Regressor-Segmenter Mutual Prompt Learning for Crowd Counting

Supplementary Material

6. Details of Training a Segmenter

Training via point Annotation. For methods [41, 48, 64], they adopt point-segmentation map P as the target of the segmenter, as formulated in Equation (7).

$$\mathcal{L}_{poi} = -\sum_{i=1}^{N} \sum_{(h=1,w=1)}^{(H,W)} p(h,w) \log \hat{p}(h,w), \tag{7}$$

where N is the batch size, H and W are the height and width of image x_i, respectively. $p(h,w)$ represents the value of position (h,w) in binarized ground-truth density map P_i, and $\hat{p}(h,w)$ is the corresponding predicted value. To this end, we build the mPrompt$_{poi}$ under the loss $\mathcal{L}_{poi}$, formulated as

$$\boldsymbol{\mathcal{L}} = \mathcal{L}_{den} + \lambda_s \mathcal{L}_{poi}, \tag{8}$$

where λ_s is a super-parameter to balance the two losses.

As elucidated in scratch, mPrompt$_{poi}$ exhibits a challenge in assimilating spatial information. This limitation primarily stems from the fact that the targets for both the segmenter and regressor are manually created from dot annotations, which intrinsically do not convey any spatial information.

Training via Box Annotation. To strengthen the segmenter's ability in integrating spatial information, we pretrain it using head-box annotations of NWPU [57] dataset, and generate pseudo mask (m_p) of all datastes for mutual prompt learning. Concretely, suppose $B_{i,v}$ is the v-th box annotation in the image x_i and its annotation is (x_l, y_l, x_r, y_r) representing upper-left corner and lower-right corner. The head box region $R_{i,v}$ is defined as:

$$\begin{aligned} &B^{i,v}_{x_{min}} = x_l, B^{i,v}_{y_{min}} = y_l, B^{i,v}_{x_{max}} = x_r, B^{i,v}_{y_{max}} = y_r \\ &R^i_v = \left\{(x,y) | B^{i,v}_{x_{min}} \leq x \leq B^{i,v}_{x_{max}}, B^{i,v}_{y_{min}} \leq y \leq B^{i,v}_{y_{max}}\right\} \end{aligned} \tag{9}$$

Then the ground-truth box-segmentation map S_i for pretraining the head segmenter is defined as

$$S_i = \cup_{v=1}^{V_i} R^i_v, \tag{10}$$

where V_i denotes the number of boxes in image x_i and $\cup$ indicates the union operator. In this case, for $s(h,w)$ which indicates the value of position (h,w) in the S_i , we have

$$s(h,w) = \mathbb{I}\left((h,w) \in S_i\right). \tag{11}$$

The function $\mathbb{I}(cond)$ is the indicator function, which is equal to 1 only if the condition holds, and 0 otherwise. We utilize $\mathcal{L}_{box}$ to encourage the segmenter to predict a value of 1 for positions falling within any heads, and a value of 0 for positions outside them. Formulately, $\mathcal{L}_{box}$ is defined as

$$\mathcal{L}_{box} = -\sum_{i=1}^{N} \sum_{(h=1,w=1)}^{(H,W)} s(h,w) \log \hat{s}(h,w), \tag{12}$$

where N is the batch size, H and W are the height and width of image x_i, resp. $s(h,w)$ represents the value of position (h,w) in S_i, and $\hat{s}(h,w)$ is the corresponding predicted value.

Similar to $\mathcal{L}_{poi}$ and $\mathcal{L}_{box}$, $\mathcal{L}_{seg}$ in manuscript is implemented on density map ($\hat{y}$) and the pseudo mask (m_p) generated by the pretrained segmenter. For regressor, MSE loss: $\mathcal{L}_{den} = \frac{1}{2N}\sum_{i=1}^{N} \|\hat{Y}_i - Y_i\|_2^2$ is used. $\hat{Y}_i$ is the predicted density map and N the batchsize. Finally, mPrompt is trained with $\mathcal{L}_{seg}$ as follows:

$$\mathcal{L} = \mathcal{L}_{den} + \lambda_s \mathcal{L}_{seg} \tag{13}$$

where λ_s balances the two losses.

