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-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 \( \mathcal{A} = \{(a_i, x_i)\}_i \), where \( a_i \) is the predicted density of pixel \( x_i \in \mathbb{R}^2 \) and \( n \) is the number of pixels. We denote \( \hat{a} = [a_i] \) as the predicted density map. The ground-truth dot map is \( \mathcal{B} = \{(b_j, y_j)\}_j \), where \( y_j \) 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 \( \mathcal{B} = [b_j]_j = 1_m \).

Our loss function is based on the entropic-regularized unbalanced optimal transport cost, 
\[
L^\varepsilon_C(\mathcal{A}, \mathcal{B}) = \min_{\mathcal{P} \in \mathbb{R}^n \times m} \langle C, \mathcal{P} \rangle - \varepsilon H(\mathcal{P}) + \tau D_1(\mathcal{P}1_m|\hat{a}) + \tau D_2(\mathcal{P}^\top 1_n|b).
\]
\[
C \in \mathbb{R}^n \times m \text{ is the transport cost matrix, whose entry } C_{ij} \text{ measures the cost of moving the predicted density at } x_i \text{ to GT dot annotation } y_j \text{ via the cost function } C_{ij} = c(x_i, y_j). \]
\( \mathcal{P} \) is the transport matrix, which (fractionally) assigns each each location \( x_i \) from \( \mathcal{A} \) to \( y_j \) from \( \mathcal{B} \) for measuring the cost. The optimal transport cost is obtained by minimizing the loss over \( \mathcal{P} \). Note that \( \hat{a} = \mathcal{P}1_m \) is the construction of an intermediate density map representation from the GT annotations, while \( \hat{b} = \mathcal{P}^\top 1_n \) is the reconstruction of the GT dot annotations. See Fig. 2 (top) for an example.

The loss function decomposes into four terms. The first term \((C, \mathcal{P})\) 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 \(H(\mathcal{P}) = -\sum_{ij} P_{ij} \log P_{ij} \) is the entropic regularization term, which favors partial transports between locations, resulting in spread-out (less compact) density maps. Larger values of \( \varepsilon \) will yield less compact predicted density maps, and vice-versa. The third term \(D_1(\mathcal{P}1_m|\hat{a})\) is the pixel-wise loss between the predicted density map \( \hat{a} \) and the constructed intermediate density map representation \( \mathcal{A} = \mathcal{P}1_m \), i.e., the “ground-truth” density map. Finally, the fourth term \(D_2(\mathcal{P}^\top 1_n|b)\) is the point-wise loss between the reconstructed annotations \( \hat{b} = \mathcal{P}^\top 1_n \) 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 our implementation, we use squared L2 norm for the pixel-wise term and L1-norm for the point-wise term,
\[
D_1(\mathcal{P}1_m|\hat{a}) = \|\mathcal{P}1_m - \hat{a}\|^2_2, \quad D_2(\mathcal{P}^\top 1_n|b) = \|\mathcal{P}^\top 1_n - b\|_1.
\]

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, \( L^2_{ij} = \|x_i - y_j\|^2 \), which considers all distances equally throughout the image. How-
ever, 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:

$$C_{ij} = \exp\left(\frac{1}{\eta(x_i, y_j)} \|x_i - y_j\|_2\right),$$

(4)

where $\eta(x_i, y_j)$ is an adaptive perspective factor, which is mapped between an interval based on the average height $\frac{1}{2}(h_x + h_y)$, where $h_x \in [0, 1]$ refers to normalized height of the pixel $x$ 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 $P$ is unique, and has the form

$$P = \text{diag}(u)K\text{diag}(v), \quad K = \exp(-C/\epsilon),$$

(5)

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

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

(6)

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

### 3.4. Density map counting and localization

To apply our loss function to density mapping, we learn a density map estimator $f(I)$, whose input is the image $I$, and output is the density map vector $a$. The predicted density map $A$, together with the corresponding GT annotations $B$, are fed into the loss function in (1). To compute the loss, the transport matrix $P$ 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 $a$. First, $a$ is up-sampled 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 $3 \times 3$ window centered on the pixel. Finally, the candidates with density larger than 0.05 are the final location predictions.

