Improving Point-based Crowd Counting and Localization Based on Auxiliary Point Guidance

I-Hsiang Chen[1], Wei-Ting Chen[1,2], Yu-Wei Liu[1], Ming-Hsuan Yang[2,3], and Sy-Yen Kuo[1,4]

[1] National Taiwan University, Taiwan
[2] University of California at Merced, USA
[3] Google DeepMind, USA
[4] Chang Gung University, Taiwan
{f09921058,f05943089,r12943109}@ntu.edu.tw, mhyang@ucmerced.edu, sykuo@ntu.edu.tw

Abstract. Crowd counting and localization have become increasingly important in computer vision due to their wide-ranging applications. While point-based strategies have been widely used in crowd counting methods, they face a significant challenge, i.e., the lack of an effective learning strategy to guide the matching process. This deficiency leads to instability in matching point proposals to target points, adversely affecting overall performance. To address this issue, we introduce an effective approach to stabilize the proposal-target matching in point-based methods. We propose Auxiliary Point Guidance (APG) to provide clear and effective guidance for proposal selection and optimization, addressing the core issue of matching uncertainty. Additionally, we develop Implicit Feature Interpolation (IFI) to enable adaptive feature extraction in diverse crowd scenarios, further enhancing the model's robustness and accuracy. Extensive experiments demonstrate the effectiveness of our approach, showing significant improvements in crowd counting and localization performance, particularly under challenging conditions.

Keywords: Crowd Counting, Crowd Localization, Auxiliary Learning , Feature Interpolation

1 Introduction

Recent years have witnessed the advances and importance of crowd counting and localization in numerous tasks, including surveillance, event management, and urban planning [1,8,14,18,21,22,27,46,48]. The pursuit of accurately estimating crowd size and discerning individual locations is fraught with challenges, ranging from fluctuating crowd densities and occlusions to varying environmental settings.

Project page: https://apgcc.github.io/

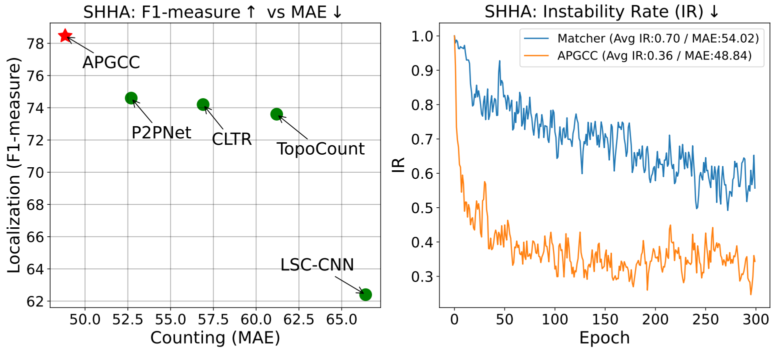

Fig. 1: (Left) Crowd Counting and Localization: Comparison with state-of-the-art methods (e.g., LSC-CNN [39], TopoCount [1], P2PNet [43] and CLTR [20]) demonstrating the proposed APGCC's effectiveness in accurately counting and localizing in crowded scenes. **(Right) Matching Process Instability:** Illustrates the instability in selecting point proposals during the matching process by existing point-based methods (e.g., Matcher [16]) across training epochs, indicated by the Instability Rate (IR), which measures the inconsistency rate of point proposal selection per epoch, leading to limited performance. Both evaluations are conducted on the ShanghaiTech A (SHHA) [53] dataset.

Within the domain of crowd analysis, two principal methodologies emerge: map-based and localization-based approaches. Map-based methods, employing Gaussian kernels to render density maps, effectively provide models with critical information for learning crowd densities. Renowned for their high accuracy in crowd counting, these methods have been validated across a series of studies, including [11,14,25,26,28,31]. Despite their capacity to achieve localization through additional designs [1, 13, 46], they still confront challenges such as the overlapping of maps in densely populated areas and the need for multi-scale representations. This leads to difficulties in precise localization with non-differentiable post-processing techniques like "find-maxima".

Localization-based approaches encompass two divergent strategies: detection-based and point-based methods. Detection-based techniques [29, 39], characterized by the initiation of pseudo ground truth bounding boxes using nearest-neighbor distances, are tailored for specific scenarios but encounter accuracy limitations in highly congested and sparse areas. Despite their practicality, these methods often contend with the constraints of heuristic post-processing, such as non-maximum suppression, potentially leading to inaccuracies [20].

In contrast, the elegance of point-based methods [20, 24, 43] lies in directly using point annotations as learning targets. These frameworks can direct the regression of individual coordinates, simplifying the localization process. These methods are celebrated for their simplicity, end-to-end trainability, and independence from complex pre-processing and multi-scale feature map fusion. However, a significant challenge in point-based methods for crowd analysis is the instability of proposal-target matching during training, as illustrated in Figure 1. In each epoch, a large proportion of target points are matched with different point proposals compared to the previous epoch. This issue arises due to the absence of an effective learning strategy to guide the network in consistently selecting the most appropriate proposals during optimization. Consequently, the constantly changing relationships between proposals and targets lead to vague and unclear learning objectives for each proposal. This uncertainty in the learning process often results in localized inaccuracies, manifesting as either underestimation or overestimation in specific areas of crowded scenes.

In this paper, we address the prevailing issue of uncertainty in proposal-target matching within point-based methods for crowd analysis. We introduce a novel learning paradigm, **A**uxiliary **P**oint **G**uidance **C**rowd **C**ounting (APGCC), designed to instruct the network on the precise selection and optimization of point proposals for matching with target points. APGCC provides a clear and effective directive, ensuring accurate and informed decisions in the proposal selection and optimization process.

To facilitate the application of APGCC, which necessitates feature extraction at arbitrary positions, we propose a method utilizing Implicit Feature Interpolation. This technique adeptly addresses the challenge of accessing features from diverse locations within the network, thereby ensuring the versatility and efficacy of our model in various crowd scenarios. By enhancing the robustness of the matching relationships between proposals and targets, our approach significantly improves the precision and reliability of crowd analysis models.

