].

Density map based counting Most of recent methods use a deep neural network (DNN) to predict density maps [42] from crowd images, where the sum over density map is the crowd count [19]. The DNN is trained using L2 pixel-wise loss. Various DNN structures are proposed to address scale variation [50, 33, 15], to refine the predicted density map [30, 31, 35], and to encode context information [36, 47]. To improve the generalization ability, [49] proposes a cross-scene crowd counting method. [46] proposes a synthetic dataset and a domain adaptation method to adapt DNNs trained from synthetic data to real images. [7] focuses on semantic consistency across different domains. Since the labeling of crowd images is time-consuming, semi-supervised and weakly-supervised methods are proposed. [26] proposes a ranking loss to utilize unlabeled data, while [38] proposes a Gaussian Process-based iterative model with limited labeled data. [28] proposes to learn generic features with self-training on surrogate tasks. Active learning is also used for crowd counting with limited supervision [51].

Loss functions Although most of the crowd counting methods use L2 norm as the loss function, L2 loss is sensitive to the choice of variance in the Gaussian kernel [40]. Therefore, Bayesian loss (BL) [29] is proposed with point-wise supervision. However, BL cannot well handle false positives in the background, and requires a special design for the background region. [41] proposes a generative model for spatial noise in dot annotations, and derives a novel loss function that is robust to annotation noise.

The most related work to ours is the concurrent work of DM-count [44], which considers density maps and dot maps as probability distributions, and uses *balanced* OT to match the shape of the two distributions. The DM-count loss is composed of three terms: the OT loss, a total variation (TV) pixel-wise loss, and a counting loss. There are four key differences between our work and DM-count. First, DM-count normalizes the density map predictions and the dot map to compute the *balanced* OT between them. Since normalization removes the actual count in the two maps, an additional counting loss is required to ensure that the count (i.e., the sum of the density map) is predicted correctly. However, this counting loss provides poor supervision, since its gradient adds the same constant value to all pixels (see Supp. A). In contrast, our proposed loss is based on *unbalanced* OT, which preserves the count of the prediction and GT dot annotations – any mismatch in counts is penalized by our pixel-wise and point-wise loss terms, which give direct pixel-wise supervision on the erroneous predictions. Second, the TV loss used in DM-count is a pixel-wise loss between the normalized density map prediction and the normalized *dot* map, which is prone to over-fitting especially for the localization task. In contrast, our work contains a pixel-wise loss between the predicted density map and optimized constructed density map (via the transport matrix), which is less prone to over-fit. Third, we show that our loss is a generalization of other loss functions (L2 and BL) when a sub-optimal fixed transport matrix is used. Fourth, DM-count uses the standard squared Euclidean distance as the transport cost, while our work uses a perspective-guided transport cost to increase the separation between people's density in crowded regions, which improves localization. We compare our loss with DM-count in the ablation study.

Crowd Localization To perform counting, density map estimation and localization simultaneously, [13] proposes a composition loss function. [23] proposes to localize crowd locations by a recurrent zooming network, while [22] proposes a detection-based method with RGB-D data. [32] propose to count, localize, and estimate head size simultaneously. [45] proposes a large-scale benchmark for crowd counting and localization. These works use pixel-wise losses, which are sensitive to the kernel bandwidth. In contrast, our loss pushes density towards annotation and is less sensitive to the bandwidth, and thus robust for localization.

3. A Generalized Loss Function for Crowd Counting and Localization

Recent work [40] shows that learning the intermediate density map representation yields improved performance in counting networks, which suggests the importance of directly using the ground-truth (GT) dot annotations for supervision, rather than a fixed GT density map. However, [40] uses L2 pixel-wise loss for training, which is not appropriate since small changes in background density (which are large localization errors) are equivalent to small changes in density over a person (which are small localization errors). Thus a loss function that penalizes the distance of the error to the annotation is preferred, as in optimal transport (OT) cost between the predicted density map and the dot annotations. Note that the predicted density map and GT dot annotations may not have the same count, due to mis-predictions, or annotation noise (missing/duplicate annotations). Considering these issues, we propose to use *unbalanced* optimal transport (UOT) as the loss function for training. This loss function both learns the density map representation (uses dot annotations as supervision), and handles count mismatches between the prediction and GT.

3.1. Generalized loss function

Formally, let the predicted density map be $\mathcal{A} = \{(a_i, \boldsymbol{x}_i)\}_{i=1}^n$, where a_i is the predicted density of pixel $\boldsymbol{x}_i \in \mathbb{R}^2$ and n is the number of pixels. We denote $\boldsymbol{a} = [a_i]_i$ as the predicted density map. The ground-truth dot map is $\mathcal{B} = \{(b_j, \boldsymbol{y}_j)\}_{j=1}^m$, where $\boldsymbol{y}_j$ is the location of the j-th annotation, m is the number of annotation points, and b_j is the number of people represented by the annotation. In this paper, we assume $\boldsymbol{b} = [b_j]_j = 1_m$.

Our loss function is based on the entropic-regularized unbalanced optimal transport cost,

$$\mathcal{L}_{\mathbf{C}}^{\tau}(\mathcal{A}, \mathcal{B}) = \min_{\mathbf{P} \in \mathbb{R}_+^{n \times m}} \langle \mathbf{C}, \mathbf{P} \rangle - \varepsilon H(\mathbf{P}) + \tau D_1(\mathbf{P}1_m | \boldsymbol{a}) + \tau D_2(\mathbf{P}^\top 1_n | \boldsymbol{b}). \quad (1)$$

$\mathbf{C} \in \mathbb{R}_+^{n \times m}$ is the transport cost matrix, whose entry C_{ij} measures the cost of moving the predicted density at $\boldsymbol{x}_i$ to GT dot annotation $\boldsymbol{y}_i$ via the cost function $C_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j)$. $\mathbf{P}$ is the transport matrix, which (fractionally) assigns each each location $\boldsymbol{x}_i$ from $\mathcal{A}$ to $\boldsymbol{y}_i$ from $\mathcal{B}$ for measuring the cost. The optimal transport cost is obtained by minimizing the loss over $\mathbf{P}$. Note that $\hat{\boldsymbol{a}} = \mathbf{P}1_m$ is the construction of a intermediate density map representation from the GT annotations, while $\hat{\boldsymbol{b}} = \mathbf{P}^\top 1_n$ is the reconstruction of the GT dot annotations. See Fig. 2 (top) for an example.