7. Details of Extension to Foundation Model

The broadly acknowledged foundational model SAM [25] for image segmentation functions at the pixel level, similar to crowd counting tasks based on density map method. Therefore, SAM has been selected as the foundation model for extending our mPrompt approach, aiming at modifying the hidden representations of a frozen pre-trained model.

The position of adapter. The pre-trained SAM's image encoder, equipped with adapter modules identical to the scaled parallel adapter [15], has supplanted the backbone of our previous architecture. We fixed the parameters of the image encoder, making only a few parameters trainable, including the adapter modules, regressor and segmenter. Specifically, the image encoder is composed of 12 stacked blocks, each containing two types of sublayers: multi-head self-attention (MHA) and a fully connected feed-forward network (FFN).

Adapters are utilized to modify the outputs of MHA and FFN in the transformer blocks. The output from the last adapter module serves as the input for the segmenter. Throughout this process, the adapter modules function as a context prompt (akin to mask prompts in SAM), referred to here as learnable prompt modules.

The performance of training with adapter. To further validate the potential performance enhancement of mPrompt on foundation model, we evaluated its effectiveness on SHA under the same architecture (image encoder + adapter + regressor + segmenter), but with varied training strategies.

These strategies include full fine-tuning without mutual prompt learning, adapter training without mutual prompt learning, and learnable prompt modules with mutual prompt learning. As presented in Table 5, it is evident that our method offers significant improvement opportunities when applied to foundational model.

full ft	adapter	mPrompt
54.8	56.2	53.2

Table 5. Performance on SHA about MAE, when adopting different training strategies.

8. Analysis of Convergence Speed via Context Constraint

We have introduced $\mathcal{L}_{con}$ as a mechanism to guide non-zero values of the density map ($\hat{y}$) to fall within mask ($\hat{m}$), a crucial factor for achieving rapid convergence of the regressor. To validate this assertion, we conducted an experiment comparing convergence speed based on the inclusion or exclusion of $\mathcal{L}_{con}$. Figure 9 displays the Mean Absolute Errors (MAEs) and Mean Squared Errors (MSEs) of the initial 100 epochs during the training process on SHA. The green curve represents the model trained without $\mathcal{L}_{con}$ ($\lambda_{con} = 0$), while the blue curve signifies the model trained with $\lambda_{con} = 1$. Upon examining these results, it becomes clear that both the MAEs and MSEs of the model trained with $\lambda_{con} = 1$ are consistently lower than those of the model trained without it. These findings underscore that the incorporation of λ_{con} effectively aids in achieving faster convergence of the regressor.

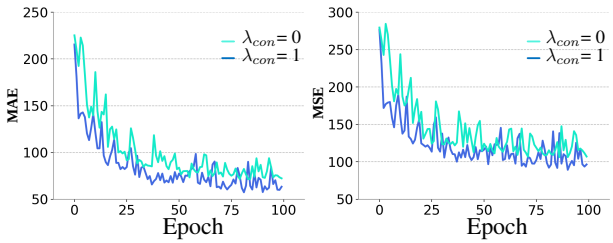


Figure 9. MAEs and MSEs of $\lambda_{con} = 0$ and $\lambda_{con} = 1$.

9. Hyper-parameters

We investigate hyper-parameters including K in K-NN, epoch κ to begin online point prompt, and loss weights λ_d, λ_s, λ_c. Grid-search is infeasible due to computational constraints. Initially, we explore K while fixing other settings at $\kappa = 0$, $\lambda_d = \lambda_s = \lambda_c = 1$. Fig. 10 reveals $K = 3$ as optimal, yielding the lowest MAE of 55.0. Using $K = 3$, we find $\kappa = 50$ reduces MAE to 53.3. Two conclusions can be drawn: 1) An appropriate K can reduce MAE by approximately a gap of 1. For very large K values ($e.g.$, $K = 5$), the performance is similar to $K = 0$. This occurs because a large K means m_K covers nearly the entire image, rendering m_K almost ineffective. 2) $\kappa = 50$ delivers the best MAE, indicating the learning of regressor is fast due to the deployment of $\mathcal{L}_{con}$.

