# Supplementary Material for Crowd Localization on Density Map via Optimal Transport Minimization

Wei Lin Antoni B. Chan
Department of Computer Science, City University of Hong Kong
elonlin24@gmail.com, abchan@cityu.edu.hk

## 1. Pseudo-Code of OT-M

Algo. 1 summarizes the procedure of OT-M algorithm. At the beginning, the number of elements in $\mathcal{B}$ is calculated by summing up $[a_i]$ in $\mathcal{A}$, and then initializing $[\boldsymbol{y}_j]$ randomly using *top-k*, *uniform*, or *adaptive* initialization. After that, OT-step and M-step are performed repetitively.

In $k$-th OT-step, cost matrix $\mathbf{C}^{(k)}$ is firstly computed between $\mathcal{A}$ and $\mathcal{B}^{(k)}$, and the Gibbs kernel is obtained accordingly. Next, the Sinkhorn algorithm is implemented in a sub-loop to find $\hat{\mathbf{v}}$ and $\hat{\mathbf{u}}$, which are then used to construct the $k$-th transport plan $\mathbf{P}^{(k)} = \text{diag}(\hat{\mathbf{u}})\mathbf{K}^{(k)}\text{diag}(\hat{\mathbf{v}})$.

In $k$-th M-step, the barycenter of density assigned to $j$-th point in $\mathcal{B}^{(k-1)}$ by transport plan $\mathbf{P}^{(k)}$ is calculated as the updated coordinates, $\boldsymbol{y}_j^{(k)}$. On CPU, M-step is implemented via a loop (Line.14∼17 in Algo. 1). However, the loop can be performed in parallel, and thus the M-step reduces to one step using matrix operations on GPU:

$$\boldsymbol{y} = \text{diag}\left(\mathbf{P}^\top \mathbf{1}_m\right) \mathbf{P}^\top \boldsymbol{x}. \tag{1}$$

Once the stopping criterion is met, OT-M will be stopped and return the final solution, $\hat{\mathcal{B}} = \{\hat{\boldsymbol{y}}_j\}_{j=1}^m$.

The source code is available at https://github.com/Elin24/OT-M.

**Algorithm 1** OT-M Algorithm

**Input:** Soft label $\mathcal{A} = \{a_i, \boldsymbol{x}_i\}_{i=1}^n$, the blur factor $\varepsilon$.
**Output:** Hard label $\mathcal{B}$.

1: Get the size of $\mathcal{B}$: $m = \lfloor \sum_{i=1}^n a_i \rceil$.
2: Initialize $\mathcal{B}^{(0)} = \{\boldsymbol{y}_j^{(0)}\}_{j=1}^m$ with $\boldsymbol{y}_j^{(0)} \in \mathbb{R}^2$ randomly.
3: Normalize $\boldsymbol{a}\&\boldsymbol{b}$ for balanced OT: $\boldsymbol{a} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$, $\boldsymbol{b} = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}$.
4: **repeat**
5: {*OT-step*}
6: Compute the cost matrix $\mathbf{C}^{(k)}$ according to squared Euclidean distance: $C_{ij}^{(k)} = \|\boldsymbol{x}_i - \boldsymbol{y}_j^{(k)}\|_2^2$
7: Get Gibbs kernel $\mathbf{K}^{(k)} \leftarrow \exp(\mathbf{C}^{(k)}/\varepsilon)$.
8: Initialize $\mathbf{v}^{(0)} \leftarrow \mathbf{1}_m$.
9: **repeat**
10: $\mathbf{u}^{(l+1)} \leftarrow \boldsymbol{a}/\left(\mathbf{K}^{(k)}\mathbf{v}^{(l)}\right)$,
11: $\mathbf{v}^{(l+1)} \leftarrow \boldsymbol{b}/\left({\mathbf{K}^{(k)}}^\top \mathbf{u}^{(l+1)}\right)$,
12: **until** reaching an equilibrium state with $\hat{\mathbf{v}}$ and $\hat{\mathbf{u}}$.
13: Calculate plan $\mathbf{P}^{(k)} = \text{diag}(\hat{\mathbf{u}})\mathbf{K}^{(k)}\text{diag}(\hat{\mathbf{v}})$
14: {*M-step*}
15: **for** $j \leftarrow 1$ to $m$ **do**
16: $\boldsymbol{y}_j^{(k)} \leftarrow \left(\sum_{i=1}^n P_{ij}^{(k)} \boldsymbol{x}_i\right) / \left(\sum_{i=1}^n P_{ij}^{(k)}\right)$
17: **end for**
18: **until** $[\boldsymbol{y}_j^{(k)}]$ converges to $[\hat{\boldsymbol{y}}_j]$.
19: **return** $\hat{\mathcal{B}} = \{\hat{\boldsymbol{y}}_j\}_{j=1}^m$.