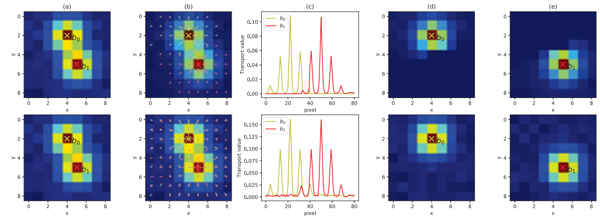


Figure 2: The relationship between density map, annotations, and transport matrix: (top) optimal transport; (bottom) fixed transport from Bayesian loss [29]. (a) predicted density maps with 81 pixels and 2 annotations; (b) transport matrix $\mathbf{P} \in \mathbb{R}^{81 \times 2}$, where the arrow length represents the transport value, and the direction points to the assigned annotation; (c) columns of $\mathbf{P}$; (d,e) the transport plan for annotation b_0 and b_1, reshaped to a map, equivalent to the ground-truth density map for each annotation. The fixed transport (bottom) assigns false positives in the background to annotations, while the optimal transport (top) only considers nearby density. The optimal transport is more sparse, which is better for localization.

The loss function decomposes into four terms. The first term $\langle \mathbf{C}, \mathbf{P} \rangle$ is the transport loss, which encourages prediction of density values near the annotations; it pushes the predicted density towards the annotation during training. The second term $H(\mathbf{P}) = -\sum_{ij} P_{ij} \log P_{ij}$ is the entropic regularization term, which favors partial transports between locations, resulting in spread-out (less compact) density maps. Larger values of ε will yield less compact predicted density maps, and vice-versa. The third term $D_1(\mathbf{P}1_m|\boldsymbol{a})$ is the *pixel-wise* loss between the predicted density map $\boldsymbol{a}$ and the constructed intermediate density map representation $\hat{\boldsymbol{a}} = \mathbf{P}1_m$, i.e., the "ground-truth" density map. Finally, the fourth term $D_2(\mathbf{P}^\top 1_n|\boldsymbol{b})$ is the *point-wise* loss between the reconstructed annotations $\hat{\boldsymbol{b}} = \mathbf{P}^\top 1_n$ and the GT annotations. The last two terms are complementary – the pixel-wise term D_1 ensures that all predicted density values have a corresponding annotation, while the point-wise term D_2 ensures that all GT annotations are accounted for (used in the transport plan). In other words, any predicted density that is not associated with an annotation is penalized, and any annotation that is not used is penalized.

In our implementation, we use squared L2 norm for the pixel-wise term and L1-norm for the point-wise term,

$$D_1(\mathbf{P}1_m|\boldsymbol{a}) = \|\mathbf{P}1_m - \boldsymbol{a}\|_2^2, \quad (2)$$

$$D_2(\mathbf{P}^\top 1_n|\boldsymbol{b}) = \|\mathbf{P}^\top 1_n - \boldsymbol{b}\|_1. \quad (3)$$

In Sec. 4, we show that L2 and BL are suboptimal solutions to our proposed generalized loss function, which use a fixed intermediate representation (i.e., transport matrix).

3.2. Perspective-Guided Transport Cost

We next propose a transport cost function for crowd counting. A typical cost function is the squared Euclidean distance between the two points, $L_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2$, which considers all distances equally throughout the image. However, due to the perspective effect in crowd images, people that are farther from the camera will appear closer together in the image, while those closer to the camera will be farther apart in the image. In order to keep the density of people in the "far" crowds from leaking together, the transport costs for those regions in the image should be higher, which will make the density for those people more compact.

To encode perspective information in crowd images, a perspective-guided cost function is proposed to have larger penalty for the transport of density far from the camera. Formally, the cost function is defined as:

$$C_{ij} = \exp\left(\frac{1}{\eta(\boldsymbol{x}_i, \boldsymbol{y}_j)} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2\right), \quad (4)$$

where $\eta(\boldsymbol{x}_i, \boldsymbol{y}_j)$ is an adaptive perspective factor, which is mapped between an interval based on the average height $\frac{1}{2}(h_{\boldsymbol{x}_i} + h_{\boldsymbol{y}_j})$, where $h_{\boldsymbol{x}} \in [0, 1]$ refers to normalized height of the pixel $\boldsymbol{x}$ in the image. We use the exponential in (4) to enhance the cost of moving densities over long distances, which makes the predicted density maps more compact.

3.3. Optimization of transport matrix

As shown in [1], the solution to (1) for the optimization of transport matrix $\mathbf{P}$ is unique, and has the form

$$\mathbf{P} = \mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v}), \quad \mathbf{K} = \exp(-\mathbf{C}/\varepsilon), \quad (5)$$

where $\mathbf{K}$ is the Gibbs kernel constructed from the cost matrix $\mathbf{C}$, and $\exp$ is element-wise exponential. For the optimization of $\mathbf{P}$ in (1), we approximate D_1 and D_2 with KL divergence, since this yields an efficient algorithm.[1] The $\mathbf{u}, \mathbf{v}$ are computed with the generalized Sinkhorn iterations,

$$\mathbf{u}^{(\ell+1)} = \left(\frac{\boldsymbol{a}}{\mathbf{K}\mathbf{v}^{(\ell)}}\right)^{\frac{\tau}{\tau+\epsilon}}, \quad \mathbf{v}^{(\ell+1)} = \left(\frac{\boldsymbol{b}}{\mathbf{K}^\top\mathbf{u}^{(\ell+1)}}\right)^{\frac{\tau}{\tau+\epsilon}}, \quad (6)$$

where the division and exponent operations are element-wise. To compute network gradients, the optimal $\mathbf{v}^*$ is considered as a constant, and $\mathbf{u}^*$ is a function of $\boldsymbol{a}$, i.e., $\mathbf{P} = \mathrm{diag}(\mathbf{u}^*(\boldsymbol{a}))\mathbf{K}\mathrm{diag}(\mathbf{v}^*)$.