In Table 6, our approach, even when applied with a basic hyper-parameter search, successfully reduces MAE to 52.5. This is achieved with the settings $\lambda_d = 1$, $\lambda_s = 0.5$, and $\lambda_c = 0.5$. We further have the following two conclusions: 1) Employing only λ_d, the network still reduces MAE to MAE 58.4, a marginally performance gain compared to noPrompt$_{reg}$. This affirms the rationality of incorporating the attention as implicit spatial context from segmenter to regressor. Additionally, when λ_s and λ_c are incorporated into the network learning process, we observe further performance gain, underlining the effect of these elements. 2) Upon examining the last three row in Table 6, we confirm that appropriate selection of loss weights can further help enhance performance.

10. Visualization of mPrompt on Tackling Point Annotation Variance in Highly Congested Scenarios

In this section, we delve deeper into the validation of mPrompt's efficiency in addressing point annotation variance in highly crowded scenarios. Figure 11 exhibits the respective density maps as predicted by mPrompt$_{\ddagger}$ and noPrompt$_{reg}$ (named "baseline" in the image). In the presented graphic, we have highlighted certain regions using color-coded boxes for ease of understanding. The areas shaded in blue represent the background regions, wherein noPrompt$_{reg}$ exhibits high activations, contrasting with mPrompt$_{\ddagger}$, which does not. In the regions designated by red boxes, we demonstrate the head areas where noPrompt$_{reg}$ displays inaccurate density blobs, whereas mPrompt$_{\ddagger}$ successfully predicts accurate blobs. The white boxes highlight the head areas that noPrompt$_{reg}$ failed to identify correctly, while, conversely, mPrompt$_{\ddagger}$ delivers correct activations. Lastly, the yellow boxes underscore the head regions where noPrompt$_{reg}$ exhibits activations displaced from the center of the corresponding boxes. In contrast, mPrompt$_{\ddagger}$ generates density blobs precisely at the center of the heads. In summary, for all these four identified situations, mPrompt$_{\ddagger}$ consistently outperforms noPrompt$_{reg}$ in accurately predicting head density blobs.

11. Visualization of mPrompt on Predicting Density Maps

In the main manuscript, we have previously illustrated a selection of examples from the ShanghaiTech Part A

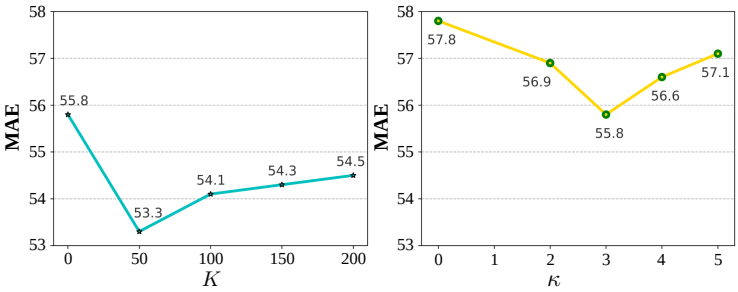


Figure 10. Evaluation of K and κ on SHA.

λ_d	λ_s	λ_c	MAE
1	0	0	58.4
1	1	0	55.9
1	1	1	53.3
1	0.5	1	53.7
1	1	0.5	53.5
1	**0.5**	**0.5**	**52.5**

Table 6. Regularization factors.