## 2. GMM for Localization with Density Map

Besides the proposed OT-M algorithm, Gaussian Mixture Model(GMM) can also be used to locate objects on density map [1]. For convenience, we normalize $\mathcal{A} = \{(a_i, \boldsymbol{x}_i)\}_{i=1}^n$ ($\sum_{i=1}^n a_i = 1$) so that $a_i$ and $\boldsymbol{x}_i \in \mathbb{R}^2$ indicate the normalized density value and coordinate of the $i$-th pixel. In GMM, $\mathcal{B} = \{(b_j, \boldsymbol{y}_j, \sigma_j^2)\}_{j=1}^m$ is a set of $m$ independent Gaussian distributions. The mean of $j$-th one is $\boldsymbol{y}_j$, its weight is $b_j$ and the covariance is $\sigma_j^2 \mathbf{I}_{2\times 2}$.

GMM is solved through Expectation-Minimization (EM) algorithm. We compute the soft assignments in the ($l$+1)-th **E-step**. In particular, the likelihood that assigns the $i$-th density pixel to the $j$-th Gaussian distribution is formulated as:

$$\hat{z}_{ij} = p(z_i = j | \boldsymbol{x}_i, \mathcal{B}^{(l)}) = \frac{b_j \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{y}_j, \sigma_j^2)}{\sum_{k=1}^m b_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{y}_k, \sigma_k^2)}, \tag{2}$$

where $\mathcal{N}$ is the 2D Gaussian distribution. After all $\hat{z}_{ij}$ is obtained, we update $\mathcal{B}$ in **M-step** by minimize the likelihood, and the solution is:

$$\boldsymbol{y}_j^{(l+1)} = \frac{1}{\hat{N}_j} \sum_{i=1}^n a_i \hat{z}_{ij} \boldsymbol{x}_i, \quad \hat{N}_j = \sum_{i=1}^n a_i \hat{z}_{ij}, \tag{3}$$

Besides, the parameters in Gaussian distrubtions are up-

| gate | confidence | MAE | MSE |
|---|---|---|---|
| | | 127.69±4.52 | 216.50±11.45 |
| | ✓ | 123.85±5.92 | 212.23±12.02 |
| ✓ | | 125.32±7.62 | 214.96±12.57 |
| ✓ | ✓ | **120.13±7.34** | **208.87±11.65** |

Table S1. Detailed ablation on components of confidence-weighted generalized loss.

dated according to:

$$\sigma_j^{(l+1)} = \sqrt{\frac{1}{\hat{N}_j}\sum_{i=1}^{n} a_i \hat{z}_{ij} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2} \tag{4}$$

$$b_j^{(l+1)} = \frac{\hat{N}_j}{\sum_{i=1}^{n} a_i} = \frac{\hat{N}_j}{m} \tag{5}$$

In practice, we give a limitation that $1 \leq \sigma_j \leq 16$ and $b_j = \frac{1}{m}$ for reasonable estimation.

## 3. Ablation Study on C-GL

In Tab S1, we present details of the ablation study on the proposed confidence-weighted generalized loss, including the gating scheme and confidence weighting strategy. When vanilla GL is adopted as the loss function, the average MAE is 127.69. While only a confidence strategy is adopted, MAE is reduced to 123.85. Meanwhile, if only gate scheme is used, MAE and MSE are 125.32 and 214.96, respectively. However, combining them yields the lowest estimation errors (*e.g.*, MAE: 127.69 → 120.13, MSE: 216.50 → 208.87).