3.4. Density map counting and localization

To apply our loss function to density map counting, we learn a density map estimator $f(\mathcal{I})$, whose input is the image $\mathcal{I}$, and output is the density map vector $\boldsymbol{a}$. The predicted density map $\mathcal{A}$, together with the corresponding GT annotations $\mathcal{B}$, are fed into the loss function in (1). To compute the loss, the transport matrix $\mathbf{P}$ is optimized for each input separately using (5) and the iterations in (6). Given the test image, the density map estimator predicts the density map, which is then summed to obtain the count.

We perform localization by applying simple post-processing to the predicted density map $\boldsymbol{a}$. First, $\boldsymbol{a}$ is upsampled to the image size since the density map is $\frac{1}{8}$ of the input image size (due to pooling operations). Then, a pixel is considered a candidate for a predicted location if its value is the local maximum in a 3×3 window centered on the pixel. Finally, the candidates with density larger than 0.05 are the final location predictions.

[1]Solving for $\mathbf{P}$ using D_1 and D_2 in (2) and (3) requires an inefficient nested optimization.

4. Relationship with traditional losses

In this section, we prove that the traditional L2 loss and Bayesian loss (BL) [29] are suboptimal solutions to the unbalanced OT in our loss function in (1). In particular, L2 and BL are both 2-stage approximations to solve (1), consisting of: 1) constructing a half-iteration approximate solution of the transport matrix $\mathbf{P}$ using entropic-regularized *balanced* OT with squared Euclidean transport cost; 2) substituting the approximate $\mathbf{P}$ into our loss in (1). Because the computed $\mathbf{P}$ is a half-iteration approximation to the minimization in (1), both L2 and BL are suboptimal approximations of our loss function.

4.1. Half-iteration approximations for $\mathbf{P}$

We first derive closed-form solutions to approximate $\mathbf{P}$ under the entropic-regularized *balanced* OT problem. Removing the last two terms in (1), we obtain the entropic regularized OT problem,

$$\mathcal{L}^\varepsilon_{\mathbf{C}}(\mathcal{A}, \mathcal{B}) = \min_{\mathbf{P}\in\mathbb{R}^{n\times m}_+} \langle \mathbf{C}, \mathbf{P}\rangle - \varepsilon H(\mathbf{P}), \quad (7)$$

where $\mathbf{C}$, $\mathcal{A}$, $\mathcal{B}$ are defined as before. As shown in [1], the solution to (7) is unique with $\mathbf{P} = [P_{ij}]_{ij}$,

$$P_{ij} = u_i K_{ij} v_j, \quad K_{ij} = \exp(-C_{ij}/\varepsilon), \quad (8)$$

where $\mathbf{u} = [u_i]_i, \mathbf{v} = [v_j]_j$ are from the Sinkhorn iterations,

$$\mathbf{u}^{(\ell+1)} = \frac{\boldsymbol{a}}{\mathbf{K}\mathbf{v}^{(\ell)}}, \quad \mathbf{v}^{(\ell+1)} = \frac{\boldsymbol{b}}{\mathbf{K}^\top\mathbf{u}^{(\ell+1)}}, \quad (9)$$

where the division is element-wise. Typically, the iterations are initialized with $\mathbf{v} = 1_m$.

We next obtain 2 approximate solutions to $\mathbf{P}$, by substituting a half Sinkhorn iteration, $\mathbf{u}^{(\ell+1)}$ or $\mathbf{v}^{(\ell+1)}$, into (8),

$$\hat{P}_{ij} = \left[\frac{\boldsymbol{a}}{\mathbf{K}\mathbf{v}}\right]_i K_{ij} v_j, \quad \tilde{P}_{ij} = u_i K_{ij} \left[\frac{\boldsymbol{b}}{\mathbf{K}^\top\mathbf{u}}\right]_j. \quad (10)$$

If $\mathbf{v}$ is uniform (as in the typical initialization) and the cost function C_{ij} is the squared Euclidean distance, then

$$\hat{P}_{ij} = \frac{K_{ij}}{\sum_{j=1}^m K_{ij}} a_i = \hat{\pi}_{ij} a_i, \quad \hat{\pi}_{ij} = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2/\varepsilon)}{\sum_{j=1}^m \exp(-\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2/\varepsilon)}.$$

Similarly, assuming $\mathbf{u}$ is initialized as uniform and C_{ij} is the squared Euclidean distance, then for $\tilde{P}_{ij}$ we have

$$\tilde{P}_{ij} = \frac{K_{ij}}{\sum_{i=1}^n K_{ij}} b_j = \tilde{\pi}_{ij}, \quad \tilde{\pi}_{ij} = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2/\varepsilon)}{\sum_{i=1}^n \exp(-\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2/\varepsilon)},$$

since $b_j = 1$. Note that the difference between $\hat{\pi}_{ij}$ and $\tilde{\pi}_{ij}$ is the summation in the denominator is either over GT annotation locations $\boldsymbol{y}_j$ or density map pixels $\boldsymbol{x}_i$, respectively.