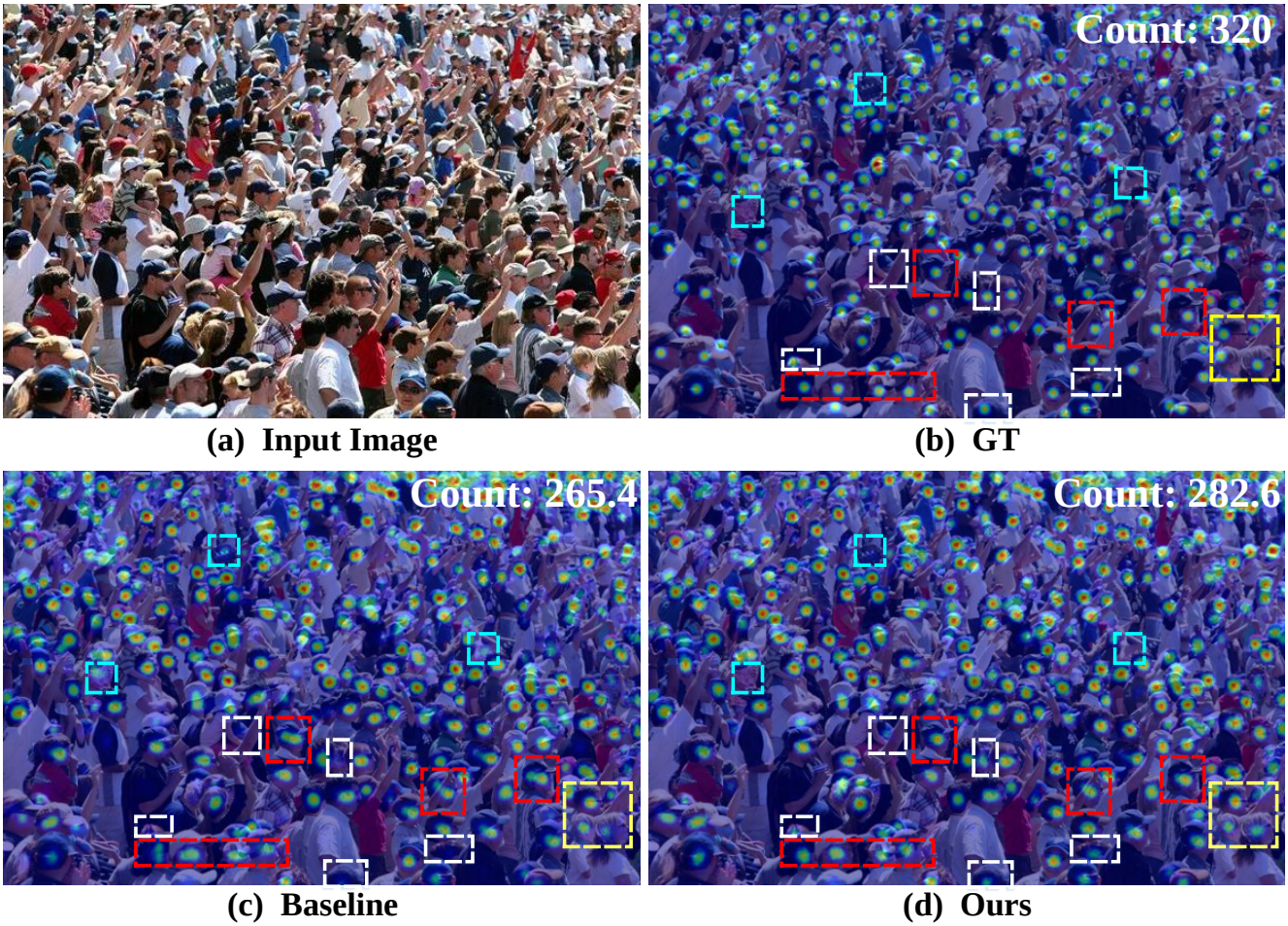