## 4. Ablation Study on $\gamma$

Fig. S1 discusses the influence of different $\gamma$ used in confidence weights (Eq. 20 in the paper). $\gamma = 0$ means confidence is not added in GL, which yields higher MAE and MSE. With the increase of $\gamma$, estimation errors are reduced gradually. However, both metrics increase while a large $\gamma$ is adopted. Thus, in our experiments we set $\gamma = 0.5$.

## 5. More Examples of OT-M

Fig. S2 and Fig. S3 present more examples demonstrating the convergence process of OT-M on synthetic density maps (produced using Gaussian kernels applied to point-maps) and density maps predicted by a CNN, respectively.

In Fig. S4, we present two examples to visualize the influence of the initialization methods. GMM's results are dramatically affected by the initialization – the three initialization methods produce results that are different from each other. Compared with GMM, OT-M is much more robust since OT-M's point maps are similar and close to ground truth.

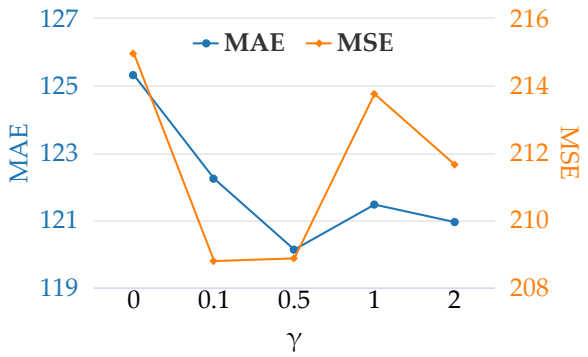


Figure S1. The effect of $\gamma$ on semi-supervised counting performance.

## 6. Limitation

OT-M algorithm is limitted by its efficency. We test a demo image with a resolution of $384 \times 896$. The runtime of P2PNet [5] is 0.016s. The runtime of OT-M consists of two parts: density map estimation (0.013s); and OT-M (0.067s), and the total runtime is 0.080s. OT-M costs more time because of the OT & M iterations. Although M-step is reduced to one step by (1), the loop to obtain the optimal transport map (i.e., the Sinkhorn algorithm) takes about 0.023s for 35 rounds.

During semi-supervised training, input images are cropped into $512 \times 512$. The average training time is 0.34 seconds per sample. For comparison, GP [4], IRAST [3], and DAC [2] take 0.05s, 0.47s, and 0.07s.

## References


[1] 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*, 29(5):1408–1422, 2018. 1

[2] Hui Lin, Zhiheng Ma, Xiaopeng Hong, Yaowei Wang, and Zhou Su. Semi-supervised crowd counting via density agency. In *ACM Multimedia*, 2022. 2

[3] Yan Liu, Lingqiao Liu, Peng Wang, Pingping Zhang, and Yinjie Lei. Semi-supervised crowd counting via self-training on surrogate tasks. In *European Conference on Computer Vision*, pages 242–259. Springer, 2020. 2

[4] 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. In *European Conference on Computer Vision*, pages 212–229. Springer, 2020. 2

[5] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yang Wu. Rethinking counting and localization in crowds: A purely point-based framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3365–3374, 2021. 2