4.2. Relationship with L2 Loss

We now derive the L2 loss as a special case of our loss function when using the suboptimal transport matrix $\tilde{\mathbf{P}}$. Substituting into (1), we note that the first 2 terms, $\langle \mathbf{C}, \tilde{\mathbf{P}}\rangle$ and $H(\tilde{\mathbf{P}})$ are constants w.r.t. $\boldsymbol{a}$, and thus do not affect the loss in terms of $\boldsymbol{a}$. Next, it is straightforward to show that $\tilde{\mathbf{P}}^\top 1_n = 1_m$, and therefore the fourth term in (1) is $D_2(\tilde{\mathbf{P}}^\top 1_n|\boldsymbol{b}) = 0$. Only the third term (i.e, the pixel-wise loss) remains, and assuming $\tau = 1$, we have the loss

$$\tilde{\mathcal{L}}(\mathcal{A}, \mathcal{B}) = D_1(\tilde{\mathbf{P}}1_m|\boldsymbol{a}) = \sum_{i=1}^{n}(a_i - \tilde{a}_i)^2, \tag{11}$$

$$\tilde{a}_i = [\tilde{\mathbf{P}}1_m]_i = \sum_{j=1}^{m}\tilde{\pi}_{ij} = \sum_{j=1}^{m}\frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2/\varepsilon)}{\sum_{i=1}^{n}\exp(-\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2/\varepsilon)}. \tag{12}$$

Note that $\tilde{a}_i$ is equivalent to a "ground-truth" density map value at pixel location i, which places a Gaussian kernel with squared-bandwidth $\epsilon/2$ at each annotation $\boldsymbol{y}_j$. The denominators in (12) are the normalization constants of each Gaussian. Thus from (11) and (12), our loss using the approximate transport matrix $\tilde{\mathbf{P}}$ is equivalent to L2 loss with traditional Gaussian-based density maps for supervision.

4.3. Relationship with Bayesian Loss

We next derive BL [29] as a special case of our loss using approximation $\hat{\mathbf{P}}$. Note that $\hat{\pi}_{ij}$ is the probability of assigning the density value of the i-th pixel to the j-th annotation point, as defined in [29]. Since $[\hat{\mathbf{P}}1_m]_i = \sum_{j=1}^{m}\hat{\pi}_{ij}a_i = a_i$, then the third term in (1) is $D_1(\hat{\mathbf{P}}1_m|\boldsymbol{a}) = 0$. Assuming that ϵ is small (so that the entropy term can be ignored) and $\tau = 1$, we have the loss

$$\hat{\mathcal{L}}(\mathcal{A}, \mathcal{B}) = \langle \mathbf{C}, \hat{\mathbf{P}} \rangle + D_2(\hat{\mathbf{P}}^\top 1_n|\boldsymbol{b}) \tag{13}$$

$$= \sum_{i=1}^{n}\omega_i a_i + \sum_{j=1}^{m}|1 - \sum_{i=1}^{n}\hat{\pi}_{ij}a_i|, \tag{14}$$

where $\omega_i = \sum_{j=1}^{m}C_{ij}\hat{\pi}_{ij} = \sum_j ||\boldsymbol{x}_i - \boldsymbol{y}_j||^2\hat{\pi}_{ij}$ is a weight on the prediction a_i. The second term in (14) is exactly the point-wise Bayesian loss defined in [29]. The first term in (14) can be interpreted as a background loss, which penalizes non-zero values of a_i for pixels $\boldsymbol{x}_i$ far from any annotation (i.e., false positives). In particular, the weight ω_i on the i-th pixel is based on a weighted average of squared distances from the pixel to the annotations.

We now relate the background term in (14) with the background model used in BL [29]. The background loss in [29] is based on the nearest annotation to each point $\boldsymbol{x}_i$ (details in Supp. B),

$$\mathcal{L}_{BG} = |0 - \sum_{i=1}^{n}\bar{\omega}_i a_i| = \sum_{i=1}^{n}\bar{\omega}_i a_i, \tag{15}$$

where the weight $\bar{\omega}_i = \frac{\bar{k}_i}{\bar{k}_i + \sum_{j=1}^{m}K_{ij}}$ and $\bar{k}_i = \exp(-(d - ||\boldsymbol{x}_i - \boldsymbol{y}_{\eta(i)}||)^2/\varepsilon)$, and $\eta(i)$ is the index of the annotation nearest to $\boldsymbol{x}_i$. The weight can be rewritten as $\bar{\omega}_i = \exp(\frac{2d}{\varepsilon}||\boldsymbol{x}_i - \boldsymbol{y}_{\eta(i)}|| - \frac{d^2}{\varepsilon})\bar{\pi}_i$, where $\bar{\pi}_i = \frac{K_{i,\eta(i)}}{\bar{k}_i + \sum_{j=1}^{m}K_{ij}}$ is the weight contribution for the nearest neighbor $\eta(i)$. Note that (15) and the first term in (14) have the same form, but use different weight values $\bar{\omega}_i$ or ω_i. For BL, the weight $\bar{\omega}_i$ is based on the exponential distance to the nearest annotation $\eta(i)$. In contrast, for the background term in (14), the weight ω_i is based on a weighted average of squared distances to all annotations. Therefore, the background model used in [29] is a special case of the background term in (14). Fig. 3 shows a visualization of the weights maps for $\bar{\omega}_i$ and ω_i. Both BL and (14) have large weights for background regions and small weights for head (annotation) regions. However, the weight map for (14) is smoother since all annotations are considered using squared distances, while the weight map for BL contains flat regions since it uses the exponential distance to only the nearest annotation.

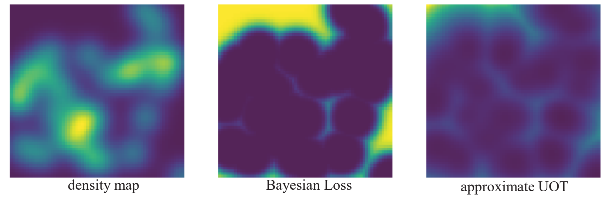


Figure 3: Comparison of background weight maps for Bayesian Loss ($\bar{\omega}$) and approximate UOT (ω) in (14).

Thus, from (14), BL with background model is a special case of our proposed loss in (1), where the approximate transport matrix is $\hat{\mathbf{P}}$ and a single-neighbor approximation of the cost matrix $\mathbf{C}$ is used to compute the cost term (i.e.,. the background loss). If no background model is used, then the cost matrix is assumed to be 0.