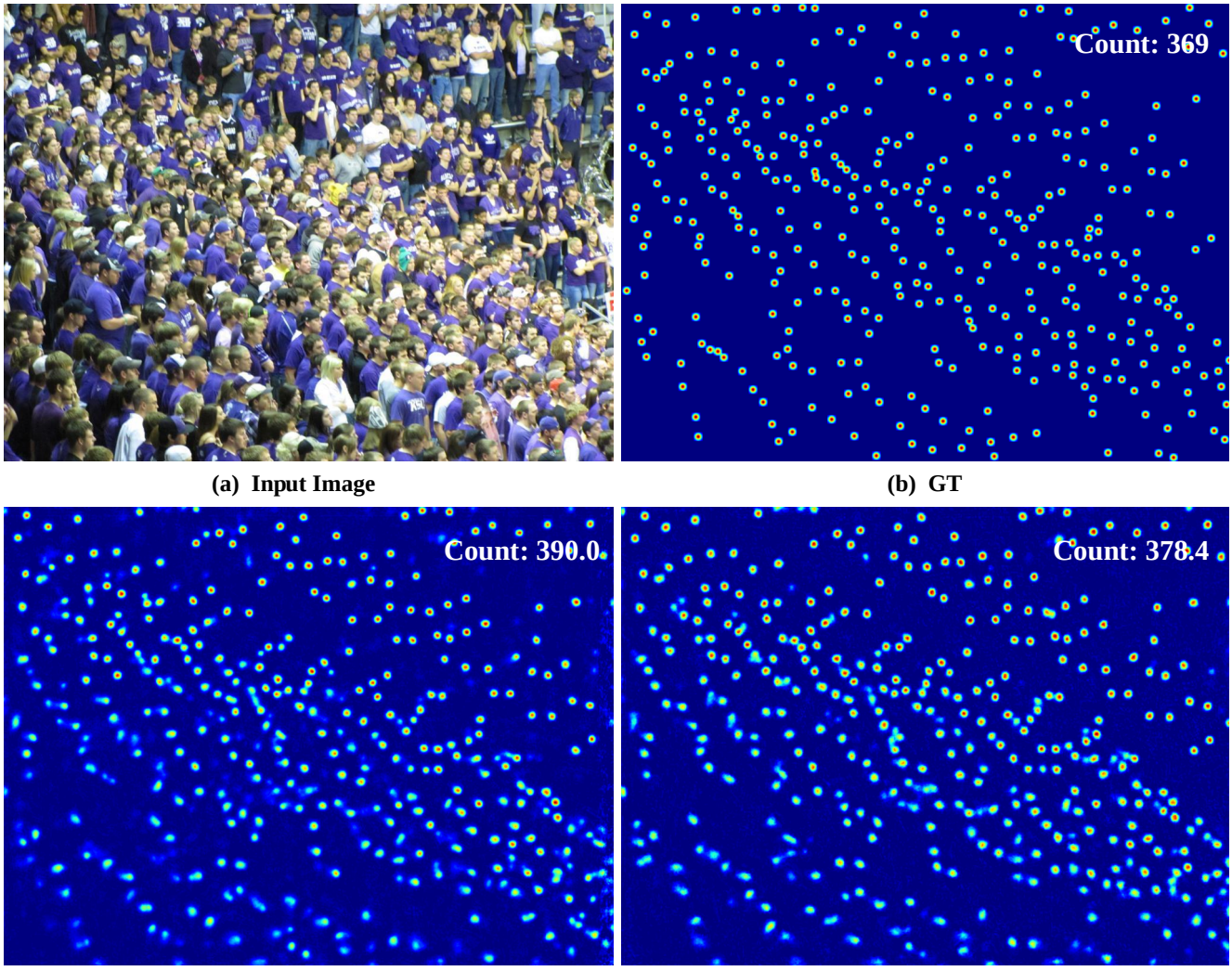