Extensive experimental results demonstrate that the proposed APG strategy effectively addresses the instability issues in proposal-target matching during the training process (orange curve in Figure 1). Moreover, it significantly enhances the performance of crowd counting. Our method performs robustly and favorably against state-of-the-art schemes in both crowd counting and localization tasks. We make the following contributions in this work:

- We introduce Auxiliary Point Guidance, a novel strategy to address the uncertainty in proposal-target matching within point-based crowd counting methods. APG guides the precise selection and optimization of proposals, enhancing model accuracy.
- We develop an Implicit Feature Interpolation method, enabling effective feature extraction at arbitrary positions. This technique improves the robustness and versatility of our model, particularly in various crowd scenarios.

2 Related Work

In the realm of crowd counting, methodologies are broadly categorized into map-based [2, 2, 11, 14, 18, 25, 25, 26, 28, 31, 33, 49] and localization-based approaches [17, 19, 23, 29, 39], each with distinct strategies and challenges.

Map-based Approaches use Gaussian kernel density maps to achieve localization and counting. Pioneered by researchers like Idrees *et al.* [13] and Gao *et al.* [8], these methods identify individual positions as peaks on density maps.

However, they encounter challenges with overlapping in dense crowds. Innovations such as the Distance Label Map [51], Focal Inverse Distance Transform Map (FIDTM) [21], and Independent Instance Map (IIM) [9] have been introduced to mitigate these issues, though they still require complex post-processing steps. Concurrently, these approaches also advance crowd counting accuracy by integrating density map values, with enhancements like composition loss [13] and inter-domain feature segregation [8]. These approaches successfully reduce overlaps in crowded areas, yet they require a post-processing step, such as "find-maxima", to pinpoint individual locations. Additionally, their reliance on multi-scale feature maps adds to their complexity, detracting from their simplicity and elegance.

Localization-based Approaches: In crowd analysis, localization-based methods integrate both detection-based and point-based strategies. Detection-based approaches, utilizing frameworks like Faster RCNN [37], focus on generating pseudo bounding boxes using techniques such as nearest neighbor distance, as seen in [39], and a winner-take-all loss for refining box selection, especially beneficial for high-resolution images. Liu *et al.* [29] employs curriculum learning to enhance detection and bounding box prediction accuracy. However, these methods often contend with the challenge of pseudo-ground-truth boxes derived from weak point supervision, which can be particularly unreliable in densely populated areas, thus impeding model training and leading to imprecise box predictions. Additionally, they typically involve Non-Maximum Suppression (NMS) in their box filtering process, which is not designed for end-to-end training [48].

In contrast, point-based approaches like those proposed by Song *et al.* [43] (P2PNet), Liang *et al.* [20] (CLTR), and Liu *et al.* [24] (PET) emphasize directly estimating individual head positions, dynamically adjusting to various crowd densities. These methods significantly enhance the accuracy and process efficiency of localization tasks. Nevertheless, their performance can be limited by the instability of proposal-target matching during training, often leading to regional underestimation or overestimation due to unclear learning objectives for proposals.

3 Preliminary: Point-based Crowd Counting Framework

This framework [43] comprises three main components essential for point-based crowd counting: Point Proposal Prediction, Proposal-Target Matching, and Loss Calculation.

Point Proposal Prediction involves generating point proposals from the deep feature map $\mathcal{F}_s$ outputted by the backbone network, where s is the downsampling stride, and $\mathcal{F}_s$ has a size of $H \times W$. The process includes two parallel branches: regression for predicting point coordinate offsets and classification for determining confidence scores. Each pixel on $\mathcal{F}_s$ corresponds to a patch in the input image, with a predefined set of reference points $\mathcal{R} = R_k | k \in \{1, ..., K\}$ where K is the total number of reference points. The regression branch outputs $H \times W \times K$ point proposals, with the coordinates of a proposal $\hat{p}_j = (\hat{x}_j, \hat{y}_j)$

calculated as: $\hat{x}_j = x_k + \gamma \varDelta^k_{jx}$ and $\hat{y}_j = y_k + \gamma \varDelta^k_{jy}$ where γ scales the predicted offsets, $\varDelta^k_{jx}$ and $\varDelta^k_{jy}$ present predicted offsets for its coordinates of a proposal $\hat{p}_j = (\hat{x}_j, \hat{y}_j)$.

Proposal-Target Matching follows the Point Proposal Prediction, utilizing the Hungarian algorithm [16] as proposal-target matching $\varOmega(\mathcal{P}, \hat{\mathcal{P}}, \mathcal{D})$. This strategy ensures a one-to-one correspondence where each ground truth target from $\mathcal{P}$ is matched with a point proposal in $\hat{\mathcal{P}}$. The matching is based on the pair-wise cost matrix $\mathcal{D}$ of size $N \times M$ where N and M denote the number of ground truth points and point proposals. The matrix $\mathcal{D}$ combines the Euclidean distance between point pairs and the confidence score $\hat{c}_j$ of each proposal, defined as $D(P, \hat{P}) = \left(\tau \left|\left|p_i - \hat{p}_j\right|\right|_2 - \hat{c}_j\right)_{i \in N, j \in M}$, where τ balances the pixel distance and $\hat{c}_j$ is the confidence score of proposal $\hat{p}_j$.

After the matching process, in the optimal matching results denoted as $\varTheta$, each ground truth point p_i is optimally matched to a point proposal $\hat{p}_j$, with the matching result represented by the permutation $\psi = \varTheta(\mathcal{P}, \hat{\mathcal{P}}, \mathcal{D})$. Thus, $\hat{p}_{\psi(i)}$ is the proposal matched to ground truth point p_i. The set of matched proposals, $\hat{\mathcal{P}}\text{pos} = \hat{p}_{\psi(i)} | i \in \{1, ..., N\}$, are considered positives, while the unmatched ones, $\hat{\mathcal{P}}\text{neg} = \hat{p}_{\psi(i)} | i \in \{N+1, ..., M\}$, are negatives.