5. Experiments

In this section, we evaluate the counting and localization performance using the proposed general loss function.

5.1. Experimental setups

Datasets: We evaluate the performance of the proposed loss function on four datasets: ShanghaiTech [50], UCF-QNRF [13], JHU-CROWD++ [39], and NWPU-Crowd [45]. ShanghaiTech contains two parts: Part A (482/300 for training/testing) and Part B (716/400 for training/testing). UCF-QNRF is a large-scale dataset consists of 1,535 high-resolution crowd images (1,201/334 for training/testing). JHU-CROWD++ contains 4,317 images (2,722/500/1,600 images are for training/validation/testing). NWPU-Crowd is the largest dataset with 3,109 training images, 500 validation images, and 1,500 testing images (whose labels are not release to public for fair comparison). We report results on the NWPU-Crowd test set.

Evaluation metrics: Mean absolute error (MAE) and root mean squared error (MSE) are used as the evaluation metric for counting performance, as in previous works [16]:

$$\text{MAE} = \frac{1}{N}\sum_i |y_i - \hat{y}_i|, \ \text{MSE} = (\frac{1}{N}\sum_i (y_i - \hat{y}_i)^2)^{1/2},$$

where $y_i, \hat{y}_i$ are the GT and predicted counts. To evaluate the localization performance, we followed the protocols used in NWPU-Crowd and UCF-QNRF, respectively. For

Table 1: Test results comparing different loss functions with different backbones on UCF-QNRF.

	VGG19 [29]		CSRNet [21]		MCNN [50]	
	MAE	MSE	MAE	MSE	MAE	MSE
L2	98.7	176.1	110.6	190.1	186.4	283.6
BL [29]	88.8	154.8	107.5	184.3	190.6	272.3
NoiseCC [41]	85.8	150.6	96.5	**163.3**	177.4	259.0
DM-count [44]	85.6	148.3	103.6	180.6	176.1	263.3
Ours	**84.3**	**147.5**	**92.0**	165.7	**142.8**	**227.9**

NWPU-Crowd, Precision, Recall and F-measure are used, and Precision, Recall and AUC are used for UCF-QNRF.

Backbone and training: Following the experiment settings in [41], we use 3 backbone networks: VGG19 [29], CSRNet [21], and MCNN [50]. We train the counting network using our loss function in (1), where $\mathbf{P}$ is solved using the generalized Sinkhorn iterations in (6). We set $\varepsilon = 0.005$. In practice, ε-scaling heuristic is used for acceleration, which needs less than 20 iterations until converge, and the computation is calculated in log-domain for numerical stability as in [6]. In preliminary studies using the exponential transport cost, we observe that $\eta \in \{0.6, 0.8\}$ yield better performance (Fig. 4a). Thus for the perspective-guided cost, we simply map the range of image pixel y-coordinates to $\eta \in [0.6, 0.8]$. VGG19 and CSRNet are pre-trained on ImageNet, and MCNN is trained from scratch. Adam optimizer [17] is used to train the networks with learning rate 10^{-5} for VGG19/CSRNet, and 10^{-4} for MCNN.

5.2. Ablation studies

We first conduct ablation studies on our loss function on UCF-QNRF or JHU-CROWD++.

5.2.1 Comparison with different losses

In Table 1, we compare the performance of loss functions with different backbones, including L2 pixel-wise loss, Bayesian loss (BL) [29], NoiseCC [44], which models noisy annotations, and DM-count [44], which uses balanced OT as part of their loss. Our proposed loss function achieves the lowest MAE among all loss functions. Our loss function outperforms L2 and BL since we use an optimal transport plan, instead of fixed as shown in Sec. 4, and a better transport plan (i.e., density map) can achieve better performance as shown in [43]. Compared to DM-count, our loss function achieves better performance especially for MCNN trained from scratch. Our loss function is based on unbalanced OT using exponential transport cost, while DM-count is based on balanced OT and squared-Euclidean cost. We further compare the unbalanced/balanced OT frameworks and transport cost functions in the next 2 ablation studies.

5.2.2 The effect of transport cost functions

We evaluate the effectiveness of the proposed perspective-guided transport cost by comparing with other cost functions, including Euclidean distance (L_{ij}), squared Euclidean (L_{ij}^2), and exponential of Euclidean ($e^{L_{ij}}$). The test results are shown in Fig. 4a. First, the standard cost function based on Euclidean distance is less effective than the exponential cost function. The perspective-guided cost achieves the best performance, which confirms that adapting the cost function to the perspective changes is effective for crowd counting. We visualize the density maps predicted with different cost functions in Fig. 6. Using exponential cost yields a more compact density map compared to the squared Euclidean cost. Furthermore, using perspective-guided cost yields more sparsity for high-density regions, which demonstrates that its efficacy at pushing away density from background to annotations.

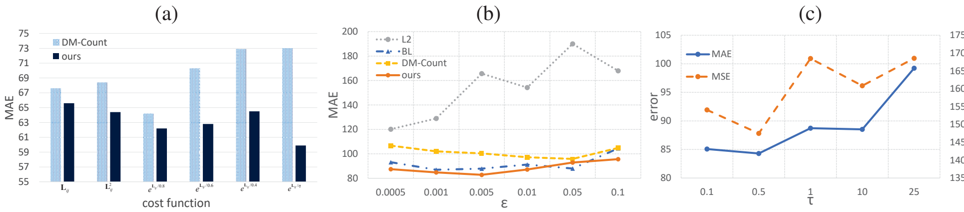
Finally, TV loss used in [44] assumes the same smoothness for all annotations, which is incompatible with the perspective-guided (PG) cost that produces different smoothness for each annotation. To confirm this, we conduct an experiment by decreasing the weight of TV loss by 10x, and the performance using PG cost improved to MAE 66, which is still worse than using exponential cost with fixed $\eta = 0.8$ (MAE 64). Thus, TV loss hinders the PG transport cost (with adaptive smoothness), but works with the exponential cost with fixed η (i.e., fixed smoothness).