Figure 11. Visualization of density maps. (Best viewed in color)

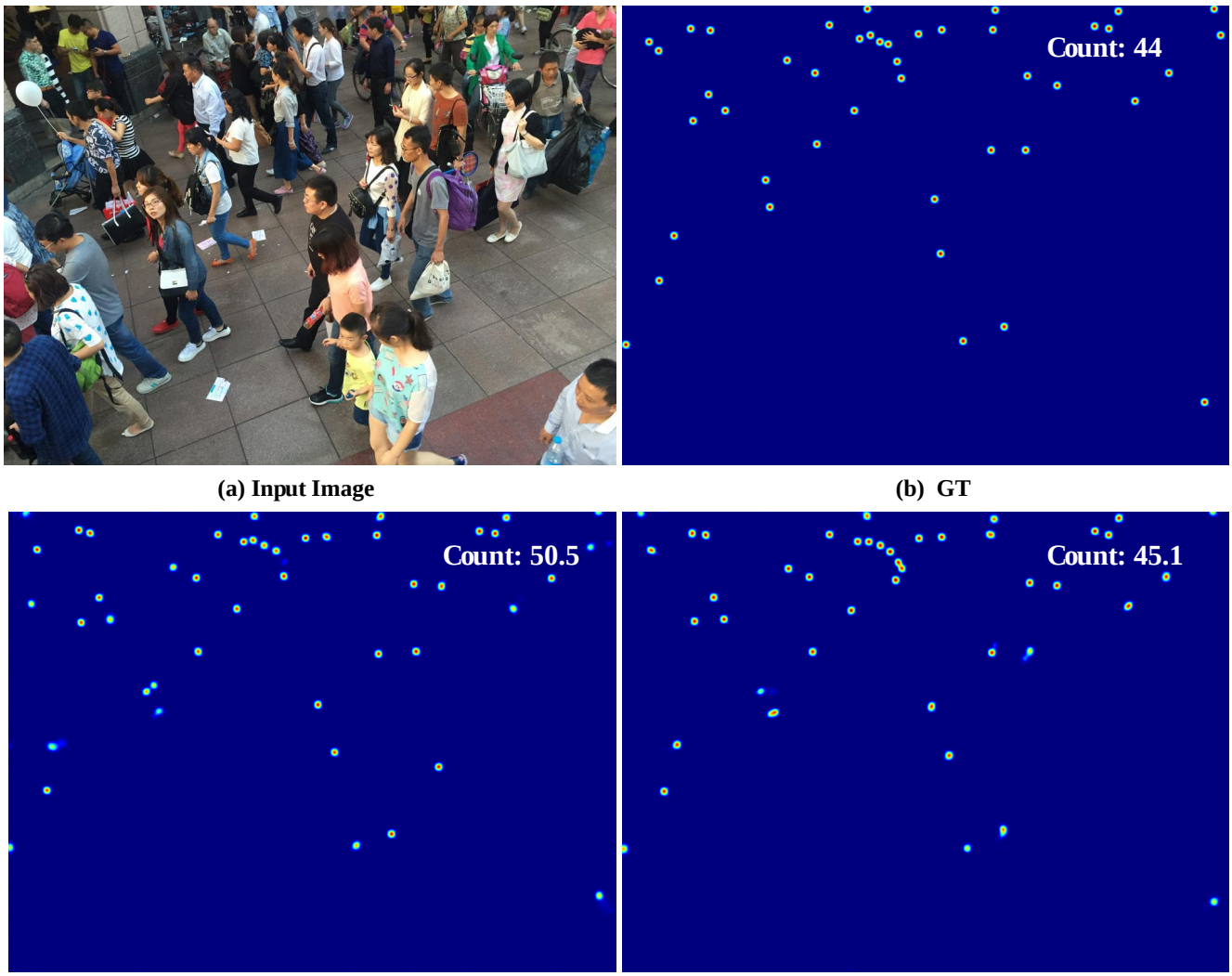
(SHA) dataset. We now expand on this by presenting additional visual results derived from ShanghaiTech Part A (SHA), ShanghaiTech Part B (SHB), UCF-QNRF (QNRF), and NWPU Crowd (NWPU) datasets, corresponding to their respective test samples. As can be observed in Figures 12 13 14 15, mPrompt$_\ddagger$ consistently outperforms noPrompt$_{reg}$ in generating superior density maps. This superiority is apparent across various regions, whether dense or sparse, in each of the SHA, SHB, QNRF, and NWPU datasets. Thus, mPrompt$_\ddagger$ demonstrates marked improvement in performance across different types of crowd scenes.