Loss Calculation integrates Euclidean loss $\mathcal{L}_{loc}$ for point regression and Cross Entropy loss $\mathcal{L}_{cls}$ for proposal classification. The combined loss function $\mathcal{L}_{point}$ is:

$$\mathcal{L}_{cls} = -\frac{1}{M}\left(\sum_{i=1}^{N} \log \hat{c}_{\psi(i)} + \lambda_1 \sum_{i=N+1}^{M} \log(1 - \hat{c}_{\psi(i)})\right), \tag{1}$$

$$\mathcal{L}_{loc} = \frac{1}{N}\sum_{i=1}^{N} \left|\left|p_i - \hat{p}_{\psi(i)}\right|\right|_2^2, \tag{2}$$

$$\mathcal{L}_{point} = \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{loc}, \tag{3}$$

where $\hat{c}_{\psi(i)}$ is the confidence score of the matched proposal $\hat{p}_{\psi(i)}$, λ_1 adjusts the impact of negative proposals, and λ_2 balances the regression loss.

4 Proposed Method

While current Point-based Approaches demonstrate promising results in crowd counting and localization, we identified instability in the optimization of the matching process, potentially limiting overall performance. We introduce several components designed to stabilize and enhance the matching mechanism to address this challenge.

4.1 Auxiliary Point Guidance

We introduce an explicit guidance mechanism to enhance the optimization process's stability during the network's matching phase. As shown in Figure 2,

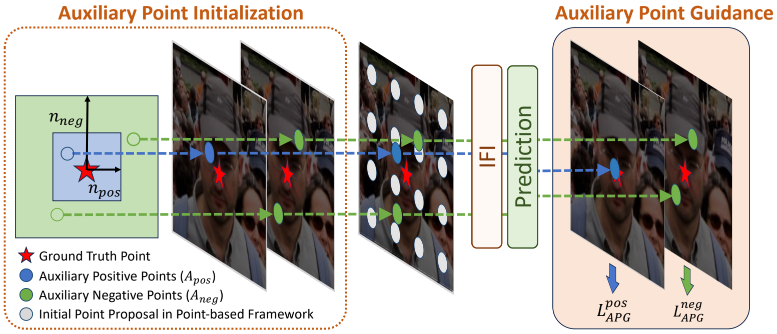
Auxiliary Point Initialization
Auxiliary Point Guidance
n_{neg}
n_{pos}
IFI
Prediction
Ground Truth Point
Auxiliary Positive Points (A_{pos})
Auxiliary Negative Points (A_{neg})
Initial Point Proposal in Point-based Framework
L_{APG}^{pos}
L_{APG}^{neg}

Fig. 2: Illustration of the Auxiliary Point Guidance framework. During the model's training, we additionally introduce auxiliary positive (A_{pos}) and negative (A_{pos}) points based on each ground truth position to guide the network's learning. This approach helps in directing the optimization process more effectively by distinguishing between potential positive and negative matches.

this involves the strategic designation of auxiliary positive (A_{pos}) and negative (A_{neg}) points within the optimization framework, determined based on ground truth coordinates (x, y). The sets of positive and negative points are defined as $A^i_{\text{pos}} = \{(x+R^{i,x}_{\text{pos}}, y+R^{i,y}_{\text{pos}}) \mid i = 1, 2, \ldots, k_{\text{pos}}\}$ and $A^j_{\text{neg}} = \{(x+R^{j,x}_{\text{neg}}, y+R^{j,y}_{\text{neg}}) \mid j = 1, 2, \ldots, k_{\text{neg}}\}$. Here, $R^{i,x}_{\text{pos}}$ and $R^{i,y}_{\text{pos}}$ represent a series of randomness numbers used to generate the x and y coordinates of positive points, respectively, with each number uniformly distributed between $-n_{\text{pos}}$ and n_{pos}. Similarly, $R^{j,x}_{\text{neg}}$ and $R^{j,y}_{\text{neg}}$ denote series of randomness numbers for generating the x and y coordinates of negative points, each uniformly distributed between $[-n_{\text{neg}}, -n_{\text{pos}}]$ or $[n_{\text{pos}}, n_{\text{neg}}]$. The variables k_{pos} and k_{neg} denote the total number of positive and negative points generated, respectively. Each set R^i_{pos} and R^j_{neg} is used to create a unique set of coordinates for A_{pos} and A_{neg}, thereby offsetting the ground truth position (x, y) by these randomness numbers.

Based on the auxiliary positive points A_{pos}, we extract their corresponding features. From these features, we then predict the confidence $\hat{c}^\star_{\text{pos}}$ and offset and then calculate the position of the proposal $\hat{p}^\star_{\text{pos}}$ for each point. Our objective is to ensure that the confidence of auxiliary positive points is as close to one as possible and that their predicted offsets closely match the added randomness number. To achieve this, we formulate the loss function for the auxiliary positive point as follows:

$$\mathcal{L}^{pos}_{APG} = \frac{1}{N}\frac{1}{k_{\text{pos}}}\sum_{l=1}^{N}\sum_{i=1}^{k_{\text{pos}}}\left(\log\hat{c}^\star_{\text{pos}}(l,i) + \lambda_3 \left|\left|p_l - \hat{p}^\star_{\text{pos}}(l,i)\right|\right|_2^2\right), \tag{4}$$

where λ_3 represents a scaling factor.

For the auxiliary negative points (A_{neg}), our aim is for their confidence $\hat{c}^\star_{\text{neg}}$ to be as close to zero as possible. Similarly, we desire their offsets $\Delta^\star_{\text{neg}}$ to approach zero, preventing negative points from using offsets to bring their proposal

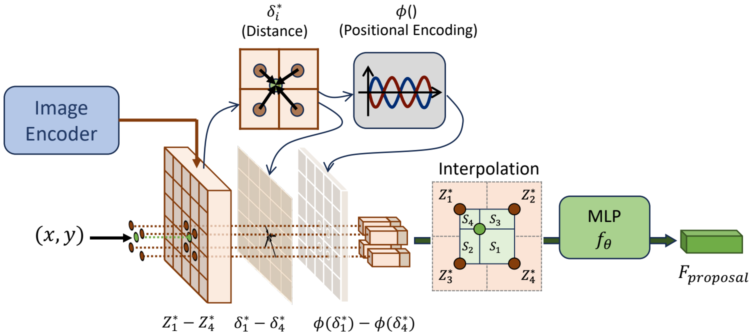

Fig. 3: Illustration of Implicit Feature Interpolation. Given an arbitrary desired point position (x, y), we concatenate the nearest four feature maps ($Z_1^\star$ - $Z_4^\star$) along with their distances ($\delta_1^\star$ - $\delta_4^\star$) to the (x, y) with positional encoding ϕ and utilize a Multi-Layer Perceptron (MLP) f_θ to interpolate the latent feature for that specific location. This approach enables precise feature extraction at non-grid locations, facilitating more flexible and accurate feature representation.

coordinates close to the ground truth. This is crucial to mitigate the potential of these negative points being erroneously considered as matched proposals during the matching process. The loss function specifically formulated for auxiliary negative points is as follows:

$$\mathcal{L}_{APG}^{neg} = \frac{1}{N}\frac{1}{k_{\text{neg}}}\sum_{l=1}^{N}\sum_{j=1}^{k_{\text{neg}}}(\log(1-\hat{c}_{\text{neg}}^{\star}(l,j)) + \lambda_4 \left|\left|\Delta_{\text{neg}}^{\star}(l,j)\right|\right|_2^2) \, , \tag{5}$$

where λ_4 represents a scaling factor.