5.2.3 The effect of unbalanced/balanced OT

We next compare our unbalanced OT framework with the balanced OT of DM-count [44], using the same transport cost functions in Sec. 5.2.2. As seen in Fig. 4a, our proposed loss outperforms DM-Count when using different cost functions, which demonstrates the efficacy of unbalanced OT for the density map regression problem. Our proposed loss is based on the unbalanced OT problem, where extra/missing density is penalized using both point-wise and pixel-wise losses. In contrast, DM-count uses balanced OT, and requires an additional count loss, which is a map-wise loss that is less effective (see discussion in Sec. 2 “loss functions”). Second, the TV loss in DM-Count uses the normalized *dot* map for pixel-wise supervision, which is prone to overfitting. Finally, our proposed cost is more effective at pushing away density from background to annotations compare to squared Euclidean cost (see Sec. 5.2.2).

5.2.4 The effect of ε

We next investigate the effect of blur factor ε, which is equivalent to the Gaussian squared-bandwidth (variance) used to generate the ground-truth density maps for L2 and BL, as shown in Sec. 4.2. The results for varying ε are shown in Fig. 4b. The L2 loss is sensitive to ε, with MAE increasing significantly as ε increases. In contrast, BL and our proposed loss are less sensitive, since the background model in BL and the transport loss in ours can push density towards annotations, always making the predicted density maps compact. The proposed loss function is generally better than BL because the BL background model only consid-



(d)

		point-wise			
		L1		L2	
		MAE	MSE	MAE	MSE
pixel-wise	L1	88.0	149.5	88.7	152.1
	L2	**84.3**	**147.5**	88.3	160.0

Figure 4: Ablation study: **(a)** Comparison of different transport cost functions on JHU-CROWD++, using our approach and DM-count [44]. The base cost is Euclidean distance $\mathbf{L}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2$. **(b)** The effect of ε for different loss functions. ε is the Gaussian bandwidth for density maps when using L2 and BL, or the blur factor for our unbalanced OT. **(c)** The effect of τ. **(d)** Comparison of divergences for point-wise and pixel-wise costs on UCF-QNRF.

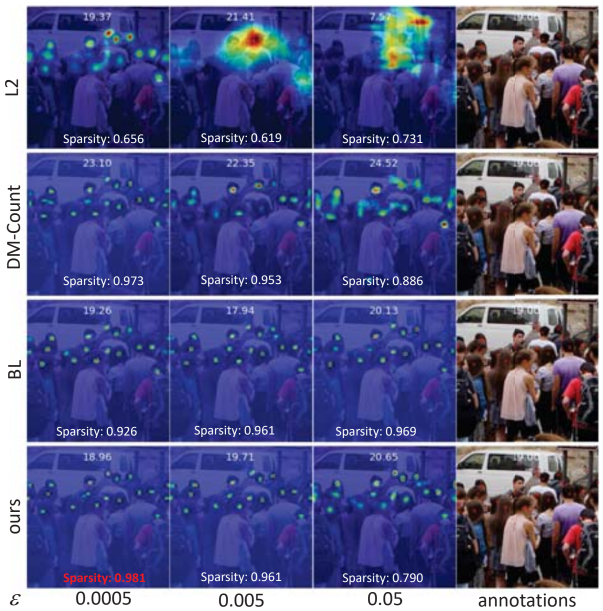


Figure 5: Visualization of density maps predicted from models trained with different loss functions and blur factors ε. Note that the width and height of the trained images patches are normalized to 1. The sparsity is defined as the percentage of pixels with density less than 0.001, and the most sparse density map is shown in red bold.

ers the nearest annotation, while our loss considers all annotations (see Sec. 4.3). A visualization is shown in Fig. 5. As the ε increases, the density map for L2 become more blurry and inaccurate, which demonstrates that L2 is sensitive to ε. BL and our loss are generally robust to the choice of ε, and the network can learn a sharper density map with the proposed loss function, which is better for localization.

5.2.5 The effect of τ and divergence function

Next, we study the effect of τ and the divergence function for point-wise and pixel-wise cost functions. We try different combinations of L1 and L2 norms with $\tau = 0.5$, and the results are presented in Fig. 4d. The best performance occurs with point-wise L1 and pixel-wise L2, which matches the common practice for the individual point-wise and pixel-wise-based losses [29, 42]. Next, using L1 and L2 for the point-wise and pixel-wise costs, we vary τ (see Fig. 4c), and visualize the learned density maps in Fig. 6 As τ decreases, the density maps becomes more compact (more sparse), since the transport cost dominates and

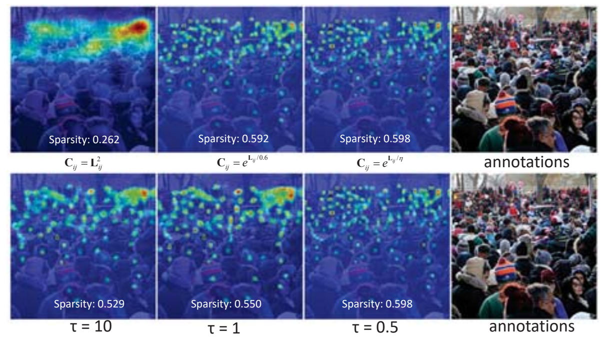


Figure 6: Visualization of the effect of (top) transport cost functions, and (bottom) τ. The sparsity is defined as the percentage of pixels with density less than 0.001.

Table 2: Effectiveness of terms in the loss function on UCF-QNRF.

Component	Combinations			
$\langle \mathbf{C}, \mathbf{P} \rangle$	✓	✓	✓	✓
$H(\mathbf{P})$		✓	✓	✓
$D_1(\mathbf{P}1_m \vert \mathbf{a})$	✓		✓	✓
$D_2(\mathbf{P}^\top 1_n \vert \mathbf{b})$	✓	✓		✓
MAE	91.1	85.4	85.0	84.3

pushes the density more towards the annotations.