(a) Input Image **(b) GT**

(c) Baseline **(d) Ours**

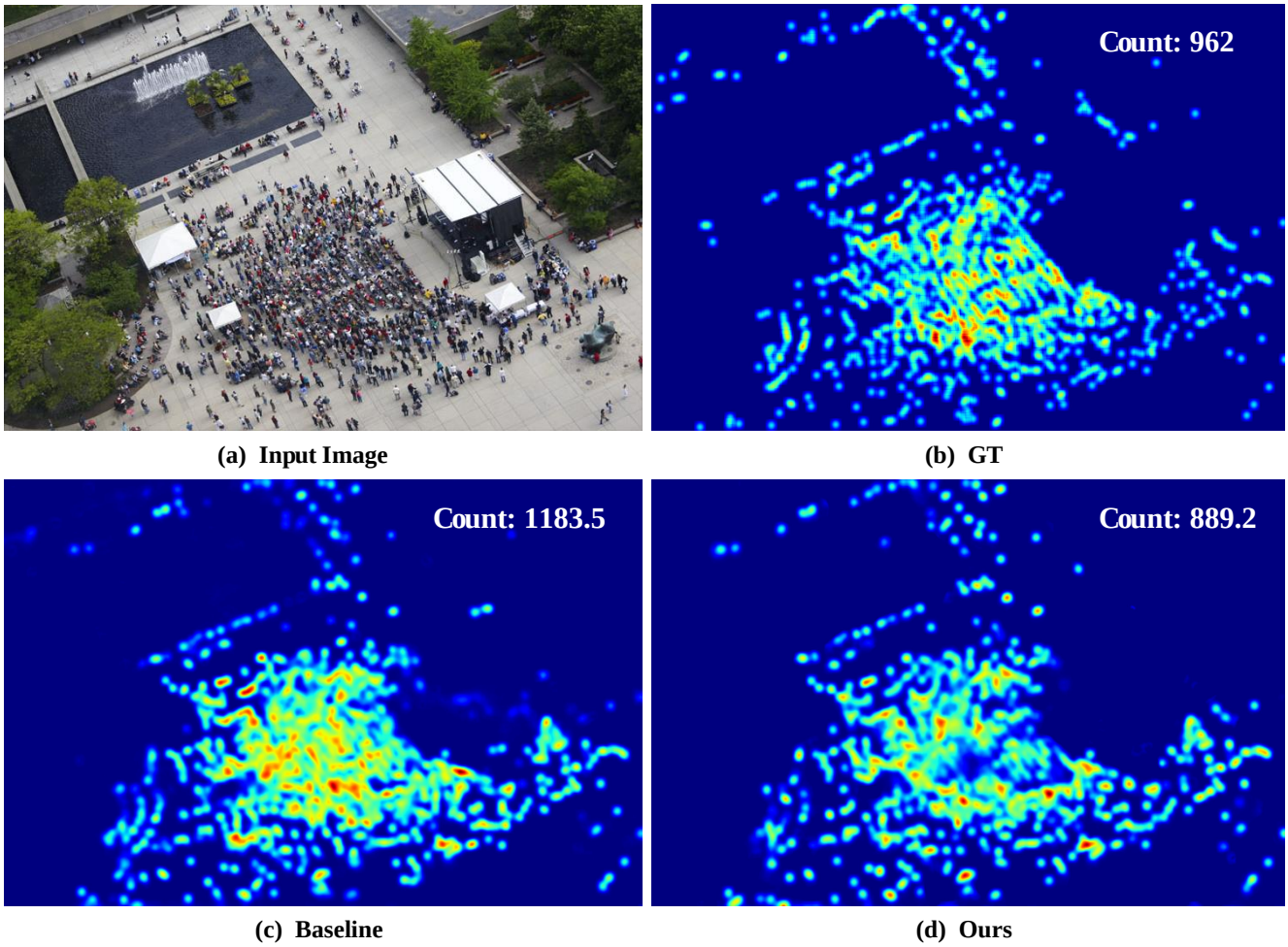
Figure 12. Visualization of predicted density maps from SHA.