The total loss of the Auxiliary Point Guidance can be formulated as:

$$\mathcal{L}_{APG} = \mathcal{L}_{APG}^{pos} + \mathcal{L}_{APG}^{neg}. \tag{6}$$

Through this additional guidance, we can direct the network to train point proposals closest to the ground truth points as positive points, while treating those farther away as negative points. This guidance assists the network in consistently selecting the same positive point for each ground truth point during the matching process. Importantly, the chosen positive point is likely to be the correct match, being in close proximity to the ground truth point. By employing this guidance, we address the instability issue inherent in the matching process, thereby enhancing the network's performance.

However, since auxiliary points are randomly assigned based on ground truth coordinates, traditional bilinear interpolation is not suitable for extracting features at these arbitrary positions. Therefore, we propose the use of implicit feature interpolation to obtain these features. The details of this approach will be described in the following section.

4.2 Implicit Feature Interpolation

Implicit functions have demonstrated their efficacy in providing robust and continuous feature representations, significantly benefiting various computer vision tasks as evidenced in previous studies [4, 34, 35]. In Auxiliary Point Guidance, we leverage implicit function-based interpolation to extract latent features that are both arbitrary and robust. As depicted in Figure 3, for a given point location (x, y), we first determine the four nearest latent features, denoted as $Z_i^*|i \in \{1, ..., 4\}$. We then calculate their respective spatial distances from the target latent feature, represented as $\delta_i^*|i \in \{1, ..., 4\}$. These four latent features, along with their calculated distances, are concatenated channel-wise. This concatenated information is then fed into a MLP to yield the target latent feature. However, it is known that MLPs tend to prioritize low-frequency information, often overlooking crucial high-frequency details, which can impact the performance of the MLP [3, 36, 44]. To counter this limitation, we employ positional encoding as suggested in [52], enhancing the dimensionality of the distance information. By integrating positional encoding with the distance data, we address this high-frequency detail loss. The entire implicit feature interpolation process is encapsulated in the following formulation:

$$F_{proposal}(x, y) = \sum_{i=1}^{4} \frac{S_i}{S} f_\theta(Z_i^*, \delta_i^*, \phi(\delta_i^*)), \tag{7}$$

where S_i represents the area surrounding the diagonal point with the target point, and S is the sum of these areas, calculated as $S = \sum_{i=1}^{4} S_i$. Here, $f_\theta(\cdot)$ symbolizes the MLP, $\phi(\cdot)$ denotes positional encoding, and $F_{proposal}(x, y)$ is the resultant interpolated feature for the point (x, y).

4.3 Architecture Overview

Our architecture, as illustrated in Figure 4, begins with the extraction of image features using a pre-trained backbone, specifically VGG-16 [41]. We focus on the feature maps from the final two layers (i.e., conv 3 and conv 4). These features are then enhanced for scale diversity through the application of Atrous Spatial Pyramid Pooling (ASPP) [7]. Following this, each set of features undergoes an implicit feature interpolation process, which results in the computation of the corresponding features $F_{proposal}(x, y)$. The interpolated features are subsequently concatenated and input into both the confidence and regression modules. These modules are responsible for predicting the confidence level and offset for each point in the image. The training of the network is accomplished using a combination of the original point-based constraint, as defined in (3), and the proposed Auxiliary Point Guidance, as stated in (6). The total loss function, denoted as $\mathcal{L}_{overall}$, is formulated as follows:

$$\mathcal{L}_{overall} = \mathcal{L}_{point} + \lambda_5 \mathcal{L}_{APG}, \tag{8}$$

where λ_5 represents a scaling factor.

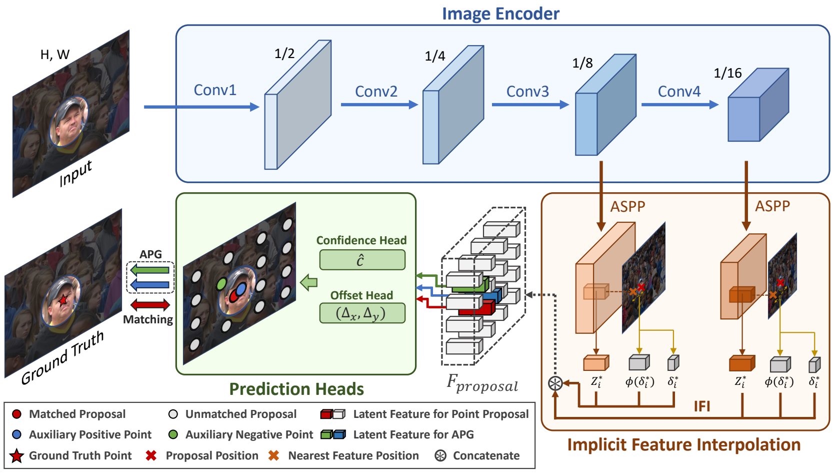

Fig. 4: Illustration of the proposed APGCC for crowd counting and localization. A VGG encoder extracts image features, where features from conv3 and conv4 layers undergo refinement via Atrous Spatial Pyramid Pooling [7]. Subsequently, target latent features are interpolated using implicit feature interpolation. These latent features are then processed through a prediction head to obtain confidence score $\hat{c}$ and offsets (Δ_x, Δ_y), facilitating precise crowd counting and localization.