5.2.6 The effect of terms in the loss function

Finally, we evaluate the effect of different terms in the proposed loss function in Table 2. The most important term is entropic regularization, which controls the smoothness of the prediction to prevent over-fitting. Unbalanced OT (UOT) with either pixel-wise loss (removing D_2) or point-wise loss (removing D_1) can be effective, and is better than the corresponding approximations BL and L2 loss (85.4 vs 88.8; 85.0 vs. 98.7), which shows the effectiveness of using a better transport solution. Finally, UOT with both pixel-wise and point-wise losses further improves the model.

5.3. Comparison with state-of-the-arts

To evaluate the overall counting performance, we compare VGG19 [29] trained with our loss function with state-of-the-art methods in Table 3. First, compared with the baseline method BL and DM-Count, our method achieves significantly better performance especially for the large-scale datasets NWPU-Crowd, JHU-CROWD++, and UCF-QNRF. Second, our model achieves the best MAE on the

Table 3: Comparison with state-of-the-art crowd counting methods.

		NWPU		JHU++		UCF-QNRF		ShTech A		ShTech B	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [50]	CVPR'16	232.5	714.6	188.9	483.4	277.0	426.0	110.2	173.2	26.4	41.3
SwitchCNN [33]	CVPR'17	-	-	-	-	228.0	445.0	90.4	135.0	21.6	33.4
CP-CNN [36]	ICCV'17	-	-	-	-	-	-	73.6	106.4	20.1	30.1
ACSCP [34]	CVPR'18	-	-	-	-	-	-	75.7	102.7	17.2	27.4
CSRNet [21]	CVPR'18	121.3	387.8	85.9	309.2	110.6	190.1	68.2	115.0	10.6	16.0
CL [13]	ECCV'18	-	-	-	-	132.0	191.0	-	-	-	-
SANet [3]	ECCV'18	190.6	491.4	91.1	320.4	-	-	67.0	104.5	8.4	13.6
DSSINet [25]	ICCV'19	-	-	133.5	416.5	99.1	159.2	60.6	96.0	6.8	10.3
MBTTBF [37]	ICCV'19	-	-	81.8	299.1	97.5	165.2	60.2	94.1	8.0	15.5
BL [29]	ICCV'19	105.4	454.2	75.0	299.9	88.7	154.8	62.8	101.8	7.7	12.7
LSCCNN [32]	TPAMI'20	-	-	112.7	454.4	120.5	218.2	66.5	101.8	7.7	12.7
KDMG [43]	TPAMI'20	100.5	415.5	69.7	268.3	99.5	173.0	63.8	99.2	7.8	12.7
RPNet [48]	CVPR'20	-	-	-	-	-	-	61.2	96.9	8.1	11.6
ASNet [14]	CVPR'20	-	-	-	-	91.6	159.7	57.8	**90.1**	-	-
AMSNet [10]	ECCV'20	-	-	-	-	101.8	163.2	56.7	93.4	**6.7**	**10.2**
AMRNet [27]	ECCV'20	-	-	-	-	86.6	152.2	61.6	98.4	7.0	11.0
LibraNet [24]	ECCV'20	-	-	-	-	88.1	**143.7**	**55.9**	97.1	7.3	11.3
DM-count [44]	NeurIPS'20	88.4	357.6	68.4	283.3	85.6	148.3	59.7	95.7	7.4	11.8
NoiseCC [41]	NeurIPS'20	96.9	534.2	67.7	**258.5**	85.8	150.6	61.9	99.6	7.4	11.3
Ours		**79.3**	**346.1**	**59.9**	259.5	**84.3**	147.5	61.3	95.4	7.3	11.7

Table 4: Localization performance on NWPU-Crowd dataset.

	Precision	Recall	F-measure
Faster RCNN [25]	**0.958**	0.035	0.068
TinyFace [9]	0.529	**0.611**	0.567
VGG+GPR	0.558	0.496	0.525
RAZNet [23]	0.666	0.543	0.599
ours	0.800	0.562	**0.660**

Table 5: Localization performance on UCF-QNRF dataset.

	Precision	Recall	AUC
MCNN [50]	0.599	0.635	0.591
ResNet [8]	0.616	0.669	0.612
DenseNet [11]	0.702	0.581	0.637
Encoder-Decoder [2]	0.718	0.630	0.670
CL [13]	0.758	0.598	0.714
DM-Count [44]	0.731	0.638	0.692
VGG19+L2	0.605	0.670	0.623
VGG19+BL [29]	0.767	0.654	0.720
VGG19+ours	**0.782**	**0.748**	**0.763**

3 largest datasets and competitive performance on ShanghaiTech, without any special design to extract multi-scale features or to handle noisy annotations. The experiment confirms the effectiveness of the proposed loss function.

5.4. Localization

Finally, since our loss function trains the model to predict compact density maps that are suitable for localization, we evaluate the localization performance on NWPU-Crowd and UCF-QNRF. We compare against other state-of-the-art that have reported results on localization, and the results are presented in Tables 4 and 5. On NWPU-Crowd, our loss achieves the overall best performance as quantified by F-measure. Faster RCNN, a detector-based approach, has the highest precision but lowest recall, which shows that it cannot handle the small objects far from the camera. In contrast, TinyFace has the highest recall, but the lowest precision, showing that it has many false-positives. Our loss yields a more balanced localization result, obtaining the best F-measure, and the 2nd best precision and recall. RAZNet also achieves better performance than the detection-based methods, via its recurrent zooming mechanism for handling small objects. However, our loss outperforms RAZNet, without any special design for predicting high-resolution density maps or zooming mechanism. One localization example from the NWPU-Crowd test set is shown in Fig. 7.