### 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 $P$ using entropic-regularized balanced OT with squared Euclidean transport cost; 2) substituting the approximate $P$ into our loss in (1). Because the computed $P$ 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 $P$

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

$$\mathcal{L}_C(A, B) = \min_{P \in \mathbb{R}^n \times m} \langle C, P \rangle - \epsilon H(P),$$

(7)

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

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

(8)

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

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

(9)

where the division is element-wise. Typically, the iterations are initialized with $v = 1_n$.

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

$$\tilde{P}_{ij} = \left[\frac{a}{Kv^{(\ell)}}\right]_{ij} K_{ij} v_j, \quad \hat{P}_{ij} = u_i K_{ij} \left[\frac{b}{K^\top u^{(\ell+1)}}\right]_j.$$  

(10)

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

$$\tilde{P}_{ij} = \frac{K_{ij} a_i}{\sum_{j=1} K_{ij}} a_i, \quad \hat{P}_{ij} = \frac{\exp(-\|x_i - y_j\|^2/\epsilon)}{\sum_{i=1} \exp(-\|x_i - y_j\|^2/\epsilon)}.$$  

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

$$\tilde{P}_{ij} = \frac{K_{ij} b_j}{\sum_{i=1} K_{ij}} b_j, \quad \hat{P}_{ij} = \frac{\exp(-\|x_i - y_j\|^2/\epsilon)}{\sum_{i=1} \exp(-\|x_i - y_j\|^2/\epsilon)},$$

since $b_j = 1$. Note that the difference between $\tilde{P}_{ij}$ and $\hat{P}_{ij}$ is the summation in the denominator is either over GT annotation locations $y_j$ or density map pixels $x_i$, 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 $\tilde{P}$. Substituting into (1), we note that the first 2 terms, $\langle C, \tilde{P} \rangle$ and $H(\tilde{P})$ are constants w.r.t. $a$, and thus do not affect the loss in terms of $a$. Next, it is straightforward to show

1Solving for $P$ using $D_1$ and $D_2$ in (2) and (3) requires an inefficient nested optimization.
shows a visualization of the weights maps for QNRF \cite{li2020}, and therefore the fourth term in (1) is $D_2(\tilde{P}^T1_{m}|b) = 0$. Only the third term (i.e., the pixel-wise loss) remains, and assuming $\tau = 1$, we have the loss

$$\hat{L}(A, B) = D_1(P1_{m}|a) = \sum_{i=1}^{n}(a_i - \hat{a}_i)^2,$$

(11)

$$\hat{a}_i = \tilde{P}_{1m}|i = \sum_{j=1}^{m} \tilde{P}_{ij} = \sum_{j=1}^{m} \exp\left(-\frac{||x_i - y_j||^2}{\epsilon}\right).$$

(12)

Note that $\hat{a}_i$ 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 $y_j$. The denominators in (12) are the normalization constants of each Gaussian. Thus from (11) and (12), our loss using the approximate transport matrix $\tilde{P}$ is equivalent to L2 loss with traditional Gaussian-based density maps for supervision.