5 Implementation Details

5.1 Datasets

We use the ShanghaiTechA [53], ShanghaiTechB [53], UCF_CC_50 [12], UCF-QNRF [13], JHU-Crowd++ [42], and NWPU-Crowd [47] datasets to evaluate the performance of the proposed method against the state-of-the-art approaches. **ShanghaiTech A** dataset [53] includes 482 images with 244,167 annotated points. The dataset is divided into 300 training images and 182 testing images.

ShanghaiTech B (SHHB) dataset [53] features 716 images and 88,488 annotated points, with a split of 400 training images and 316 testing images.

UCF_CC_50 dataset [12] encompasses 50 images, totaling 63,974 annotated points. We adhere to a five-fold cross-validation as outlined in [12].

UCF-QNRF dataset [13] contains 1,535 high-resolution web-collected images, with over 1.25 million annotated points. It splits into 1,201 training images and 334 testing images, featuring a broad people count range from 49 to 12,865.

JHU-Crowd++ dataset [42] comprises 4,372 images, totaling 1.51 million annotated points. It allocates 2,272 images for training, 500 for validation, and reserves 1,600 images for testing.

NWPU-Crowd dataset [47] includes 5,109 images with more than 2.13 million annotated points, distributed across 3,109 training images, 500 validation images, and 1,500 testing images.

5.2 Evaluation Protocol

Counting Metrics. We employ Mean Absolute Error (MAE) and Mean Squared Error (MSE) as our primary performance metrics, in line with standard practices in the field, defined as $MAE = \frac{1}{Q}\sum_{i=1}^{Q}|GT_i - N_i|$, $MSE = \sqrt{\frac{1}{Q}\sum_{i=1}^{Q}(GT_i - N_i)^2}$, where Q represents the total number of images in the dataset, with GT_i and N_i indicating the actual and predicted crowd counts for the i-th image, respectively.
Localization Metrics. To assess localization accuracy, we utilize Precision (P), Recall (R), and F1-measure (F), following the methodologies in [13,47]. A predicted point is considered a True Positive (TP) if its distance from the corresponding ground truth (GT) point is within a specified threshold σ. For the NWPU-Crowd dataset [47], which includes box-level annotations, σ is defined as $\sqrt{(w^2 + h^2)}/2$, where w and h are the width and height of each head. In contrast, for the ShanghaiTech dataset, we apply fixed thresholds of $\sigma = 4$ and $\sigma = 8$.

5.3 Training Details

We utilize Adam optimization [15] with a learning rate of 10^{-4} for general model optimization. Given that the VGG-16 backbone network weights are pre-trained on ImageNet, a reduced learning rate of 10^{-5} is applied for these components. We adopted a grid layout strategy [43] for mapping proposals. The initial point proposal stride is set at $s = 8$. The number of reference points K varies depending on the dataset: 4 for most and 8 for the QNRF dataset, aligned with dataset statistics to ensure $M > N$. The prediction head comprises four layers with hidden feature dimensions of [1024, 512, 256, 256], and a shared prediction head is used for our point proposals. For point regression, we set γ at 100, and the matching weight term τ at 5×10^{-2}. In auxiliary points learning, the number of positive and negative points (k_{pos}, k_{neg}) are set to (2, 2). Randomness ranges (n_{pos}, n_{neg}) are set to (2, 8). The loss coefficients are adjusted as $\lambda_1 = 0.5$, $\lambda_2 = 2 \times 10^{-4}$, $\lambda_3 = 2 \times 10^{-4}$, $\lambda_4 = 2 \times 10^{-4}$, and $\lambda_5 = 0.2$ to balance different term contributions.

Data augmentation involves initial random scaling (factor range: [0.7, 1.3]), ensuring the shorter side is at least 128 pixels. Images are then randomly cropped to 128 × 128 patches and subjected to random flipping with a 0.5 probability. The training batch size is 8. The longer side of each image is restricted to 1920 pixels for UCF-QNRF, JHU-Crowd++, and NWPU-Crowd, while maintaining the original aspect ratio.

6 Experimental Results

In this section, we present the evaluation results of the proposed method for crowd counting and localization. More results are available in the supplementary material.

Table 1: Evaluation of crowd counting on SHHA [53], SHHB [53], UCF-QNRF [13], JHU-Crowd [42] datasets.

Method	Localization	Manner	SHHA		SHHB		UCF-QNRF		JHU-Crowd+	
			MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓
AMSNet [11]	✗	Map-based	56.7	93.4	6.7	10.2	101.8	163.2	-	-
SDA+DM [30]	✗	Map-based	55.0	92.7	-	-	80.7	146.3	59.3	248.9
GauNet+CSRNet [5]	✗	Map-based	61.2	97.8	7.6	12.7	84.2	152.4	69.4	262.4
DC [50]	✗	Map-based	61.6	96.7	7.1	11.1	91.4	157.5	67.2	288.2
ChfL [40]	✗	Map-based	57.5	94.3	6.9	11.0	80.3	137.6	57.0	235.7
HMoDE [6]	✗	Map-based	54.4	87.4	6.2	9.8	-	-	55.7	**214.6**
LSC-CNN [39]	✓	Detection-based	66.4	117.0	8.1	12.7	120.5	218.2	112.7	454.4
TopoCount [1]	✓	Detection-based	61.2	104.6	7.8	13.7	89.0	159.0	60.9	267.4
GL [46]	✓	Map-based	61.3	95.4	7.3	11.7	84.3	147.5	59.9	259.5
P2PNet [43]	✓	Point-based	52.7	85.1	6.2	9.9	85.3	154.5	-	-
CLTR [20]	✓	Point-based	56.9	95.2	6.5	10.6	85.8	141.3	59.5	240.6
PET [24]	✓	Point-based	49.3	78.7	6.1	9.6	**79.5**	144.3	58.5	238.0
APGCC	✓	Point-based	**48.8**	**76.7**	**5.6**	**8.7**	80.1	**136.6**	**54.3**	225.9

Table 2: Evaluation of crowd counting on UCF_CC_50 [12] dataset.

Method	Localization	Manner	MAE ↓	MSE ↓
BL [31]	✗	Map-based	229.3	308.2
AMSNet [11]	✗	Map-based	208.4	297.3
GauNet+CSRNet [5]	✗	Map-based	215.4	296.4
HMoDE [6]	✗	Map-based	159.6	211.2
P2PNet [43]	✓	Point-based	172.7	256.1
PET [24]	✓	Point-based	159.9	223.7
APGCC	✓	Point-based	**154.8**	**205.5**

Table 3: Evaluation of crowd counting on NWPU [47] dataset.