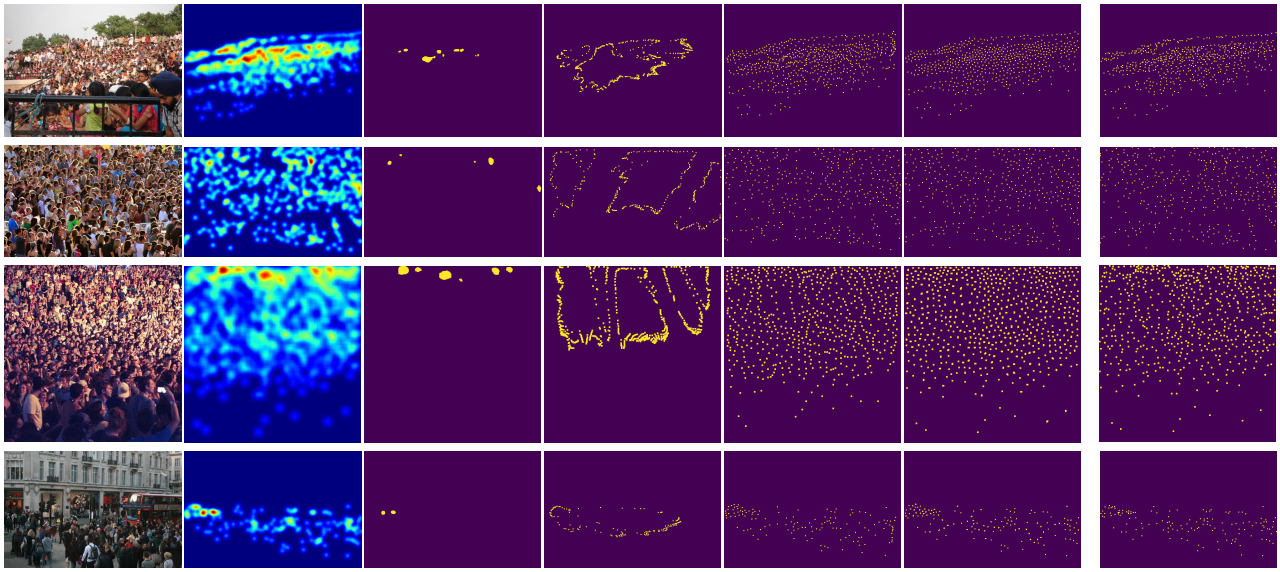


Figure S2. The convergence process of OT-M when synthetic density maps are used.

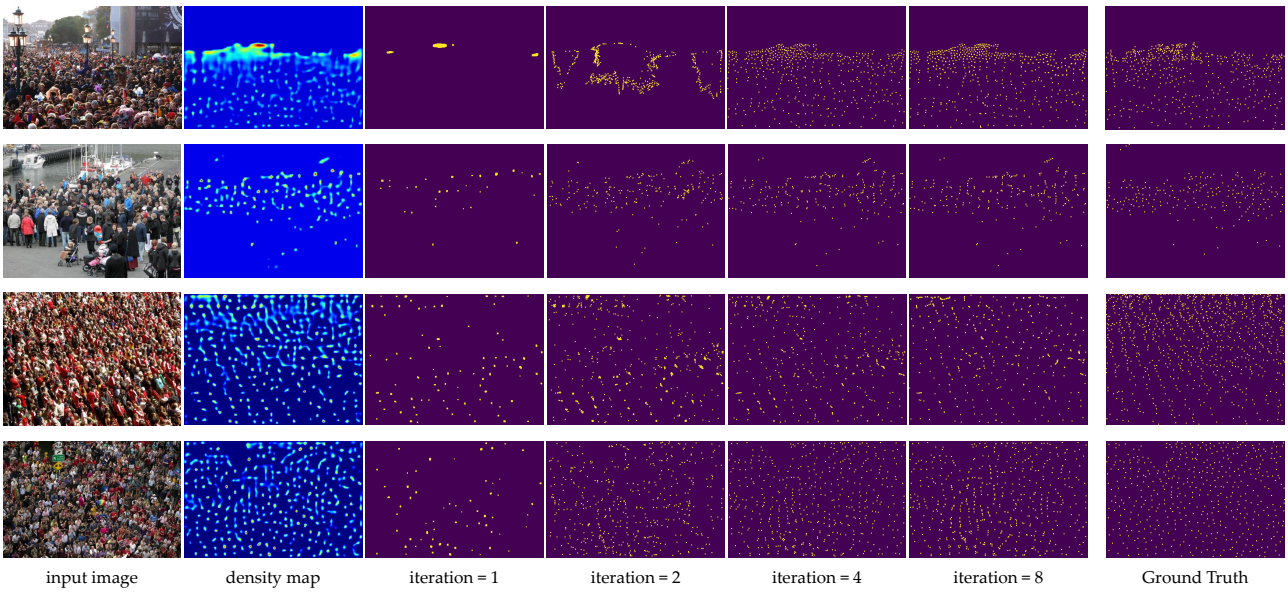


Figure S3. The convergence process of OT-M when density maps are predicted by CNN.