(a) Input Image **(b) GT**

(c) Baseline **(d) Ours**

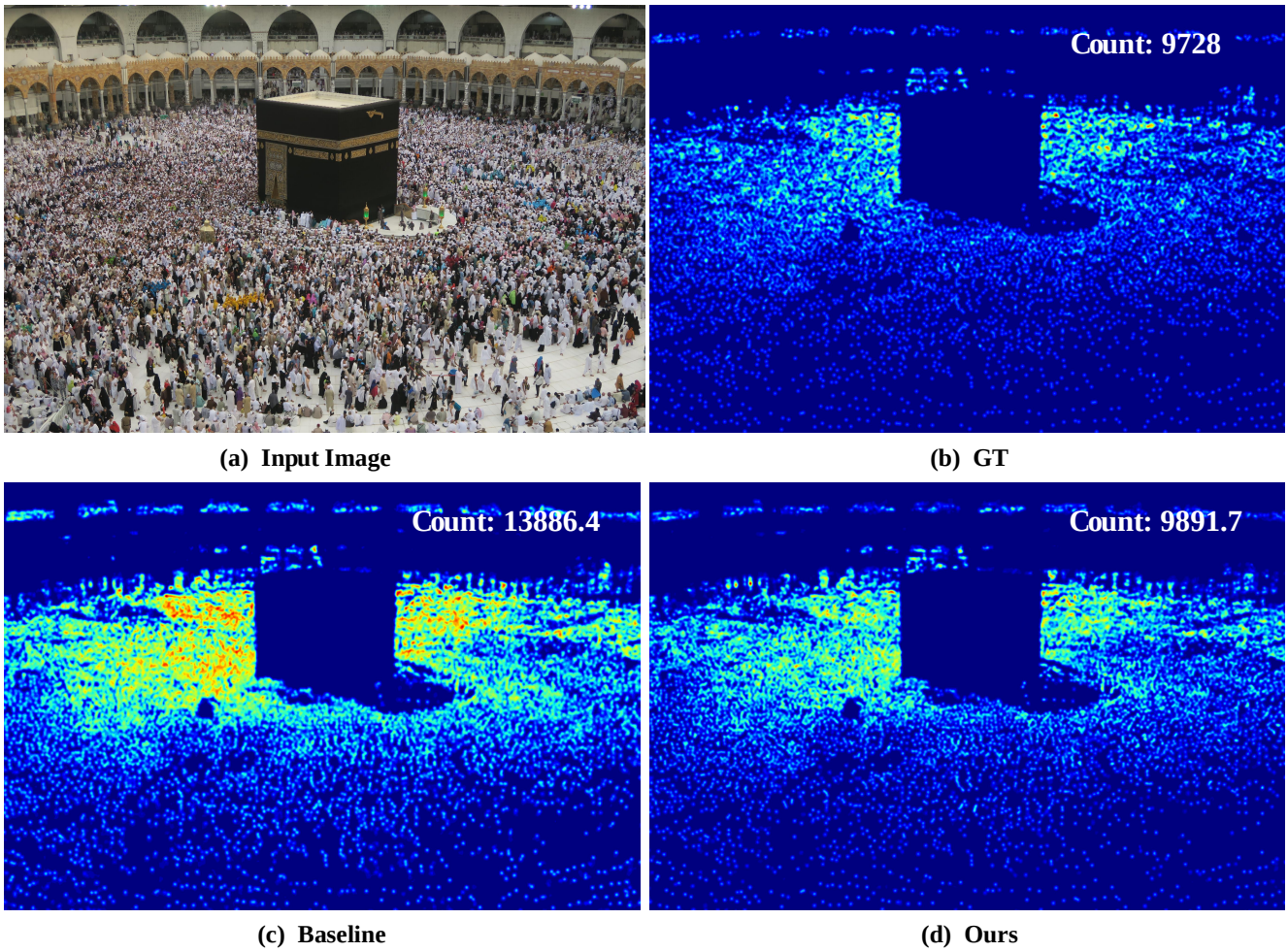
Figure 13. Visualization of predicted density maps from SHB.



(a) Input Image **(b) GT**

(c) Baseline **(d) Ours**

Figure 14. Visualization of predicted density maps from QNRF.



(a) Input Image **(b) GT**

(c) Baseline **(d) Ours**

Figure 15. Visualization of predicted density maps from NWPU.