Method	Localization	Manner	MAE ↓	MSE ↓
NoisyCC [45]	✗	Map-based	96.9	534.2
UOT [32]	✗	Map-based	87.8	387.5
MAN [22]	✗	Map-based	76.5	323.0
ChfL [40]	✗	Map-based	76.8	343.0
HMoDE+REL [6]	✗	Map-based	73.4	331.8
RAZ [23]	✓	Map-based	151.4	634.6
GL [46]	✓	Map-based	79.3	346.1
AutoScale [51]	✓	Map-based	123.9	515.5
TopoCount [1]	✓	Detection-based	107.8	438.5
P2PNet [43]	✓	Point-based	77.4	362.0
CLTR [20]	✓	Point-based	74.3	333.8
PET [24]	✓	Point-based	74.4	328.5
APGCC	✓	Point-based	**71.7**	**284.4**

6.1 Evaluation on Crowd Counting

This section outlines our comparative analysis of crowd counting methods, where our approach is benchmarked against an array of state-of-the-art techniques across diverse datasets. We evaluate our performance against both map-based [5, 6, 11, 22, 23, 30–32, 40, 45, 46, 50, 51], detection-based [1, 39] and point-based [20, 24, 43] methodologies. Our experiments, detailed in Tables 1, 2, and 3, highlight APGCC's leading performance, with the **best** results in bold and the second-best results underlined. These findings affirm the effectiveness and adaptability of our approach in various crowd counting scenarios.

Table 1 focuses on datasets such as SHHA [53], SHHB [53], UCF-QNRF [13] and JHU-Crowd [42], showcasing APGCC's significant improvements in accuracy metrics like MAE and MSE. For example, compared to P2PNet [43] on the SHHA [53] and SHHB [53] datasets, APGCC achieved substantial reductions in both MAE and MSE, demonstrating its effectiveness even in sparse and simple scene conditions.

In Table 2, we specifically examine the UCF_CC_50 dataset [12], a challenging set of 50 images with complex scenes. Our approach notably excels, achieving impressive results that underscore the efficiency and stability of our learning strategy, particularly beneficial for datasets with limited images.

Finally, Table 3 presents our performance on the NWPU-Crowd dataset [47], the most extensive congested dataset considered in our study. Our approach out-

Table 4: Evaluation of crowd localization on NWPU [47] dataset.

Method	Manner	σ_l (large threshold)			σ_s (small threshold)		
		F(%) ↑	P(%) ↑	R(%) ↑	F(%) ↑	P(%) ↑	R(%) ↑
RAZ [23]	Map-based	59.8	66.6	54.3	57.6	47.0	51.7
AutoScale [51]	Map-based	67.3	57.4	62.0	-	-	-
GL [46]	Map-based	66.0	80.0	56.2	58.7	71.1	50.0
TinyFaces [10]	Detection-based	56.7	52.9	61.1	52.6	49.1	56.6
TopoCount [1]	Detection-based	63.7	65.1	62.4	-	-	-
P2PNet [43]	Point-based	71.2	72.9	69.5	67.5	68.4	**66.6**
CLTR [20]	Point-based	68.5	69.4	67.6	59.1	59.9	58.3
PET [24]	Point-based	74.2	75.2	73.2	67.5	68.4	**66.6**
APGCC	Point-based	**76.4**	**79.2**	**73.6**	**68.9**	**71.5**	66.5

Table 5: Evaluation of crowd localization on SHHA [53] dataset.

Method	Manner	$\sigma = 4$			$\sigma = 8$		
		F(%) ↑	P(%) ↑	R(%) ↑	F(%) ↑	P(%) ↑	R(%) ↑
LOBB [38]	Map-based	25.9	34.9	20.7	53.9	67.6	44.8
LCFCN [17]	Detection-based	32.5	43.3	26.0	56.3	75.1	45.1
LSC-CNN [39]	Detection-based	32.6	33.4	31.9	62.4	63.9	61.0
TopoCount [1]	Detection-based	41.1	41.7	40.6	73.6	74.6	72.7
P2PNet [43]	Point-based	40.6	41.5	39.8	74.6	76.2	73.1
CLTR [20]	Point-based	43.2	43.6	42.7	74.2	74.9	73.5
APGCC	Point-based	**48.7**	**49.2**	**48.3**	**78.4**	**79.1**	**77.7**

performs the competition, including the second-best method, HMoDE+REL [6], by achieving lower MAE and MSE. This success can be attributed mainly to our implementation of Auxiliary Points Guidance and the enhancement provided by the Implicit Feature Interpolation technique, which together significantly improve model reliability and adaptability to different scales and densities.

6.2 Evaluation on Crowd Localization

We benchmark our approach against a diverse array of methods, including map-based methods such as RAZ [23], AutoScale [51], LOBB [38], and GL [46]; detection-based methods like TinyFaces [10], TopoCount [1], LSC-CNN [39], and LCFCN [17]; as well as point-based methods including P2PNet [43], CLTR [20], and PET [24].

Table 4, focusing on the NWPU dataset [47], showcases APGCC's superior performance. Compared to detection-based methods that utilize box-level annotations and other point-based approaches, APGCC leverages IFI to acquire precise features and utilizes closer proposal predictions to achieve optimal precision. Conversely, in the SHHA dataset [53], as detailed in Table 5, APGCC secures comprehensive improvements: at a $\sigma = 4$, the F1-measure increased by 5.5%, and at $\sigma = 8$, it rose by 3.8%.

6.3 Evaluation of Model Complexity

In Table 6, we benchmark APGCC against other point-based methods, focusing on the number of parameters and inference time. The inference time evaluations are conducted on an NVIDIA 3090 GPU with an input resolution of 1024×1024. The findings illustrate that APGCC maintains efficient computational complexity and delivers superior performance in crowd counting and localization tasks. Note that since our APG training mechanism is employed only during training, it does not incur additional computational overhead during inference. Compared to the original point-based method (i.e., P2PNet [43]) that utilizes traditional upsampling to process features, our use of IFI allows for more accurate representation learning with fewer parameters. This method enhances computaional efficiency, as employing MLPs for feature interpolation is known to be efficient [35].