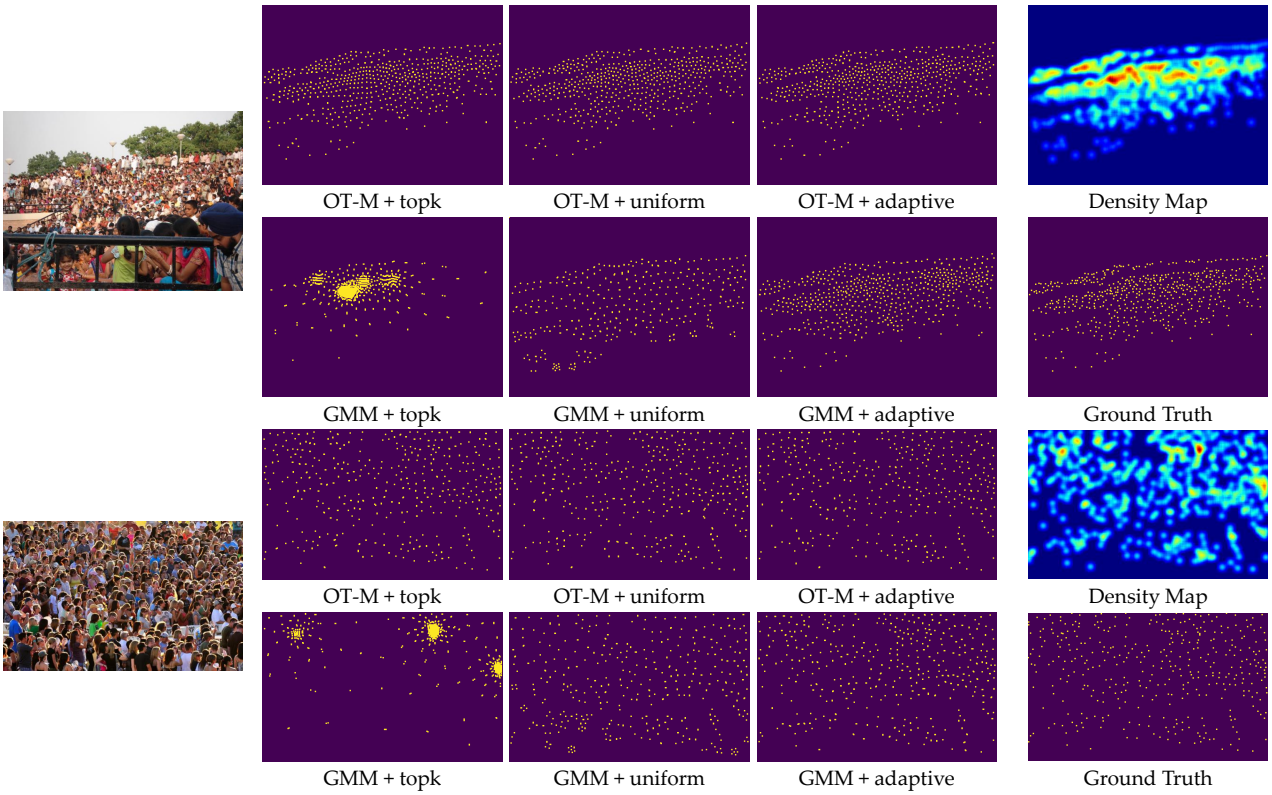


Figure S4. The predicted point maps using different initializations (topk, uniform, adaptive) for OT-M and GMM localization..