On UCF-QNRF, the proposed loss outperforms other loss functions including composition loss (CL), which is designed for localization. Our loss also outperforms its baselines L2 and BL, showing the efficacy of the proposed loss over the purely pixel-wise and point-wise losses. The experiment demonstrates that the proposed loss can be naturally used for localization, since the density is encouraged to be compact around the annotations during optimization with transport loss and exponential cost.

In Supp. D, we show a comparison of the localization results for different loss functions. For L2 loss, many false negatives appear in dense regions, and the recall is the worst, which shows that L2 loss cannot handle small objects far from the camera. BL and DM-Count have better recall, but BL has many false positives in high-density regions, and DM-Count has many false positives even in low-density regions. Our proposed loss achieves both high precision and recall, yielding the best F-measure.

Figure 7: Example localization result on NWPU-Crowd test set.

6. Conclusion

In this paper, we propose a generalized loss function for learning density maps for crowd counting and localization, which is based on unbalanced optimal transport. We prove that traditional L2 and Bayesian loss are special cases and suboptimal solutions of our loss function. A perspective-guided cost function is proposed to handle perspective transformation in crowd images. We then conduct extensive experiments and achieve superior performance on large-scale datasets. Finally, we apply the proposed loss function to crowd localization and achieve the best performance without any special design of the architecture.

Acknowledgments. This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11212518).

References

[1] Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019. 4

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 8

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 8

[4] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008. 1, 2

[5] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *International Conference on Computer Vision*, pages 545–551, 2009. 2

[6] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019. 6

[7] Tao Han, Junyu Gao, Yuan Yuan, and Qi Wang. Focus on semantic consistency for cross-domain crowd understanding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1848–1852. IEEE, 2020. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[9] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 951–959, 2017. 8

[10] Yutao Hu, Xiaolong Jiang, Xuhui Liu, Baochang Zhang, Jungong Han, Xianbin Cao, and David Doermann. Nas-count: Counting-by-density with neural architecture search. *arXiv preprint arXiv:2003.00217*, 2020. 8

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 8

[12] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013. 1, 2

[13] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018. 2, 5, 8

[14] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4706–4715, 2020. 8

[15] Di Kang and Antoni B. Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *British Machine Vision Conference*, page 89, 2018. 2

[16] Di Kang, Zheng Ma, and Antoni B Chan. Beyond counting: comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 5

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 878–885, 2005. 2

[19] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010. 2

[20] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *International Conference on Pattern Recognition*, pages 1–4, 2008. 2

[21] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, 2018. 6, 8

[22] Dongze Lian, Jing Li, Jia Zheng, Weixin Luo, and Shenghua Gao. Density map regression guided detection network for rgb-d crowd counting and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1830, 2019. 3

[23] Chenchen Liu, Xinyu Weng, and Yadong Mu. Recurrent attentive zooming for joint crowd counting and precise localization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1217–1226. IEEE, 2019. 2, 8

[24] Liang Liu, Hao Lu, Hongwei Zou, Haipeng Xiong, Zhiguo Cao, and Chunhua Shen. Weighing counts: Sequential crowd counting by reinforcement learning. *arXiv preprint arXiv:2007.08260*, 2020. 8

[25] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. Crowd counting with deep structured scale integration network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1774–1783, 2019. 8

[26] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[27] Xiyang Liu, Jie Yang, and Wenrui Ding. Adaptive mixture regression network with local counting map for crowd counting. *arXiv preprint arXiv:2005.05776*, 2020. 8

[28] Yan Liu, Lingqiao Liu, Peng Wang, Pingping Zhang, and Yinjie Lei. Semi-supervised crowd counting via self-training

on surrogate tasks. *arXiv preprint arXiv:2007.03207*, 2020. 2

[29] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6142–6151, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[30] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *European Conference on Computer Vision*, pages 278–293, 2018. 2

[31] Deepak Babu Sam and R. Venkatesh Babu. Top-down feedback for crowd counting convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7323–7330, 2018. 2

[32] Deepak Babu Sam, Skand Vishwanath Peri, Mukuntha Narayanan Sundararaman, Amogh Kamath, and Venkatesh Babu Radhakrishnan. Locate, size and count: Accurately resolving people in dense crowds via detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3, 8

[33] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4031–4039, 2017. 2, 8

[34] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5245–5254, 2018. 8

[35] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5382–5390, 2018. 2

[36] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *IEEE International Conference on Computer Vision*, pages 1879–1888, 2017. 2, 8

[37] Vishwanath A Sindagi and Vishal M Patel. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1002–1012, 2019. 8

[38] Vishwanath A Sindagi, Rajeev Yasarla, Deepak Sam Babu, R Venkatesh Babu, and Vishal M Patel. Learning to count in the crowd from limited labeled data. *arXiv preprint arXiv:2007.03195*, 2020. 2

[39] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *Technical Report*, 2020. 5

[40] Jia Wan and Antoni Chan. Adaptive density map generation for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1130–1139, 2019. 1, 2, 3

[41] Jia Wan and Antoni Chan. Modeling noisy annotations for crowd counting. In *Advances in Neural Information Processing Systems*, to appear Dec 2020. 1, 2, 6, 8

[42] Jia Wan, Wenhan Luo, Baoyuan Wu, Antoni B Chan, and Wei Liu. Residual regression and semantic prior for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 7

[43] Jia Wan, Qingzhong Wang, and Antoni B Chan. Kernel-based density map generation for dense object counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 6, 8

[44] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. In *Advances in Neural Information Processing Systems*, to appear Dec 2020. 2, 6, 7, 8

[45] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3, 5

[46] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8198–8207, 2019. 2

[47] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. Spatiotemporal modeling for crowd counting in videos. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5161–5169, 2017. 2

[48] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. Reverse perspective network for perspective-aware object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4374–4383, 2020. 8

[49] Cong Zhang, Hongsheng Li, X. Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015. 2

[50] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016. 2, 5, 6, 8

[51] Zhen Zhao, Miaojing Shi, Xiaoxiao Zhao, and Li Li. Active crowd counting with limited supervision. *arXiv preprint arXiv:2007.06334*, 2020. 2