6.4 Ablation Study

We evaluate the effectiveness of the proposed two modules, APG and IFI. We evaluate the performance on SHHA dataset [53].

Effectiveness of APG. We explore the impact of different optimization strategies, with several distinct settings as follows: (a) "Matcher" solely employs the matching strategy as described in Section 3 and [43], (b) "Nearest Point" directly selects the proposal closest to the ground truth as the positive proposal, with all others considered negative, (c) "APG", which exclusively utilizes the proposed APG for training, and (d) "Matcher + APG" (Ours). The experimental results, as shown in Table 7, indicate that while strategy (b) may seem intuitive and straightforward to design, it risks multiple ground truths mapping to the same proposal, severely underestimating the final counting. Consequently, using (a) can guide the model on how to allocate proposals for learning and introduce confidence information to enhance discrimination. Our proposed APG effectively addresses the shortcomings of the nearest point, providing an equivalent number of proposals for the model to learn to match the closest proposals. However, as auxiliary positive points cannot be provided during the inference phase, relying solely on APG can make the model overly dependent on reference values. Therefore, by combining the advantages of Matcher and APG (i.e., (d)), we not only teach the model how to allocate a fixed number of proposals but also guide it to make more elegant choices.

Optimizing APG Setup. Our exploration into optimizing APG focuses on two key aspects: (i) determining the optimal number of potential positive and negative points (k_{pos} and k_{neg}), as defined in Section 4.1, and (ii) adjusting the randomness scale of APG (n_{neg} and n_{pos}), with findings detailed in Tables 8 and 9.

Table 8 explores the impact of the number of auxiliary positive and negative points on performance, essential for validating the APG's effectiveness. The results indicate that while using only auxiliary positive points slightly favors the selection of nearest proposals, it's limited in preventing duplicate predictions. Introducing auxiliary negative points enhances differentiation and training stability by encouraging the model to reject distant proposals. Despite an increase in auxiliary point pairs leading to better stabilization (lower Avg. IR and Avg. Δ), the effect on final performance (MAE) is minimal. Thus, we suggest utilizing a balanced (2, 2) ratio of positive to negative points to achieve better learning performance. This is because training with a (5, 5) setting requires roughly twice the training effort compared to (2, 2), without a significant improvement in performance.

Additionally, the degree of randomness applied to auxiliary proposals plays a pivotal role in the configuration of the APG. Our experiments, executed with a stride of 8, have demonstrated that a precise range of randomness is crucial for attaining optimal results, as evidenced in Table 9. While a constrained randomness range may limit the diversity in proposal selection, an excessive range of randomness could jeopardize the model's confidence uniformity across different areas, impacting its effectiveness.

Table 6: Comparison of model complexity with Point-based Approaches.

Method	P2PNet [43]	CLTR [20]	PET [24]	APGCC
Parameters (M) ↓	21.6	43.4	20.9	**18.68**
Inference Time (s) ↓	0.074	0.107	0.097	**0.071**

Table 7: Analysis on alternatives of optimization strategies.

Setting	Strategy	MAE ↓	MSE ↓
(a)	Matcher	54.04	86.97
(b)	Nearest Point	76.91	118.60
(c)	APG	58.46	96.71
(d)	Matcher + APG (Ours)	**48.84**	**76.79**

Table 8: Evaluation of different number of auxiliary points

Num. of Auxiliary Points (k_{pos}, k_{neg})	(0, 0)	(1, 0)	(2, 0)	(1, 1)	(2, 2)	(5, 5)
MAE ↓	54.04	51.57	51.47	49.24	48.84	**48.81**
Avg. IR ↓	0.70	0.49	0.48	0.38	0.36	0.34
Avg. Δ ↓	6.87	3.89	3.37	1.62	1.58	1.49

Table 9: Comparison of different range of randomness for auxiliary points

Randomness Range (n_{pos}, n_{neg})	(1, 4)	(2, 8)	(3, 12)	(4, 16)
MAE ↓	49.25	**48.84**	50.23	51.23

Table 10: Ablation study of Implicit Feature Interpolation.

Setting	Method	MAE ↓	MSE ↓	F1@4 ↑	F1@8 ↑
(a)	Nearest Neighbor without MLP	53.16	83.31	43.59	76.27
(b)	Bilinear Interpolation without MLP	51.25	79.92	45.78	77.67
(c)	IFI with Single Reference Point	49.73	77.97	47.40	78.27
(d)	IFI w/o Positional Encoding	49.24	78.27	47.97	78.38
(e)	IFI	**48.84**	**76.79**	**48.76**	**78.46**

Effectiveness of IFI. To accurately capture the correct features for proposals at arbitrary positions, we introduce IFI. Other feasible approaches include: (a) Nearest Neighbor without MLP, which directly utilizes the closest latent feature without any transformation; (b) Bilinear Interpolation without MLP, deriving features at each position through bilinear interpolation, implying no use of coordinate and continuous function transformation; (c) IFI solely employing a single reference point for continuous transformation (MLP with coordinate information); (d) IFI w/o Positional Encoding; and (e) IFI.

The results, as displayed in Table 10, reveal several clear trends. First, the use of interpolation outperforms the nearest-neighbor approach by providing a richer feature context. Second, a comparison between (b) and (d) highlights the benefits of using distance information for continuous transformation. Third, incorporating Positional Encoding significantly aids the MLP in achieving better learning outcomes. By integrating all these methods, we can notably enhance the representation features obtained at any given position.

7 Conclusion

In this paper, we proposed Auxiliary Point Guidance and Implicit Feature Interpolation to address challenges in point-based crowd counting and localization. APGCC improved the stability of proposal-target matching and enabled accurate feature extraction at any position. Extensive experiments against state-of-the-art methods, our approach showed superior performance in various scenarios.

Acknowledgements

We thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources. This research was supported by the Research Center for Information Technology Innovation and Taiwan's National Science and Technology Council under Grant NSTC112-2634-F-001-001-MBK, NSTC 111-2221-E-002-136-MY3.