### 4.3. Relationship with Bayesian Loss

We next derive BL \cite{li2020} as a special case of our loss using approximation $\tilde{P}$. Note that $\tilde{P}_{ij}$ is the probability of assigning the density value of the $i$-th pixel to the $j$-th annotation point, as defined in \cite{li2020}. Since $[\tilde{P}_{1m}]_i = \sum_{j=1}^{m} \tilde{P}_{ij}a_i = a_i$, then the third term in (1) is $D_1(P1_{m}|a) = 0$. Assuming that $\epsilon$ is small (so that the entropy term can be ignored) and $\tau = 1$, we have the loss

$$\hat{L}(A, B) = \langle C, \tilde{P} \rangle + D_2(\tilde{P}^T1_{m}|b)$$

(13)

$$= \sum_{i=1}^{n} \omega_i a_i + \sum_{j=1}^{m} \left[1 - \sum_{i=1}^{n} \tilde{P}_{ij}a_i\right],$$

(14)

where $\omega_i = \sum_{j=1}^{m} C_{ij} \tilde{P}_{ij} = \sum_{j=1}^{m} ||x_i - y_j||^2 \tilde{P}_{ij}$ 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 $x_i$ far from any annotation (i.e., false positives). In particular, the weight $\omega_i$ 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 \cite{li2020}. The background loss in [29] is based on the nearest annotation to each point $x_i$ (details in Supp. B),

$$L_{BG} = ||0 - \sum_{i=1}^{n} \omega_i a_i|| = \sum_{i=1}^{n} \omega_i a_i,$$

(15)

where the weight $\tilde{w}_i = \frac{\bar{k}_i}{\bar{k}_i + \sum_{j=1}^{m} K_{ij}}$ and $\bar{k}_i = \exp(-(d - ||x_i - y_{\eta(i)}||^2)/\epsilon)$, and $\eta(i)$ is the index of the annotation nearest to $x_i$. The weight can be rewritten as $\tilde{w}_i = \exp\left(\frac{2}{\epsilon^2} ||x_i - y_{\eta(i)}|| - \bar{k}_{\tilde{w}} \right) \tilde{P}_{ij}$, where $\bar{P}_{ij} = \frac{K_{ij}}{\bar{k}_i + \sum_{j=1}^{m} K_{ij}}$ 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 $\tilde{w}_i$ or $\omega_i$. For BL, the weight $\tilde{w}_i$ is based on the exponential distance to the nearest annotation $\eta(i)$. In contrast, for the background term in (14), the weight $\omega_i$ is based on a weighted average of squared distances to all annotations. Therefore, the background model used in \cite{li2020} is a special case of the background term in (14). Fig. 3 shows a visualization of the weights maps for $\tilde{w}_i$ and $\omega_i$. 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.

Thus, from (14), BL with background model is a special case of our proposed loss in (1), where the approximate transport matrix is $\tilde{P}$ and a single-neighbor approximation of the cost matrix $C$ 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 \cite{zhu2016}, UCF-QNRF \cite{li2017}, JHU-CROWD++ \cite{li2020}, and NWPU-Crowd \cite{shi2018}. 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 \cite{li2018}

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|, \quad \text{MSE} = \left( \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \right)^{1/2},$$

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
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 \( P \) is solved using the generalized Sinkhorn iterations in (6). We set \( \varepsilon = 0.005 \). In practice, \( \varepsilon \)-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 \( \eta \) (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 compared to squared Euclidean cost (see Sec. 5.2.2).

#### 5.2.4 The effect of \( \varepsilon \)

We next investigate the effect of blur factor \( \varepsilon \), 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 \( \varepsilon \) are shown in Fig. 4b. The L2 loss is sensitive to \( \varepsilon \), with MAE increasing significantly as \( \varepsilon \) 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-
ers the nearest annotation, while our loss considers all annotations (see Sec. 4.3). A visualization is shown in Fig. 5. As the \( \varepsilon \) increases, the density map for L2 becomes more blurry and inaccurate, which demonstrates that L2 is sensitive to \( \varepsilon \). BL and our loss are generally robust to the choice of \( \varepsilon \), 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 \( \tau \) and divergence function

Next, we study the effect of \( \tau \) 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 \( \tau \) (see Fig. 4c), and visualize the learned density maps in Fig. 6. As \( \tau \) decreases, the density maps become more compact (more sparse), since the transport cost dominates and 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
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.

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


[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