References

1. Abousamra, S., Hoai, M., Samaras, D., Chen, C.: Localization in the crowd with topological constraints. In: AAAI (2021) 1, 2, 11, 12
2. Bai, S., He, Z., Qiao, Y., Hu, H., Wu, W., Yan, J.: Adaptive dilated network with self-correction supervision for counting. In: CVPR (2020) 3
3. Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., Kritchman, S.: Frequency bias in neural networks for input of non-uniform density. In: ICML (2020) 8
4. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. In: CVPR (2021) 8
5. Cheng, Z.Q., Dai, Q., Li, H., Song, J., Wu, X., Hauptmann, A.G.: Rethinking spatial invariance of convolutional networks for object counting. In: CVPR (2022) 11
6. Du, Z., Shi, M., Deng, J., Zafeiriou, S.: Redesigning multi-scale neural network for crowd counting. TIP (2023) 11, 12
7. Florian, L.C., Adam, S.H.: Rethinking atrous convolution for semantic image segmentation. In: CVPR (2017) 8, 9
8. Gao, J., Han, T., Wang, Q., Yuan, Y.: Domain-adaptive crowd counting via inter-domain features segregation and gaussian-prior reconstruction. arXiv preprint arXiv:1912.03677 (2019) 1, 3, 4
9. Gao, J., Han, T., Yuan, Y., Wang, Q.: Learning independent instance maps for crowd localization. arXiv preprint arXiv:2012.04164 (2020) 4
10. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 951–959 (2017) 12
11. Hu, Y., Jiang, X., Liu, X., Zhang, B., Han, J., Cao, X., Doermann, D.: Nas-count: Counting-by-density with neural architecture search. In: ECCV (2020) 2, 3, 11
12. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: CVPR (2013) 9, 11
13. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: ECCV (2018) 2, 3, 4, 9, 10, 11
14. Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., Yang, X., Pang, Y.: Attention scaling for crowd counting. In: CVPR (2020) 1, 2, 3
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10
16. Kuhn, H.W.: The hungarian method for the assignment problem. NRL (1955) 2, 5
17. Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M.: Where are the blobs: Counting by localization with point supervision. In: ECCV (2018) 3, 12

18. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: CVPR (2018) 1, 3
19. Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density map regression guided detection network for rgb-d crowd counting and localization. In: CVPR (2019) 3
20. Liang, D., Xu, W., Bai, X.: An end-to-end transformer model for crowd localization. In: ECCV (2022) 2, 4, 11, 12, 14
21. Liang, D., Xu, W., Zhu, Y., Zhou, Y.: Focal inverse distance transform maps for crowd localization. TMM (2022) 1, 4
22. Lin, H., Ma, Z., Ji, R., Wang, Y., Hong, X.: Boosting crowd counting via multifaceted attention. In: CVPR (2022) 1, 11
23. Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: CVPR (2019) 3, 11, 12
24. Liu, C., Lu, H., Cao, Z., Liu, T.: Point-query quadtree for crowd counting, localization, and more. In: ICCV. pp. 1676–1685 (2023) 2, 4, 11, 12, 14
25. Liu, L., Lu, H., Xiong, H., Xian, K., Cao, Z., Shen, C.: Counting objects by blockwise classification. TCSVT (2019) 2, 3
26. Liu, L., Lu, H., Zou, H., Xiong, H., Cao, Z., Shen, C.: Weighing counts: Sequential crowd counting by reinforcement learning. In: ECCV (2020) 2, 3
27. Liu, W., Durasov, N., Fua, P.: Leveraging self-supervision for cross-domain crowd counting. In: CVPR (2022) 1
28. Liu, X., Yang, J., Ding, W.: Adaptive mixture regression network with local counting map for crowd counting. In: ECCV (2020) 2, 3
29. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: CVPR (2019) 2, 3, 4
30. Ma, Z., Hong, X., Wei, X., Qiu, Y., Gong, Y.: Towards a universal model for cross-dataset crowd counting. In: ICCV (2021) 11
31. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: ICCV (2019) 2, 3, 11
32. Ma, Z., Wei, X., Hong, X., Lin, H., Qiu, Y., Gong, Y.: Learning to count via unbalanced optimal transport. In: AAAI (2021) 11
33. Miao, Y., Lin, Z., Ding, G., Han, J.: Shallow feature based dense attention network for crowd counting. In: AAAI (2020) 3
34. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. ACM (2021) 8
35. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR (2019) 8, 12
36. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: ICML. pp. 5301–5310. PMLR (2019) 8
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NeurIPS (2015) 4
38. Ribera, J., Guera, D., Chen, Y., Delp, E.J.: Locating objects without bounding boxes. In: CVPR (2019) 12
39. Sam, D.B., Peri, S.V., Sundararaman, M.N., Kamath, A., Babu, R.V.: Locate, size, and count: accurately resolving people in dense crowds via detection. TPAMI (2020) 2, 3, 4, 11, 12
40. Shu, W., Wan, J., Tan, K.C., Kwong, S., Chan, A.B.: Crowd counting in the frequency domain. In: CVPR (2022) 11

41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 8
42. Sindagi, V.A., Yasarla, R., Patel, V.M.: Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. TPAMI (2020) 9, 11
43. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: Rethinking counting and localization in crowds: A purely point-based framework. In: ICCV (2021) 2, 4, 10, 11, 12, 13, 14
44. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS (2020) 8
45. Wan, J., Chan, A.: Modeling noisy annotations for crowd counting. NeurIPS (2020) 11
46. Wan, J., Liu, Z., Chan, A.B.: A generalized loss function for crowd counting and localization. In: CVPR (2021) 1, 2, 11, 12
47. Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: A large-scale benchmark for crowd counting and localization. TPAMI (2020) 9, 10, 11, 12
48. Wang, Y., Hou, J., Hou, X., Chau, L.P.: A self-training approach for point-supervised object detection and counting in crowds. TIP (2021) 1, 4
49. Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., Shen, C.: From open set to closed set: Counting objects by spatial divide-and-conquer. In: ICCV (2019) 3
50. Xiong, H., Yao, A.: Discrete-constrained regression for local counting models. In: ECCV (2022) 11
51. Xu, C., Liang, D., Xu, Y., Bai, S., Zhan, W., Bai, X., Tomizuka, M.: Autoscale: learning to scale for crowd counting. IJCV (2022) 4, 11, 12
52. Xu, X., Wang, Z., Shi, H.: Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. arXiv preprint arXiv:2103.12716 (2021) 8
53. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: CVPR (2016) 2, 9, 11, 12, 13